Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 6487

The Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA)

G. P. Stiller (Editor)
Institut für Meteorologie und Klimaforschung

Dezember 2000

Forschungszentrum Karlsruhe

Technik und Umwelt

Wissenschaftliche Berichte

FZKA 6487

The Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA)

G.P. Stiller (Editor)

Institut für Meteorologie und Klimaforschung

Forschungszentrum Karlsruhe GmbH, Karlsruhe
2000

Als Manuskript gedruckt

Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe

Mitglied der Hermann von Helmholtz-Gemeinschaft
Deutscher Forschungszentren (HGF)

ISSN 0947-8620

Zusammenfassung

Der Karlsruher optimierte und genaue Strahlungstransport-Algorithmus KOPRA

Der Karlsruher optimierte und genaue Strahlungstransport-Algorithmus KOPRA ist ein FORTRAN90 Computercode zur Modellierung des atmosphärischen Strahlungstransports im spektralen Bereich des mittleren Infrarots. Er wurde als eigenständiger Algorithmus entwickelt, der alle wesentlichen physikalischen Effekte, die von der Troposphäre bis zur Thermosphäre im Strahlungstransport auftreten, berücksichtigt. Außerdem wird die instrumentelle Antwortfunktion des MIPAS/ENVISAT Experiments (neben anderen, weiter verbreiteten) modelliert. Die Motivation, ein neues Strahlungstransportmodell neben den bereits existierenden, guten Modellen zu entwickeln, sowie die Anforderungen an das spezielle Design dieses Modells, die durch die MIPAS-Mission definiert werden, werden in Teil I des vorliegenden Handbuches erläutert. In Teil II stellen wir den Algorithmus in physikalischer Formulierung zusammen, auf dem KOPRA basiert, um einen Überblick zu liefern, welche physikalischen Aspekte in KOPRA eingeschlossen sind. In den weiteren Teilen (III bis X) werden die Lösungsansätze und ihre Realisierung für die wesentlichen Problemstellungen in den Unterpunkten der Strahlungstransportmodellierung dargestellt. Im einzelnen sind dies die Erstellung des geophysikalischen Modells und der Schichteinteilung der Atmosphäre, die Modellierung des gekrümmten Sehstrahls durch die Atmosphäre, die Berechnung der Absorptionskoeffizienten auf einem optimierten Frequenzgitter, die Behandlung von "line-mixing", die Berücksichtigung der Absorption und Emission durch schwere Moleküle, die Berücksichtigung von Absorptions-, Extinktions- und Emissions-Kontinua durch gasförmige und feste Teilchen in der Atmosphäre, und die Integration der Strahlungstransportgleichung längs des Sehstrahls durch die Atmosphäre einschließlich der Berücksichtigung des Effektes durch das Zusammenbrechen des lokalen thermodynamischen Gleichgewichts (NLTE). Jedem Unterpunkt wurde ein eigener Teil des Berichts zugestanden, in dem das Problem, angefangen beim physikalischen Algorithmus, bis hin zur Realisierung, Codierung und Datenstruktur erläutert wird. In jedem Teil von KOPRA wurden Optimierungen im Konzept und/oder der Codierung verwendet, solange sie nicht auf Kosten der Genauigkeit oder der Flexibilität von KOPRA gehen, um die Eignung von KOPRA auch in einem Auswertewerkzeug für eine grosse Datenmenge und in einem automatisierten Auswerteansatz zu gewährleisten. Diese Optimierungen werden beschrieben wo immer sie erscheinen. Weiterhin werden in Teil XI zwei umfangreichere Optimierungen, nämlich die Modellierung der Voigt-Funktion mit Hilfe eines beschleunigten Humlicek-Algorithmus, und die optimierte Auswertung der Planck-Funktion beschrieben. Teil XII beschreibt die Modellierung der instrumentellen Antwortfunktion hinsichtlich der (apodisierten) instrumentellen Linienform und des Effekts des endlichen Gesichtsfelds, und zwar beides sowohl für die Anwendung auf das MIPAS/ENVISAT Instrument mit seinem außergewöhnlichen Design als auch für Instrumente mit Standard-Design.

Die folgenden Teile befassen sich mit verschiedenen Erweiterungen und Zusätzen zum reinen Strahlungstransport-Algorithmus und mit Analysen der Leistungsfähigkeit von KOPRA. Teil XIII beschreibt die Umsetzung der Berechnung von quasi-analytischen Ableitungen nach relevanten Parametern simultan mit der Strahlungstransportrechnung, sowie die Schnittstelle zu einem Inversionsalgorithmus zur Ableitung geophysikalischer und instrumenteller Parameter aus den Spektren. In Teil XIV wird die optimierte Auswahl von Genauigkeitsparametern, die vom Benutzer definiert werden können, hinsichtlich Genauigkeit und Rechenzeit untersucht. Eine ausführliche Validierungsstudie gegen das wohlbekannte und zuverlässige RFM (Reference Forward Model) der Universität Oxford wird in Teil XV beschrieben. Die

Auswirkung der besonderen, nicht in anderen Modellen verfügbaren Modellierungsmöglichkeiten von KOPRA auf die Reduzierung des zu erwartenden Ableitungsfehlers wurde in einer Studie untersucht, die schließlich zu einer *a posteriori* Rechtfertigung führte, welche atmosphärischen Effekte in KOPRA modelliert werden sollten. Diese Studie ist in Teil XVI beschrieben. Die Teile XVII bis XIX erläutern die Architektur von KOPRA, seine Installation, und enthalten Listen von benötigten Eingabedateien sowie seiner Module, Unterprogramme und der wichtigsten Variablen. Im letzten Teil, Teil XX, wird die graphische Benutzerschnittstelle "kopragui" beschrieben, die das Erstellen von Eingabe-Dateien, die Überprüfung, Modifikation und graphische Ausgabe von Vertikalprofilen von Eingabedaten, den Start von KOPRA-Läufen und die graphische Ausgabe von erzeugten Spektren erlaubt.

Abstract

The Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA)

The Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA) is a FORTRAN90 computer code for atmospheric radiative transfer modelling in the mid-infrared spectral range. It has been developed as self-standing algorithm including all relevant physics from the troposphere to the thermosphere as well as the instrument specific response function of the MIPAS/ENVISAT experiment besides other more standard ones. The motivation to design a new radiative transfer code besides good existing ones and the requirements to the design given by the MIPAS/ENVISAT mission are explained in Part I of this handbook. In Part II we provide the high level physical algorithm KOPRA is based on, in order to give an overview on the physical aspects included in KOPRA. In the following Parts (III to X) the solutions and their realizations for the relevant problems within the sub-tasks of radiative transfer modelling will be presented. In particular, these are the set-up of the geophysical model and the stratification of the atmosphere, the atmospheric ray path modelling, the calculation of absorption coefficients on an optimized wavenumber grid, the treatment of line-mixing, the consideration of absorption and emission of heavy molecules and of continua caused by gaseous constituents and solid particles, and the radiative transfer integration along the line of sight including the treatment of effects caused by non-local thermodynamic equilibrium (NLTE). Each sub-task has been given an individual part of the report where the problem is described starting from the high level algorithm description and going down to the realization, coding and data structure. In each part of KOPRA, optimizations have been taken into account as long as they are not on the cost of accuracy or flexibility, in order to provide a tool suitable for the analysis of numerous data in an automated retrieval set-up. These optimizations are described wherever they appear. In Part XI, two major optimizations, namely the modelling of the Voigt function by an accelerated Humlicek algorithm, as well as the optimized evaluation of the Planck function, are described. Part XII explains the modelling of the instrumental response function in terms of (apodized) instrumental line shape and the effect of finite field of view, both for the particular MIPAS/ENVISAT design details, and for more standard-designed instruments.
The following Parts deal with various extensions and add-ons to the pure radiative transfer forward algorithm and studies of the analysis of KOPRA's performance. In Part XIII the implementation of the calculation of quasi-analytical derivatives simultaneously with the radiative transfer integration and KOPRA's interface to a retrieval algorithm is described. In Part XIV the optimised choise of user-defined accuracy parameters in terms of accuracy versus computing time is analysed. An extensive validation exercise has been performed versus the renowned and reliable RFM (Reference Forward Model) from Oxford University, which is described in Part XV. The impact of specific modelling features of KOPRA on the retrieval error has been studied which led to an *a posteriori* justification of the modeling choices for KOPRA. This study is documented in Part XVI. Part XVII to XIX describe the architecture of KOPRA and its installation, and give a listing of the required input files and of its modules, subroutines and variables. In the last Part, XX, the graphical user interface (GUI) kopragui is explained, which allows to set up the input file, check and plot vertical profiles of input data, start KOPRA runs and plot the output spectra.

Contents

Introduction: Motivation and Requirements

G.P. Stiller and T. von Clarmann

Abstract: In this paper we present the motivation to develop another radiative transfer code, namely KOPRA, besides the existing ones and the requirements set up by the MIPAS/ENVISAT mission in terms of observation scenarios, instrument characteristics, and retrieval strategy which were the drivers for the design details of KOPRA.

Retrieval of atmospheric state parameters from remote measurements depends on the accurate modeling of atmospheric radiative transfer. This is in particular true for limb emission spectrometry experiments like the MIPAS experiments [1, 2, 3] where a lot of demanding physics is involved in radiative transfer. Although a number of good radiative transfer codes are available [4, 5, 6, 7, 8, 9, 10], after some general preliminary considerations [11] we decided to design and develop a new code specifically suited for the data analysis of the space-borne Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) experiment [12] which is going to be launched on ESA's Environmental Satellite 1 (ENVISAT-1), for the following reasons: Due to global and altitudinal coverage and observation geometry, the MIPAS experiment sets up a number of specific requirements with respect to modeling of geophysical conditions which are not all fulfilled by the codes available; the modeling of the very specific instrumental response of MIPAS is a mandatory requirement which is not at all or not rigorously provided by other codes; the radiative transfer code must be integrated in an automated retrieval system being able to handle very high data amounts which sets up specific needs on the performance of the code and on the controlling options; and a code written by our own can be better designed in terms of flexibility and structure and easier adopted to future needs than a foreign one.

Therefore the Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA) has been developed. KOPRA is planned to be applied during data analysis within a dedicated scientific level-2 data processor to be developed and installed at IMK and within the MIPAS/ENVISAT processor at DFD/Oberpfaffenhofen. It models the mid-IR radiative transfer through the entire atmosphere (from the boundary layer to about 200km altitude) and considers - to our knowledge - all effects relevant in this spectral and altitude range. It is therefore suited for the data analysis of MIPAS which is going to be launched on ESA's Environmental Satellite 1 (ENVISAT-1) into a polar orbit in about 800 km altitude. MIPAS is designed to be a limb-viewing emission FTIR spectrometer with 0.025 cm^{-1} spectral resolution, covering the mid infrared from 685 cm^{-1} to 2410 cm^{-1} (14.6 - 4.15 μm) by five spectral channels. Observation modes supported are a rearward looking mode with possible azimuth sweep angles from 75^o to 110^o and a sideward looking mode in anti-sun direction with possible azimuth sweep angles from 160^o to 190^o. The elevation pointing range of the instrument results in scans with achievable tangent heights from about 5 to 150 km [13]. The field-of-view of MIPAS is about 30 km in horizontal direction and 0.9 mrad in vertical direction, resulting in about 3 km width at the tangent point, depending on tangent altitude. The standard limb sampling step width will be 3 km, which results in an averaging of air masses along the line-of-sight over about 400 km. The sampling rate is about 500 km along-track and about 2800 km across-track (at the equator). Thus, the observations will cover the complete globe for day, night, and twilight conditions from the middle troposphere to the thermosphere.

The viewing direction will be close to meridional which requires to take into account the oblateness of the Earth for calculation of the line-of-sight, including refraction in a non-spherical atmosphere with horizontal density inhomogeneities. In this viewing direction, horizontal inhomogeneities in temperature, pressure, and constituent abundances may occur often, for example when crossing with the line-of-sight the vortex edge, the boundary of the innertropical convergence zone, or the terminator. Cirrus clouds in the upper troposphere, or, during polar night observations, polar stratospheric clouds may be in the line of sight. This requires modeling of particle-caused continuum-like extinction in the mid-IR spectrum. Observation of the lower atmosphere down to the middle troposphere sets up the requirement of very accurate modeling of the absorption coefficients which has to cover all effects caused by high pressure, like pressure-broadening in combination of Doppler broad-

ening (Voigt line shape) and pressure-shift of the molecular transitions, line-mixing (in particular for Q-branches of ro-vibrational bands of CO_2, but also for other branches and molecules), duration-of-collision effects in the far line wings (water vapor and CO_2 "continua"), and a careful selection of lines contributing to the signal from outside of the observed spectral interval. In the lower to middle atmosphere a rather big number of trace species with high molecular weight contribute to the overall emission signal of the atmosphere. The ro-vibrational transitions of these species are too dense to be modeled line-by-line as it is common for other molecules. Therefore methods to utilize absorption cross-section spectra provided by several molecular spectroscopy labs and provided for a (sometimes rather small) number of pressure/temperature conditions had to be developed. Scientific questions to be investigated on the basis of the retrieved data further make it desirable to handle isotopomeric species of molecules (for example H_2O, HDO, and $^{18}H_2O$; CH_4 and CH_3D; the symmetric and asymmetric isotopomers of ozone $^{16}O_3$, $^{16}O^{16}O^{18}O$, and $^{16}O^{18}O^{16}O$) independently of each other and thus not to constrain their atmospheric concentration by a common profile. In the middle to upper atmosphere non-LTE (non-local thermodynamic equilibrium) population of the molecular states will be a major concern for emission sounders. Therefore the radiative transfer code must be able to handle the non-LTE effects in the radiance signal of the atmosphere, provided information on the non-LTE population of the molecular states is available. In this region, strong horizontal gradients of geophysical state parameters may also have a considerable impact on the observed spectrum and thus need to be modeled within KOPRA. In order to model the atmospheric signal as it is observed by the MIPAS instrument, the instrument response in terms of spectral resolution and apodized instrumental line shape (AILS), field-of-view (FOV) integration of the atmospheric radiance profile, and finite-FOV effects on the ILS has to be considered. Within a retrieval procedure, the radiative transfer code has to provide much more than the simulation of spectra.
In general terms a multi-geometry radiative transfer problem can be written linearily as

$$Y = \mathbf{K}X \tag{1}$$

where Y is a vector of form

$$(y_{1,1} \cdots y_{1,mmax(1)}, y_{2,1} \cdots y_{2,mmax(2)} \cdots y_{nmax,1} \cdots y_{nmax,mmax})^\tau,$$

containing spectra of $mmax(n)$ gridpoints, related to measurement geometry $n \leq nmax$; $\mathbf{K}$ is the $\sum_{n=1}^{nmax} mmax(n) \times imax$ Jacobian, containing the partial derivatives $\partial y_{n,m}/\partial x_i$; X is the imax-dimensional vector containing atmospheric and instrumental parameters. The retrieval problem then can be written as

$$X = (\mathbf{K}^\tau\mathbf{K} + \mathbf{R})^{-1}\mathbf{K}^\tau(Y_{measured} - Y_{calculated}) \tag{2}$$

where $Y_{measured}$ is a set of measurements, while $Y_{calculated}$ is the related set of spectra calculated on the basis of the best available estimate of X. $\mathbf{R}$ is a regularization parameter. Eq. 2 can easily be rewritten in a recursive manner, reflecting iterative processing in order to take account for nonlinearities in radiative transfer.
Random error estimation follows the formalism

$$\mathbf{S_x} = (\mathbf{K}^\tau\mathbf{S_y}^{-1}\mathbf{K})^{-1} \tag{3}$$

where $\mathbf{S}_x$ and $\mathbf{S}_y$ are the covariance matrices of the retrieval parameters and the measurement, respectively, while the mapping ΔX of uncertainty Δw of a priori information w on the retrieval of state parameter X can be approximated as

$$\Delta X = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T(\frac{\partial y_{1,1}}{\partial w} \ldots \frac{\partial y_{nmax,mmax}}{\partial w})\Delta w \tag{4}$$

The requirement to the forward model is to provide at each iteration the set of spectra $Y_{calculated}$, the Jacobian $\mathbf{K}$, and, at the last iteration, the partials $\partial y_{n,m}/\partial w_j$ for all $jmax$ parameters to be considered in the systematic error assessment, while a standard forward code not run in environment of a retrieval code has to provide only once a single spectrum and no derivatives. $\mathbf{R}$ and $Y_{measured}$ are of relevance only in the inversion code. From the viewpoint of optimizing interfaces care has to be taken that:

1. Partial derivatives shall be calculated internally rather than by successive calls of the forward code for incremented x-values, whenever possible.

2. For calculation of derivatives, the vector spaces involved shall be matched. In the following some examples are given:

   (a) if the vertical profiles of state parameters are sampled finer inside the forward code than in the retrieval, inter-layer constraints have to be supported by the forward module.

   (b) an option has to be provided whether abundances of different isotopes of a molecule shall be treated as one parameter or as several independent parameters.

   (c) constraints between populations of energy levels have to be activatable and deactivatable

   (d) spectroscopic data have to be includable and excludable in/from the vector X of variable parameters.

3. Obvious behavior of partial derivatives shall be implemented in a hard-wired way; e.g. partial derivatives of spectral radiances of a spectrum of tangent altitude z (i.e. lower edge of the field of view) with respect to atmospheric parameters at altitudes below z always are zero.

4. Systematic error estimation needs partial derivatives of spectral radiances with respect to much more parameters $\partial y_{n,m}/\partial w$ than inversion. These partial derivatives obviously are needed only at the last iteration rather than at each iteration.

5. Redundant calculation of quantities shall be avoided whenever possible. This applies to

   (a) ray-tracing and airmass calculations which can be reused during several iterations;

   (b) absorption coefficients can be reused for other geometries and iterations;

Within KOPRA, the link to such a retrieval concept with high flexibility is given by the analytical calculation of spectral derivatives with respect to an in principle unlimited number of parameters, with a used-defined and completely flexible set-up of geophysical parameter vectors and by providing the options to run KOPRA in appropriate modi. The applicability in automated data processing is reflected in KOPRA's highly efficient and optimized calculation of atmospheric absorption cross

sections, its user-defined accuracy adjustment and the ability to avoid recalculation of redundant quantities when imbedded in a data processing loop. The user-defined accuracy-parameters control the calculation of absorption cross sections, the layering of the atmosphere, the mass integration along the ray path, and the accuracy of field of view and instrumental line shape modeling. However, the optimal choice of the accuracy-control parameters is not always obvious, and may depend on the actual case under investigation. This problem is accounted for in a dedicated study in order to allow optimized performance within the automated quasi-operational processing of MIPAS data.

With all these features, KOPRA reflects the requirements set up by this mission in terms of the instrument design details, the observation scenarii, the link to a retrieval concept with high flexibility, and the applicability within automated data processing. In this report, we will describe the algorithm of radiative transfer behind the coding of KOPRA, and the realization of this algorithm down to the data structure with respect to a variety of aspects in the various modules of KOPRA. We further add a study on the optimal choice of accuracy-control parameters. Although we attempted to include all physics known to be relevant for application to MIPAS in our data analysis, we were aware that some simplifications may be allowed which, however, should be made on basis of *a posteriori* decisions on the relevance of these effects by means of the full assessment of the forward modeling error and its mapping on the retrieval error. For this reason, the forward model error and its mapping on over-all retrieval error due to discarding these individual physical processes and properties of the atmosphere and the instrument has been assessed for a number of example cases. This study will serve as a guideline for designing the real MIPAS/ENVISAT data analysis. Finally we provide an installation guide with description of the line data format and an example of the complete set of input files necessary to run KOPRA.

Bibliography

[1] H. Fischer and H. Oelhaf, "Remote sensing of vertical profiles of atmospheric trace constituents with MIPAS limb-emission spectrometers," *Appl. Opt.* **35**(16), pp. 2787–2796, 1996.

[2] H. Fischer and H. Oelhaf, "Remote sensing of vertical profiles of atmospheric trace constituents with MIPAS limb emission spectrometers," in *Optical Remote Sensing of the Atmosphere and Clouds, Beijing, China, 15–17 September 1998*, J. Wang, B. Wu, T. Ogawa, and Z. Guan, eds., *Proc. SPIE* **3501**, pp. 42–46, 1998.

[3] H. Fischer, "Remote sensing of atmospheric trace gases," *Interdisc. Sci. Rev.* **18**(3), pp. 185–191, 1993.

[4] S. A. Clough, F. X. Kneizys, E. P. Shettle, and G. P. Anderson, "Atmospheric radiance and transmittance: FASCOD2," in *Proceedings of the Sixth Conference on Atmospheric Radiation*, pp. 141–146, Am. Meteorol. Soc., Williamsburg, Va., May 1986.

[5] D. P. Edwards, "Atmospheric transmittance and radiance calculations using line–by–line computer models," in *Modelling of the Atmosphere*, L. S. Rothman, ed., *SPIE proceedings* **928**, pp. 94–116, 1988.

[6] P. Morris, A. Darbyshire, and A. Dudhia, "MIPAS–development of a reference forward algorithm for the simulation of MIPAS atmospheric limb emission spectra – detailed design document," tech. rep., European Space Agency, 1997. Report of ESA Contract 11886/96/NL/GS.

[7] M. Ridolfi, B. Carli, M. Carlotti, A. Dudhia, J.-M. Flaud, M. Höpfner, P. E. Morris, P. Raspollini, G. P. Stiller, and R. J. Wells, "An optimized forward and retrieval model for MIPAS near real time data processing," in *Optical Remote Sensing of the Atmosphere and Clouds, Beijing, China, 15–17 September 1998*, J. Wang, B. Wu, T. Ogawa, and Z. Guan, eds., *Proc. SPIE* **3501**, pp. 170–185, 1998.

[8] L. J. Gordley, B. T. Marshall, and D. Allen Chu, "LINEPAK: Algorithms for modeling spectral transmittance and radiance," *J. Quant. Spectrosc. Radiat. Transfer* **52**(5), pp. 563–580, 1994.

[9] S. A. Clough, M. J. Iacono, and J.-L. Moncet, "Line-by-line calculations of atmospheric fluxes and cooling rates: Application to water vapor," *J. Geophys. Res.* **97**, pp. 15,761–15,785, 1992.

[10] S. A. Clough and M. J. Iacono, "Line-by-line calculation of atmospheric fluxes and cooling rates, 2, Application to carbon dioxide, ozone, methane, nitrous oxide, and the halocarbons," *J. Geophys. Res.* **100**, pp. 16,519–16,535, 1995.

[11] H. Fischer, M. Carlotti, T. v. Clarmann, M. Endemann, J. Flaud, C. Muller, H. Oelhaf, M. Rast, G. P. Stiller, and R. Zander, "MIPAS Data Processing and Algorithm Development (DPAD) Subgroup Report," tech. rep., ESTEC, Noordwijk, 1994.

[12] M. Endemann and H. Fischer, "Envisat's high–resolution limb sounder: MIPAS," *ESA bulletin* **76**, pp. 47–52, 1993.

[13] "Envisat, MIPAS An instrument for atmospheric chemistry and climate research." ESA Publications Division, ESTEC, P. O. Box 299, 2200 AG Noordwijk, The Netherlands, SP-1229, 2000.

Analytical expressions for modeling of radiative transfer and instrumental effects in KOPRA

S. Zorn, T. von Clarmann, G. Echle, B. Funke,
F. Hase, M. Höpfner, H. Kemnitzer,
M. Kuntz, and G. P. Stiller

Abstract: This paper reports the analytical expressions of radiative transfer and typical instrumental effects the Karlsruhe Optimized and Precise Radiative Transfer Algorithm (KOPRA) is based on. It contains formal expressions of the integrated version of the equation of radiative transfer in the atmosphere, the source function for local thermodynamic equilibrium as well as non-local thermodynamic equilibrium conditions, transmission, slant path column densities of molecules, ray tracing, absorption and extinction coefficients, line intensities as function of temperature, line shape functions, instrument line shape of an interferometer, numerical apodization, aperture effects, and field of view consideration. On this formalism the numeric treatment of spectral modeling in KOPRA is based.

1 Introduction

Computational treatment of radiative transfer processes in the atmosphere requires numeric treatment of many processes. Nevertheless the analytic expression behind the numerics actually coded is important to make a code such as the Karlsruhe Optimized and Precise Radiative Transfer Algorithm (KOPRA) understandable. The intention of this paper is presentation of these analytical expressions, such that interrelations of variables used in this program become traceable. This paper is organized as follows: First radiative transfer physics is discussed in a top to bottom manner, starting with the radiative transfer equation, and consecutively breaking it down to more detail. Second, the degradation functions of a Fourier transform spectrometer with finite field of view as well as numerical apodization are discussed.

2 Radiative Transfer Equation

The following integrated version of Chandrasekhar's [1] radiative transfer equation is used here:

$$S_\Theta(\nu, l_{obs}) = S_\Theta(\nu, l_o)\,\tau(\nu, l_{obs}, l_o) + \int_{l_{obs}}^{l_o} J(\nu, l)\,\sigma_{a,total}^{Vol}(\nu, l)\,\tau(\nu, l_{obs}, l)\,dl \quad (1)$$

where

S_Θ	spectral radiance for the viewing angle Θ
J	source funktion (see Section 2.2)
$\tau(\nu, l_1, l_2)$	transmission between l_1 and l_2 for wavenumber ν (see Section 2.3)
dl	path element (see Section 2.1)
$\sigma_{a,total}^{Vol}$	absorption coefficient including gases and aerosols (per volume) (see Section 2.3.1)
l_{obs}	position of the observer
l_o	position of background radiative source
ν	wavenumber
l	path coordinate

2.1 Ray Tracing

The integration of Eq. 1 is performed along the line of sight of the instrument, which is, through atmospheric refraction, not a straight line. This integration is performed not analytically but numerically for thin atmospheric layers characterized by constant representative state parameters which are derived from the state parameter profiles by mass-weighted integration. In a medium of smoothly varying refractive index $n(z)$, there is refraction as soon as the gradient of $n(z)$ is not parallel to the raypath. After traveling the infinitesimal distance dl, the angular difference between the original and the new direction $d\theta$ is [2]

$$d\theta = dl \times |\vec{e_s} \dot{\nabla} n_b| / n_b \quad (2)$$

where $\vec{e_s}$ is the component of unity in direction perpendicular to the old direction and the gradient of refractive index. The the length of each path segment l in layer j used for integration of Equation 4 then is

$$l_j = \int dl \quad (3)$$

The slant path column amount m_g of a species g in the layer j is

$$m_{g,j} = \int_{l_1}^{l_2} \rho_g(l)\,dl \quad (4)$$

where the particle density ρ_g of species g is calculated as

$$\rho_g(l) = C_{V,g} \cdot \frac{N_{avo}}{R} \cdot \frac{p(l)}{T_{kin}(l)} \quad (5)$$

where N_{avo} is the Avogadro constant, R the universal gas constant, $C_{V,g}$ the volume mixing ration of species g, $T_{kin}(l)$ kinetic temperature at the position l, and p is pressure.

Mass-weighted state parameters are

$$p_{average,g,j} = \frac{1}{m_{g,j}} \int p(l)\rho_g(l)dl \tag{6}$$

and

$$T_{average,g,j} = \frac{1}{m_{g,j}} \int T(l)\rho_g(l)dl \tag{7}$$

This approach supports straight forward consideration of ellipsoidal earth shape, where the local radius R_E is

$$R_E(\Psi) = \sqrt{\frac{a_E^2 \cdot b_E^2}{b_E^2 \cdot \cos\Psi^2 + a_E^2 \cdot \sin\Psi^2}} \tag{8}$$

where

$$\tan\Psi = \left(\frac{b_E}{a_E}\right)^2 \tan\Phi \tag{9}$$

with geocentric latitude Ψ, geographic latitude Φ, major semiaxis a_E, and minor semiaxis b_E.
The refractive index n_b is calculated from the wavenumber-dependent refractive index for dry air n_{b_o}, referring to reference state parameters $T_o = 288.16$ K and $p_o = 1013.25$ hPa, as

$$n_b(l) = \sqrt{\frac{2\,k\,p(l) + T(l)}{T(l) - k\,p(l)}} \tag{10}$$

where

$$k = \frac{T_o\,(n_{b_o}^2 - 1)}{p_o\,(n_{b_o}^2 + 2)} \tag{11}$$

The wavenumber dependence of the refractive index is

$$(n_{b_o} - 1) \cdot 10^6 = 83.4213 + \frac{24060.3}{130 - 10^8\nu^2} + \frac{159.97}{38.9 - 10^8\nu^2} \tag{12}$$

where ν must be given in cm^{-1}.

2.2 The Source Function

2.2.1 Local Thermodynamic Equilibrium

In the case of local thermodynamic equilibrium (LTE), the source function J_{LTE} is the Planck function B of the kinetic temperature T_{kin} of the emitting medium:

$$J^{LTE}(\nu,l) = B(\nu,T_{kin}(l)) \tag{13}$$

The Planck function as a function of frequency f is

$$B(f,T) = \frac{2f^2}{c^2} \cdot \frac{hf}{\exp\left(\frac{hf}{k_B T}\right) - 1} \tag{14}$$

where h is the Planck constant and c the velocity of light. The Planck function as a function of wavenumber ν can be deduced from Eq. 14 by post-differentiation of the Planck function by $\frac{d\nu}{df}$:

$$B(\nu,T_{kin}(l)) = \frac{2hc^2\nu^3}{\exp\left(\frac{hc\nu}{k_B T_{kin}(l)}\right) - 1} \tag{15}$$

2.2.2 Non-Local Thermodynamic Equilibrium (NLTE)

Here we follow the formalism of [3] and [4]. For a single line, the source function J in case of non-local thermodynamic equilibrium (NLTE) can be written as:

$$J^{NLTE}(\nu, l) = \frac{2hc^2\nu^3}{\frac{r_1}{r_2}\exp\left(\frac{hc\nu}{k_B T_{kin}(l)}\right) - 1} \tag{16}$$

where r_1 and r_2 are the ratios of populations between the general (NLTE) and LTE cases for the lower (1) and upper (2) states of transition n. This ratio of populations between the general (NLTE) and LTE cases for the state m can also be written as

$$r_m = f_Q(l)\,\exp\left(-\frac{E_{vib,m}}{k_B}\left(\frac{1}{T_{Vib,m}(l)} - \frac{1}{T_{kin}(l)}\right)\right) \tag{17}$$

where m is the vibrational state under consideration, $E_{vib,m}$ its energy level, T_{Vib} the related vibrational temperature, and $f_Q(l)$ is a correction factor to the vibrational partition sum given by

$$f_Q(l) = \frac{\sum_m g_m \exp\left(-\frac{hcE_{vib,m}}{k_B T_{kin}(l)}\right)}{\sum_m g_m \exp\left(-\frac{hcE_{vib,m}}{k_B T_{vib,m}(l)}\right)}. \tag{18}$$

Here, g_m represents the degeneracy factor of state m. For superimposed lines, the NLTE source function is written as

$$J^{NLTE}(\nu, l) = \beta(\nu, l) \cdot B(\nu, T_{kin}(l)) \tag{19}$$

$$\beta(\nu, l) = \frac{\sum_{gn} r_{2,gn}(l)\,\sigma_{a,gn}^{LTE}(\nu, l)\,m_g(l)}{\sum_{gn} \alpha_{gn}(\nu, l)\,\sigma_{a,gn}^{LTE}(\nu, l)\,m_g(l)} \tag{20}$$

$$\alpha_{gn} = \frac{\sigma_{a,gn}^{NLTE}(\nu, l)}{\sigma_{a,gn}^{LTE}(\nu, l)} \tag{21}$$

where

$\sigma_{a,gn}^{LTE}$	absorption coefficient (LTE) (see Eq. 27)
$\sigma_{a,gn}^{NLTE}$	absorption coefficient (NLTE) (see Eq. 28)

2.3 Transmittance

The spectral atmospheric transmission $\tau(\nu, l_1, l_2)$ between two points l_1 and l_2 on the line of sight is calculated as follows:

$$\tau(\nu, l_1, l_2) = \exp -\left\{ \int_{l_1}^{l_2} \sigma_{a,gas}^{Vol}(\nu, l)\,dl + \underbrace{\int_{l_1}^{l_2} \sigma_{e,aerosol}^{Vol}(\nu, l)\,dl}_{\text{Aerosol}} \right\} \tag{22}$$

$$= \exp - \left\{ \sum_{g=1}^{G} \int_{l_1}^{l_2} \sigma_{a,g}(\nu, l) \, \frac{\partial m_g(l)}{\partial l} \, dl + \underbrace{\int_{l_1}^{l_2} \sigma_{e,aerosol}^{Vol}(\nu, l) \, dl}_{\text{Aerosol}} \right\}$$

where

$\sigma_{e,aerosol}^{Vol}$	aerosol extinction coefficient (from external Mie calculation)
$\sigma_{a,gas}^{Vol}$	volume absorption coefficient for gases (see Eq. 24)
$\sigma_{a,g}$	absorption coefficient of the gas g (see Eqs.25)
m_g	slant path column amount of gas g (see Eq. 4)
dl	path element
G	number of gases taken into account

2.3.1 Absorption and -Extinction Coefficients

2.3.1.1 Local Thermodynamic Equilibrium

The aerosol extinction coefficient (per volume) is

$$\sigma_{e,aerosol}^{Vol}(\nu, l) = \sigma_{a,aerosol}^{Vol}(\nu, l) + \sigma_s(\nu, l) \tag{23}$$

The gas absorption coefficient is (per volume) is

$$\sigma_{a,gas}^{Vol}(\nu, l) = \sum_{g=1}^{G} \sigma_{a,g}(\nu, l) \cdot \rho_g(l) \tag{24}$$

The absorption coefficient of species g is

$$\sigma_{a,g}(\nu, l) = \sum_{n=1}^{N_g} \sigma_{a,gn}(\nu, l) + (\text{cross-sections}) + (\text{continua}) \tag{25}$$

where

$\sigma_{e,aerosol}^{Vol}$	aerosol extinction coefficient (per volume)
$\sigma_{a,aerosol}^{Vol}$	aerosol absorption coefficient (per volume)
σ_s	scattering coefficient (from external Mie calculations)
$\sigma_{a,gn}$	absorption coefficient of the transition n of species g (see Eqs. 27 / 28)
ρ_g	particle density of species g (see Eq. 5)
G	number of species under consideration
N_g	number of transitions of species g

Scattering is considered here in terms of extinction only. No additional source term of the type

$$J_s(\nu, l) = \frac{\Omega_o}{4\pi} \int \int p(\Omega, \Omega') S(\Omega, \Omega') \, d\Omega' \tag{26}$$

is considered here. The absorption coefficients $\sigma_{a,gn}^{LTE}$ of transition n of species g in LTE is written

$$\sigma_{a,gn}^{LTE}(\nu, l) = A_{gn}(T_{kin}(l), \nu) \cdot \Phi_{gn}(\nu, p(l), T_{kin}(l)) \tag{27}$$

where

A_{gn}	line intensity of transition n of species g (see Eq.2)
Φ_{gn}	profile function of transition n of species g at position l (see Chapter 2.3.1.4)

2.3.1.2 Non-Local Thermodynamic Equilibrium

In case of NLTE, the absorption coefficient $\sigma_{a,gn}^{NLTE}$ of transition n of species g is derived from the absorption coefficient in LTE by a correction factor α:

$$\sigma_{a,gn}^{NLTE}(\nu,l) = \alpha_{gn}(\nu,l) \cdot \sigma_{a,gn}^{LTE}(\nu,l) \tag{28}$$

Correction factor α, which is one in case of LTE, is calculated as

$$\alpha_{gn}(\nu,l) = \frac{\sigma_{a,gn}^{NLTE}(\nu,l)}{\sigma_{a,gn}^{LTE}(\nu,l)} = \frac{r_1 - r_2\Gamma}{1-\Gamma}. \tag{29}$$

where

$$\Gamma = \frac{g_1 n_2}{g_2 n_1} = \exp\left(-hc\nu/k_B T_{kin}\right) \tag{30}$$

$$r_1 = \frac{n_1^{NLTE}}{n_1^{LTE}} \quad ; \quad r_2 = \frac{n_2^{NLTE}}{n_2^{LTE}} \tag{31}$$

and where

r_1, r_2	ratios of populations between the general (NLTE) and LTE cases for the lower (1) and upper (2) state (see 17)
n_1^{NLTE}	population of the lower state (NLTE)
n_1^{LTE}	population of the lower state (LTE)
n_2^{NLTE}	population of the upper state (NLTE)
n_2^{LTE}	population of the upper state (LTE)
g_1, g_2	level statistical weights for the lower (1) and upper (2) state

2.3.1.3 Line Intensities

The intensity A_{gn} of line n of species g is

$$A_{gn}(T_{kin},\nu) = A_{gn}(T_o',\nu) \cdot \frac{Q(T_o')}{Q(T_{kin})} \frac{e^{-\frac{E_{gn}''}{k_B T_{kin}}}}{e^{-\frac{E_{gn}''}{k_B T_o'}}} \frac{1 - e^{-\frac{hc\nu}{k_B T_{kin}}}}{1 - e^{-\frac{hc\nu}{k_B T_o}}} \tag{32}$$

where

k_B	Boltzmann constant
E_{gn}''	lower state energy of transition n (see Eq. (3))
$Q(T)$	LTE total internal partition function evaluated at T (see Eq.(4))
$Q(T_o')$	LTE total internal partition function at T_o'
$A_{gn}(T_o')$	line intensity at reference temperature T_o'
T_{kin}	kinetic temperature
T_o'	reference temperature (296 K)
ν	wavenumber, usually approximated as $\nu_{o,n}$
$\nu_{o,n}$	central wavenumber of transition n

The lower state energy E'' is calculated from the lower state energy e'' in units of wavenumbers as given in spectroscopic databases by:

$$E''_{gn} = c\, e''_{gn}\, h \tag{33}$$

The total internal partition function $Q(T)$ describes the temperature dependence of the line intensity of a transition. In LTE it is

$$Q(T) = \sum_{j=o}^{\infty} g_j \exp\left(\frac{E_j}{k_B T}\right) \tag{34}$$

where j is the rotational-vibrational-state, g the factor of degeneration, and E the energy of level j. It is approximated as

$$Q(T) = a_0 + a_1 T + a_2 T^2 + a_3 T^3 \tag{35}$$

where a_0, a_1, a_2, and a_3 are pretabulated coefficients [5].

2.3.1.4 Profile Function

The Doppler profile function describes the statistical distribution of frequency shifts due to thermal motion:

$$\Phi_{D,n}(\nu, T) = \frac{1}{\alpha_{D,n}} \sqrt{\frac{\ln 2}{\pi}}\, e^{-\ln 2\left(\frac{\nu-\nu_{o,n}}{\alpha_{D,n}}\right)^2} \tag{36}$$

where

$$\alpha_{D,n}(T) = \frac{\nu_{o,n}}{c} \sqrt{\frac{2 k_B T \ln 2}{M}} \tag{37}$$

and where $\alpha_{D,n}$ is the Doppler halfwidth, M the molecular mass, and $\nu_{o,n}$ central wavenumber of transition n.
The Lorentzian profile function $\Phi_{L,n}$ of transition n, centered at $\nu_{o,n}$, describes pressure broadening, under assumption of infinitesimally short impact between collision partners and elastic collisions:

$$\Phi_{L,n}(\nu, p, T) = \frac{1}{\pi} \left(\frac{\alpha_{L,n}}{(\nu - \nu_{o,n})^2 + \alpha_{L,n}^2}\right) \tag{38}$$

The actual Lorentzian halfwidth $\alpha_{L,n}$ is calculated for the actual temperature T and actual pressure p from the related reference Lorentzian halfwidth $\alpha_{L_o,n}$ measured at reference pressure p_o^L and reference temperature T_o^L

$$\alpha_{L,n}(p, T) = \alpha_{L_o,n}(p_o^L, T_o^L) \frac{p}{p_o^L} \left(\frac{T_o^L}{T}\right)^{\gamma_n} \tag{39}$$

The coefficient of temperature dependence of the halfwidth γ_n, which, following classical collisional theory, should be 0.5, may deviate from this theoretical value; therefore laboratory measurements are used whenever available.
The Lorentzian halfwidth is composed of a self-broadening term and a foreign broadening term:

$$\alpha_{L_o,n} = (1 - C_{V,g})\, \alpha_{L_o,n}^{foreign} + C_{V,g}\, \alpha_{L_o,n}^{self} \tag{40}$$

where

$\alpha_{L_o,n}$	reference Lorentzian halfwidth of transition n
$\alpha_{L_o,n}^{foreign}$	reference foreign-broadening Lorentzian halfwidth of transition n of species g
$\alpha_{L_o,n}^{self}$	reference self-broadening Lorentzian halfwidth of transition n of species g
$C_{V,g}$	volume mixing ratio of species g

The Voigt function $\Phi_{V,n}$ is the convolution of Doppler and Lorentzian broadening:

$$\Phi_{L,n}(\nu,p,T) \otimes \Phi_{D,n}(\nu,T) \tag{41}$$

$$\Phi_{V,n}(\nu,p,T) = \frac{1}{\alpha_{D,n}} \sqrt{\frac{\ln 2}{\pi}} \frac{y}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{(x-t)^2 + y^2}\, dt \tag{42}$$

where

$$y = \frac{\alpha_{L,n}}{\alpha_{D,n}} \sqrt{\ln 2} \tag{43}$$

and

$$x = \left(\frac{\nu - \nu_{o,n}}{\alpha_{D,n}}\right) \sqrt{ln2} \tag{44}$$

where $\alpha_{D,n}$ and $\alpha_{L,n}$ are Doppler and Lorentzian halfwidths, respectively, and $\nu_{o,n}$ is the central wavenumber of transition n.

2.3.1.5 Line-mixing

In the case of overlapping lines broadened by collisions, line mixing effects have to be taken into account [6]. The Lorentzian line profile shape is then modified:

$$\Phi_{L,n}^{lm}(\nu,p,T) = \frac{\tilde{A}(T)}{A(T)} \frac{1}{\pi} \frac{\tilde{\alpha}_{L,n} + (\nu - \tilde{\nu}_{o,n})\tilde{Y}_n(T,p)}{(\nu - \tilde{\nu}_{o,n})^2 + (\tilde{\alpha}_{L,n})^2} \tag{45}$$

where

$\Phi_{L,n}^{lm}$	pressure broadened profile function of transition n under consideration of line-mixing
$\tilde{A}(T)$	line mixing modified line intensity
$A(T)$	line intensity (see Eq. 2)
$\tilde{\nu}_{o,n}$	line-mixing modified central wavenumber
$\tilde{\alpha}_{L,n}$	line mixing modified Lorentzian halfwidth
$\tilde{Y}_n$	line-mixing coefficient

The determination of the quantities signed by a tilde is discussed in Part VI: 'Line mixing'. Within the Rosenkranz approximation valid for low atmospheric pressures, the quantities $\tilde{A}(T)$, $\tilde{\nu}_{o,n}$, and $\tilde{\alpha}_{L,n}$ are given by their unmodified values.
This modified shape of pressure-broadened lines maps into the Voigt line shape as follows:

$$\Phi_{V,n}^{lm}(\nu,p,T) = \sqrt{\frac{\ln 2}{\pi}} \frac{1}{\alpha_{D,n}} \left[K(\tilde{x}_n,\tilde{y}_n) + \tilde{Y}_n(p,T)\, L(\tilde{x}_n,\tilde{y}_n)\right] \tag{46}$$

$$K(\tilde{x},\tilde{y}) = \frac{\tilde{y}}{\pi}\int_{-\infty}^{\infty}\frac{e^{-t^2}}{(\tilde{x}-t)^2+\tilde{y}^2}\,dt \tag{47}$$

$$L(\tilde{x},y) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{(\tilde{x}-t)e^{-t^2}}{(\tilde{x}-t)^2+\tilde{y}^2}\,dt \tag{48}$$

$$\tilde{x} = \left(\frac{\nu-\tilde{\nu}_{o,n}}{\alpha_{D,n}}\right)\sqrt{ln2} \tag{49}$$

$$\tilde{y} = \frac{\tilde{\alpha}_{L,n}}{\alpha_{D,n}}\sqrt{\ln 2} \tag{50}$$

where

$\Phi_{V,n}^{lm}$	Voigt profile function of transition n under consideration of line-mixing
$\alpha_{D,n}$	Doppler halfwidth (see Eq. 37)
$\tilde{Y}_n$	line-mixing coefficient

2.3.1.6 χ-Faktor

The impact of time-dependence of collisions on the line shape is considered by empirical χ-factors [7]

$$\Phi_\chi(\nu) = \chi\cdot\Phi_V(\nu,p,T) \tag{51}$$

where

Φ_V	Voigt-profile function [cm] (see 42)
χ	χ-factor

For CO_2, asymmetric χ-factors have to be considered. χ-factors for CO_2 self-broadening as published by Refs. [8][9][10] are compiled in Table 1. χ-factors for CO_2 foreign-broadening by N_2 as published by Refs. [11][12][10] and used in KOPRA are compiled in Table 2. The χ-factor for CO_2 foreign broadening by O_2 as published by [11] and [12] is compiled in Table 3.

These χ-factors are added weighted by the partial pressures p_{CO_2}, p_{N_2} and p_{O_2} of contributing gases CO_2, N_2 and O_2:

$$\frac{p_{CO_2}}{p_{total}}\cdot\chi_{CO_2} + \frac{p_{N_2}}{p_{total}}\cdot\chi_{CO_2\text{-}N_2} + \frac{p_{O_2}}{p_{total}}\cdot\chi_{CO_2\text{-}O_2} \tag{52}$$

where

$$p_{total} = p_{CO_2} + p_{N_2} + p_{O_2} \tag{53}$$

Table 1: χ-factor for CO_2 self-broadening; $\Delta\nu = \nu - \nu_{o,n}$, $\nu_{o,n}$ is the central wavenumber of transition n, and K_1 is the modified Bessel function of the second kind

Temperature: 296 K	
$0 \leq \lvert\Delta\nu\rvert \leq 3\ cm^{-1}$	$\chi_{CO_2} = 1$
$3 \leq \lvert\Delta\nu\rvert \leq 10\ cm^{-1}$	$\chi_{CO_2} = 1.470\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{7.782}\right)$
$10 \leq \lvert\Delta\nu\rvert \leq 120\ cm^{-1}$	$\chi_{CO_2} = 0.535\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{36.535}\right)$
$\lvert\Delta\nu\rvert \geq 120\ cm^{-1}$	
$\Delta\nu < 0$	$\Delta\nu > 0$
$\chi_{CO_2} = 0.889\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{31.627}\right)$	$\chi_{CO_2} = 0.220\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{50.063}\right)$
Temperature: 218 K	
$0 \leq \lvert\Delta\nu\rvert \leq 3\ cm^{-1}$	$\chi_{CO_2} = 1$
$3 \leq \lvert\Delta\nu\rvert \leq 10\ cm^{-1}$	$\chi_{CO_2} = 1.240319\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{13.9296}\right)$
$10 \leq \lvert\Delta\nu\rvert \leq 140\ cm^{-1}$	$\chi_{CO_2} = 0.68\ \frac{\lvert\Delta\nu\rvert}{28}\ K_1(\frac{\lvert\Delta\nu\rvert}{28})$
$\lvert\Delta\nu\rvert \geq 140\ cm^{-1}$	
$\Delta\nu < 0$	$\Delta\nu > 0$
$\chi_{CO_2} = 10.04385\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{21.233306}\right)$	$\chi_{CO_2} = 0.345\ \exp\left(-\frac{\lvert\Delta\nu\rvert}{43.448}\right)$

Table 2: χ-factor for CO_2-broadening by N_2

Temperature: 296 K	
$0 \leq \lvert\Delta\nu\rvert \leq 0.5\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 1$
$0.5 \leq \lvert\Delta\nu\rvert \leq 20\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 1.064\ \exp(-0.1235\ \lvert\Delta\nu\rvert)$
$20 \leq \lvert\Delta\nu\rvert \leq 50\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 0.125\ \exp(-0.0164\ \lvert\Delta\nu\rvert)$
$50 \leq \lvert\Delta\nu\rvert \leq 140\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 0.146\ \exp(-0.0196\ \lvert\Delta\nu\rvert)$
$\lvert\Delta\nu\rvert \geq 140\ cm^{-1}$	
$\Delta\nu < 0$	$\Delta\nu > 0$
$\chi_{CO_2\text{-}N_2} = 1.8593\ \exp(-0.03776\ \lvert\Delta\nu\rvert)$	$\chi_{CO_2\text{-}N_2} = 0.146\ \exp(-0.0196\ \lvert\Delta\nu\rvert)$
Temperatur: 238 K	
$0 \leq \lvert\Delta\nu\rvert \leq 5\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 1$
$5 \leq \lvert\Delta\nu\rvert \leq 22\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 1.968\ \exp(-0.1354\ \lvert\Delta\nu\rvert)$
$22 \leq \lvert\Delta\nu\rvert \leq 50\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 0.160\ \exp(-0.0214\ \lvert\Delta\nu\rvert)$
$\lvert\Delta\nu\rvert \geq 50\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 0.162\ \exp(-0.0216\ \lvert\Delta\nu\rvert)$
Temperature: 193 K	
$0 \leq \lvert\Delta\nu\rvert \leq 9\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 1$
$9 \leq \lvert\Delta\nu\rvert \leq 23\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 3.908\ \exp(-0.1514\lvert\Delta\nu\rvert)$
$23 \leq \lvert\Delta\nu\rvert \leq 28\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 0.207\ \exp\left(-3.778\ 10^{-3}\lvert\Delta\nu\rvert\right)$
$28 \leq \lvert\Delta\nu\rvert \leq 50\ cm^{-1}$	$\chi_{CO_2\text{-}N_2} = 0.219\ \exp(-0.0276\lvert\Delta\nu\rvert)$
$\Delta\nu < 0$	$\Delta\nu > 0$
$50 \leq \lvert\Delta\nu\rvert \leq 130\ cm^{-1}$	$50 \leq \lvert\Delta\nu\rvert \leq 135\ cm^{-1}$
$\chi_{CO_2\text{-}N_2} = 0.20894\ \exp(-0.026694\lvert\Delta\nu\rvert)$	$\chi_{CO_2\text{-}N_2} = 0.146\ \exp(-0.0196\lvert\Delta\nu\rvert)$
$130 \leq \lvert\Delta\nu\rvert \leq 160\ cm^{-1}$	$\lvert\Delta\nu\rvert > 135\ cm^{-1}$
$\chi_{CO_2\text{-}N_2} = 2.824997\ \exp(-0.0467266\lvert\Delta\nu\rvert)$	$\chi_{CO_2\text{-}N_2} = 1.164\ \exp(-0.035\lvert\Delta\nu\rvert)$
$\lvert\Delta\nu\rvert > 160\ cm^{-1}$	
$\chi_{CO_2\text{-}N_2} = 1.192053\ \exp(-0.0413334\lvert\Delta\nu\rvert)$	

Table 3: χ-factor for CO_2-broadening by O_2

Temperature: 296 K	
$0 \leq \lvert\Delta\nu\rvert \leq 3\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 1$
$3 \leq \lvert\Delta\nu\rvert \leq 8\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 3.341\ \exp(-0.4021\ \lvert\Delta\nu\rvert)$
$8 \leq \lvert\Delta\nu\rvert \leq 50\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.155\ \exp(-0.0179\ \lvert\Delta\nu\rvert)$
$50 \leq \lvert\Delta\nu\rvert \leq 70\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.238\ \exp(-0.0266\ \lvert\Delta\nu\rvert)$
$70 \leq \lvert\Delta\nu\rvert \leq 140\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.146\ \exp(-0.0196\ \lvert\Delta\nu\rvert)$
$\lvert\Delta\nu\rvert \geq 140\ cm^{-1}$	
$\Delta\nu < 0$	$\Delta\nu > 0$
$\chi_{CO_2\text{-}O_2} = 1.8593\ \exp(-0.03776\ \lvert\Delta\nu\rvert)$	$\chi_{CO_2\text{-}O_2} = 0.146\ \exp(-0.0196\ \lvert\Delta\nu\rvert)$
Temperature: 238 K	
$0 \leq \lvert\Delta\nu\rvert \leq 5\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 1$
$5 \leq \lvert\Delta\nu\rvert \leq 22\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 1.968\ \exp(-0.1354\ \lvert\Delta\nu\rvert)$
$22 \leq \lvert\Delta\nu\rvert \leq 50\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.160\ \exp(-0.0214\ \lvert\Delta\nu\rvert)$
$\lvert\Delta\nu\rvert \geq 50\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.162\ \exp(-0.0216\ \lvert\Delta\nu\rvert)$
Temperature: 193 K	
$0 \leq \lvert\Delta\nu\rvert \leq 11\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 1$
$11 \leq \lvert\Delta\nu\rvert \leq 23\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 7.908\ \exp(-0.188\ \lvert\Delta\nu\rvert)$
$23 \leq \lvert\Delta\nu\rvert \leq 35\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.122\ -\ (7.539\ 10^{-4}\ \lvert\Delta\nu\rvert)$
$35 \leq \lvert\Delta\nu\rvert \leq 50\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.349\ \exp(-0.0369\ \lvert\Delta\nu\rvert)$
$50 \leq \lvert\Delta\nu\rvert \leq 135\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 0.129\ \exp(-0.0170\ \lvert\Delta\nu\rvert)$
$\lvert\Delta\nu\rvert \geq 135\ cm^{-1}$	$\chi_{CO_2\text{-}O_2} = 1.455\ \exp(-0.0350\ \lvert\Delta\nu\rvert)$

3 Field of View and Instrumental Line Shape

While previous sections deal with radiative processes in the atmosphere which are independent of the observing system, in the following the influence of the observing system is discussed.

3.1 Field of View

The observed spectral radiance $S_{\Theta_0}^{FOV}(\nu, l_{obs})$ is the convolution of the radiances related to observation angles Θ with the weighting function W describing the FOV.

$$S_{\Theta_0}^{FOV}(\nu, l_{obs}) = \int_{-\Theta_{max}}^{\Theta_{max}} S_{\Theta}(\nu, l_{obs}) W(\Theta - \Theta_0) d\Theta \tag{54}$$

where

$W(\Theta - \Theta_0)$	weighting function related to field of view (FOV), as integrated for horizontal stripes $d\Theta$
$S_{\Theta_0}^{FOV}$	FOV-convolved spectral radiance for viewing angle Θ_0 $\left[\frac{W}{cm^2 sr \cdot cm^{-1}}\right]$
FOV	field of view
l_{obs}	position of the observer $[cm]$
ν	wavenumber (wavenumber) $[cm^{-1}]$
S_{Θ}	spectral radiance for infinitesimal viewing angle Θ
Θ	viewing angle
Θ_{max}	maximum angle covered by FOV

3.2 Instrumental Line Shape

The instrument line shape (ILS) is a function the atmospheric spectrum has to be convolved with in order to simulate what is seen by the instrument:

$$S_{\Theta}^{AILS}(\nu, l_{obs}) = \frac{\int_{-\infty}^{+\infty} S_{\Theta_0}^{FOV}(\nu', l_{obs}) \cdot AILS(\nu - \nu')\, d\nu'}{\int_{-\infty}^{+\infty} AILS(\nu - \nu')\, d\nu'} \tag{55}$$

where

S_{Θ}	spectral radiance for viewing angle Θ
$AILS$	apparatus function incl. numerical apodization (apodized instrumental line shape) (see Part XII: 'Transformation of irradiated to measured spectral distribution due to finite spectral resolution and field of view extent of a Fourier transform spectrometer')
$\int_{-\infty}^{+\infty} AILS(\nu-\nu')\, d\nu'$	normalization (if necessary; in KOPRA $\int_{-\infty}^{+\infty} AILS(\nu-\nu')\, d\nu' = 1$)

KOPRA determines a single sided ($x \geq 0$) and complex-valued modulation efficiency with symmetric real and antisymmetric imaginary parts. The AILS is the Fourier transform of the modulation efficiency. The modulation efficiency $M(x)$includes

the numeric apodization function, the modulation loss due to self apodization, the linear modulation loss and the phase error.

$$AILS(\nu) = \int_{-\infty}^{\infty} M(x)e^{-i\nu x}dx \tag{56}$$

with

$$M(x) = M_{resolution}(x) \times M_{numeric}(x) \times M_{self}(x) \times M_{linear}(x) \times M_{phase}(x) \tag{57}$$

- resolution

  The interferogram is restricted to maximal optical path difference L

  $M_{resolution}(x) = 1$ if $|x| \leq L$
  $M_{resolution}(x) = 0$ else

- apodization function

  1. sinc
     $M_{numeric} = 1$
  2. triangle
     $M_{numeric} = 1 - |x|/L$
  3. Hamming
     $M_{numeric} = 0.53856 + 0.46144 \times \cos(\pi \times x/L)$
  4. Blackmann-Harris 3-term
     $M_{numeric} = 0.42323 + 0.49755 \times \cos(\pi \times x/L) + 0.07922 \times \cos(2 \times \pi \times x/L)$
  5. Blackmann-Harris 4-term
     $M_{numeric} = 0.35875 + 0.48829 \times \cos(\pi \times x/L) + 0.14128 \times \cos(2 \times \pi \times x/L) + 0.01168 \times \cos(3 \times \pi \times x/L)$
  6. Norton-Beer weak
     $M_{numeric} = 0.384093 - 0.087577 \times \left(1 - (x/L)^2\right) + 0.703484 \times \left(1 - (x/L)^2\right)^2$
  7. Norton-Beer medium
     $M_{numeric} = 0.152442 - 0.136176 \times \left(1 - (x/L)^2\right) + 0.983734 \times \left(1 - (x/L)^2\right)^2$
  8. Norton-Beer strong
     $M_{numeric} = 0.045335 - 0.554883 \times \left(1 - (x/L)^2\right)^2 + 0.399782 \times \left(1 - (x/L)^2\right)^4$

- self apodization

  The interferometer has finite acceptance angle. It can be shown, that optical path difference depends on the inclination of the wavefront versus the optical axis. In case of homogeneously illuminated circular internal FOV of semidiameter one finds that this leads to an additional loss of modulation:

  $M_{self} = \frac{\sin(\pi \times \Delta\nu \times x)}{\pi \times \Delta\nu \times x}$ with $\Delta\nu = 0.5 \times \nu \times \alpha^2$

  Note that the self apodization (and thereby the resulting AILS) depends on spectral position ν. The additional modulation loss describes the consequences of the finite acceptance angle not completely. In addition, the spectral abscissa is scaled by $0.5 \times (\cos(\alpha) + 1)$.

- linear modulation loss

  This term is used to model the width of the imperfect AILS.

  $M_{linear} = 1 - (1 - a) \times x/L$

  the factor a gives the modulation efficiency at maximal path difference vs ideal instrument.

- phase error

  This term is used to model the asymmetry of the imperfect AILS. The phase error φ is given in radians.

  $M_{phase} = \frac{e^{-i\varphi}}{\cos(\varphi)}$

  The norm of the AILS is fixed by the real part of $M(0)$. The denominator ensures the norm to be unity.

Since the real valued AILS is the Fourier transform of the complex-valued modulation efficiency, the latter is symmetric in the real part and antisymmetric in the imaginary part. Due to this symmetry, the AILS is fully determined by a single sided modulation efficiency interferogram.

Appendix A

Symbols and Acronyms

a_E	major semi-axis
a_0, a_1, a_2, a_3	tabulated coefficients for calculation of total internal partition function (LTE)
b	focal length
b_E	minor semi-axis
c	velocity of light
dl	path element (Eq. 3)
e''	lower state energy (from HITRAN)
f_Q	NLTE correction factor to the partition sum
g	index specifying species
g_1, g_2	level statistical weights
h	Planck constant
j	rotational quantum number
k_B	Boltzmann constant
l	path coordinate
l_{obs}	position of the observer
l_o	position of background emitter
m	vibrational state
m_g	slant path column amount of species g (Eq. 4)
n	index specifying transitions
n_b	refractive index (Eq. 10)
n_{b_o}	wavenumber-dependent refractive index of dry air (Eq. 12)
n_1^{NLTE}	population of the lower state (NLTE)
n_1	population of the lower state (LTE)
n_2^{NLTE}	population of the upper state (NLTE)
n_2	population of the upper state (LTE)
p	pressure
p_{obs}	pressure at observer altitude
p_{CO_2}, p_{N_2}, p_{O_2}	CO_2, N_2 and O_2 partial pressures
p_o	reference pressure for calculation of refractive index n_B ($p_o = 1013.25$ hPa)
p_o^L	reference pressure for calculation of Lorentzian halfwidth
r	altitude coordinate + radius of earth
r_A	aperture radius
r_1, r_2	ratio of populations between the general (NLTE) and LTE cases (Eq. 17)
z	altitude coordinate
z_A	upper boundary of atmosphere

z_{obs}	observer altitude
A_{gn}	line intensity of transition n of species g (Eq. 2)
$A_{gn}(T_o')$	line intensity of transition n of species g at reference temperature T_o'
$AILS$	apodized instrumental line shape (apparatus function incl. numerical apodization)(Eq. 56)
B	Planck function (Eq. 15)
$C_0, C_1, .., C_4$	coefficients for calculation of apodization function
$C_{V,g}$	volume mixing ratio of species g
$E_{vib,m}$	energy of vibrational state m
E_{gn}''	lower state energy of transition n of species g
FOV	field of view
$FWHM$	full width at half maximum (nominal spectral resolution)
G	number of relevant species
J	source function
J^{LTE}	source function in case of LTE (Eq. 13)
J^{NLTE}	source function in case of NLTE (Eq. 16)
K_1	modified Bessel function of the second kind
LTE	local thermodynamic equilibrium
M	molecular mass
$M_{resolution}$	modulation efficiency function related to resolution
$M_{numeric}$	modulation efficiency function related to numerical apodization
M_{self}	modulation efficiency function related to self apodization
M_{linear}	modulation efficiency function related to linear modulation efficiency loss
M_{phase}	modulation efficiency function related to phase error
N	apodizing function number
N_{avo}	Avogadro constant
N_g	number of transitions of species g
$NLTE$	non-local thermodynamic equilibrium
OPD_{max}	max. optical path difference
$Q(T)$	LTE total internal partition function evaluated at T (Eq. 4)
$Q(T_o)$	LTE total internal partition function at $T_o = 296$ K
R	universal gas constant
R_E	radius of Earth (Eq. 8)
S_Θ	spectral radiance for viewing angle Θ (Eq. 1)
S^{FOV}	spectral radiance convolved by field of view (Eq. 54)
S^{AILS}	spectral radiance convolved by apodized instrumental line shape (Eq. 55)
T_{kin}	kinetic temperature
T_{obs}	temperature at observer altitude
T_{Vib}	vibrational temperature (in LTE: $T_{Vib} = T_{kin}$)
T_o	reference temperature for calculation of refractive index n_B ($T_o = 288.16$ K)
T_o'	reference temperature ($T_o' = 296$ K)
T_o^{lm}	temperature for calculation of line-mixing coefficients ($T_o^{lm} = 200$ K)
T_o^L	reference temperature for calculation of Lorentzian halfwidth ($T_o^L = 296$ K)
$W(\Theta - \Theta_0)$	weighting function of the FOV
Y_n	line-mixing coefficient (Eq. 45)
$\alpha_{D,n}$	Doppler halfwidth (transition n) (Eq. 37)
$\alpha_{L,n}$	Lorentzian halfwidth (Eq. 39)

$\alpha_{L_o,n}$	reference Lorentzian half width of transition n (Eq. 40)
$\alpha_{L_o,n}^{foreign}$	reference Lorentz half width of transition n
$\alpha_{L_o,n}^{self}$	reference Lorentz half width of transition n
χ	χ-Faktor
χ_{CO_2}	χ-faktor for self-broadening of CO_2 lines (Eq.18)
$\chi_{CO_2\text{-}N_2}$	χ-Factor for N_2-broadening of CO_2 lines (Eq.19)
$\chi_{CO_2\text{-}O_2}$	χ-Faktor for O_2-Verbreiterung von CO_2 (Eq.20)
γ_n	coefficient of temperature dependence of air-broadened halfwidth
ν	wavenumber
$\nu_{o,n}$	central wavenumber of transition n
$\Delta\nu$	wavenumber interval
ρ_g	number density of species g (Eq. 5)
σ_a^{Vol}	volume absorption coefficient (Eq. 24)
$\sigma_{a,g}$	absorption coefficient of species g (Eq. 25)
$\sigma_{a,gn}$	absorption coefficient of transition n of species g
$\sigma_{a,gn}^{LTE}$	$\sigma_{a,gn}$ in LTE (Eq. 27)
$\sigma_{a,gn}^{NLTE}$	$\sigma_{a,gn}$ in NLTE (Eq. 28)
σ_e^{Vol}	volume extinction coefficient (Eq. 23)
$\sigma_{e,\text{aerosol}}^{vol}$	aerosol extinction coefficient (from external Mie calculation
σ_s	scattering coefficient (from external Mie calculation)
$\tau(\nu, l_1, l_2)$	transmission between l_1 and l_2 at wavenumber ν (Eq. 22)
Φ	geographic latitude
Φ_{gn}	profile function of line n of species g
$\Phi_{D,n}$	Doppler profile function of transition n (Eq. 36)
$\Phi_{L,n}$	Lorentzian profile function of transition n (Eq. 38)
$\Phi_{L,n}^{lm}$	Lorentzian profile function of transition n under consideration of line-mixing (Eq. 45)
$\Phi_{V,n}$	Voigt profile function of transition n (Eq. 42)
$\Phi_{V,n}^{lm}$	Voigt profile function of transition n under consideration of line-mixing (Eq. 46)
Ψ	geocentric latitude (Eq. 9)
Θ	viewing angle

Bibliography

[1] S. Chandrasekhar, *Radiative Transfer*, Dover Publ., New York, 1960.

[2] F. Hase and M. Höpfner, "Atmospheric ray path modeling for radiative transfer algorithms," *Appl. Opt.* **38**(15), pp. 3129–3133, 1999.

[3] D. P. Edwards, M. López-Puertas, and M. Á. López-Valverde, "Non–local thermodynamic equilibrium studies of the 15 μm bands of CO_2 for atmospheric remote sensing," *J. Geophys. Res.* **98**(D8), pp. 14,955–14,977, 1993.

[4] D. P. Edwards, M. López-Puertas, and M. G. Mlynczak, "Non–local thermodynamic equilibrium limb radiance from O_3 and CO_2 in the 9–11 μm spectral region," *J. Quant. Spectrosc. Radiat. Transfer* **52**(3–4), pp. 389–407, 1994.

[5] R. R. Gamache, R. L. Hawkins, and L. S. Rothman, "Total internal partition sums for atmospheric molecules in the temperature range 70 – 2005 K: Atmospheric linear molecules," *J. Mol. Spectrosc.* **142**, pp. 205–219, 1990.

[6] B. Funke, "Modellierung des line–mixing Effekts in Q–Zweigen von CO_2–Banden für die Berechnung von atmosphärischen Infrarotspektren," Master's thesis, Institut für Meteorologie und Klimaforschung, Universität Karlsruhe, Forschungszentrum Karlsruhe, 1995.

[7] S. A. Clough, F. X. Kneizys, and R. W. Davies, "Line shape and the water vapor continuum," *Atmospheric Research* **23**, pp. 229–241, 1989.

[8] R. Le Doucen, C. Cousin, C. Boulet, and A. Henry, "Temperature dependence of the absorption in the region beyond the 4.3–μm band head of CO_2. 1: Pure CO_2 case," *Appl. Opt.* **24**(6), pp. 897–906, 1985.

[9] V. Menoux, R. Le Doucen, and C. Boulet, "Line shape in the low–frequency wing of self–broadened CO_2 lines," *Appl. Opt.* **26**(3), pp. 554–562, 1987.

[10] V. Menoux, R. Le Doucen, J. Boissoles, and C. Boulet, "Line shape in the low frequency wing of self– and N_2–broadened ν_3 CO_2 lines: temperature dependence of the asymmetry," *Appl. Opt.* **30**(3), pp. 281–286, 1991.

[11] C. Cousin, R. Le Doucen, C. Boulet, and A. Henry, "Temperature dependence of the absorption in the region beyond the 4.3–μm band head of CO_2. 2: N_2 and O_2 broadening," *Appl. Opt.* **24**(22), pp. 3899–3907, 1985.

[12] V. Menoux, R. Le Doucen, and C. Boulet, "Line shape in the low–frequency wing of N_2 and O_2 broadened CO_2 lines," *Appl. Opt.* **26**(23), pp. 5183–5189, 1987.

Geophysical model and atmospheric layering

M. Höpfner

Abstract: The quantities and variables which determine the atmospheric state in KOPRA are given. It is described how the hydrostatic equilibrium is calculated and how horizontal gradients are taken into account. Finally, the different possibilities and criteria for the atmospheric layering determining the radiative transfer discretization are presented.

1 Introduction

The radiative transfer modeling in KOPRA is based on a layer-by-layer approach, i.e. during the ray-tracing partial gas columns and Curtis-Godson means (of temperature, pressure, non-LTE/LTE population ratios) are determined for each path segment. A path segment is a part of the line-of-sight inside one layer, i.e. between two successive equi-altitude surfaces (=levels). While more levels increase the accuracy of a radiative transfer calculation the run-time depends in first order linearly on their number. Hence, a trade-off between run-time and accuracy adapted to each experiment has to be done during the discretization of the problem (the so-called 'layering').
This approach implies two important steps: (1) the calculation of atmospheric state quantities (temperature, pressure, vmr, non-LTE/LTE population ratios) at arbitrary locations in the atmosphere and (2) the layering itself. These two subjects will be described separately in the following sections.

2 Geophysical model

How does KOPRA get to know the value of an atmospheric state quantity for a location (latitude, longitude, altitude) in the atmosphere? The answer is: it is calling one of the functions **give_p, give_T, give_vmr,give_Tvib**[1] with the geolocation as arguments. Hence, inside these functions the transformation (in the simplest case interpolation) from discrete profile values (or parameters) to continuous profiles is computed. The reason why we speak generally of transformations and not only of interpolations is the distinction between input profiles and retrieval parameters (see paragraph 4 in Part XIII: 'Derivatives and interface to the retrieval'). In the following we describe the interpolation procedures for input profiles with values of atmospheric quantities at fixed altitude levels. (The transformation rules for retrieval parameters are to be defined by the user.)

2.1 Input profiles

All input profiles belong to one geolocation: **inprof(1)%lon, inprof(1)%lat.** They are stored in the following variables (e.g. for altitude levels i):

altitude	**inprof(i)%alt**
pressure	**inprof(i)%p**
temperature	**inprof(i)%T**
vmr for each species j	**inprof(i)%vmr(j)**
vibrational temperature for each state j of one isotope species k	**inprof(i)%Tvib(j,k)**
continuum absorption coefficient for each microwindow j	**inprof(i)%aerabs(j)**
continuum scattering coefficient for each microwindow j	**inprof(i)%aersca(j)**

Information about horizontal gradients for each altitude level (only for p, T, non-LTE/LTE population ratios, vmr) is stored in:

(gradient profiles along latitude circles, positive to east):

inprof(i)%latgrad%p
inprof(i)%latgrad%T
inprof(i)%latgrad%Tvib(k,j)
inprof(i)%latgrad%vmr(j)

(gradient profiles along longitude circles, positive to south):

inprof(i)%longrad%p
inprof(i)%longrad%T
inprof(i)%longrad%Tvib(j,k)
inprof(i)%longrad%vmr(j)

These variables are set up in the subroutines:

input_pTprof@input_m	**input_vmrprof@input_m**
input_pTgradprof@input_m	**input_vmrgradprof@input_m**
input_contprof@input_m	**input_Tvibprof@input_m**
input_Tvibgradprof@input_m	

Herein the profiles are read from the files (one extra file for each subroutine; the

[1] though new versions of KOPRA handle non-LTE/LTE population ratios instead of vibrational temperatures the name give_Tvib and variables with Tvib are not changed internally

file names are defined in *kopra.inp*; standard names are: *pt.prf, vmr.prf, ptgra.prf, vmrgra.prf, cont.prf, Tvib.prf, Tvibgra.prf* in directory *kopra/input/profiles/*). In these files the profiles can be on arbitrary altitude grids (one grid per file). They are interpolated/extrapolated linearly (only the pressure and pressure-gradient logarithmic) to the input level altitudes (**inprof(i)%alt**) defined in *kopra.inp*. Furthermore, for single isotopomeres of species the vmr and vmr-gradient profiles are multiplied by the isotopic abundance profile in subroutine **isomult@input_m**. (The isotopic abundances are stored in variable **mol()%iso()%abunprof()** which is initialized in **input_isoprof@input_m** by reading from file; standard name: *kopra/input/profiles/isoabu.prf*.)
After this initialization of the **inprof()%**-variables the user can select if the p/T/altitude profiles should be recalculated taking into account hydrostatic equilibrium (**make_hydroequi@input_m, press@varsub_m, alti@varsub_m, g@varsub_m**). To put pressures into hydrostatic equilibrium the following procedure is used:
First, the actual layer (between level i and $i+1$) is subdivided into n sub-layers of 0.1 km thickness. In a loop over the sub-levels $j = 1$ to n the pressure is determined iteratively ($j = 1$ corresponds to level i and $j = n+1$ to level $i+1$):

$$p_{j+1} = p_j \exp\left(-\frac{M_{air}(z_j)}{y(p_j,T_j)R} g(\varphi, z_j) \frac{z_{j+1} - z_j}{(T_{j+1} + T_j)/2}\right) \tag{1}$$

with z_j, z_{j+1} the sub-level altitudes [km], T_j, T_{j+1} the sub-level temperatures [K] which are interpolated linearly in altitude between the level temperatures T_i and T_{i+1}, p_j, p_{j+1} the level pressures [hPa], φ the geographical latitude, and R the gas-constant. The gravitational acceleration g is:

$$g(\varphi, z) = \left[g_0(\varphi) + \omega^2 R_c \cos^2\varphi\right] \left[\frac{R_e}{R_e + z}\right]^2 - \omega^2 \left[R_c + z\right] \cos^2\varphi \tag{2}$$

with $R_e(\varphi)$ the local distance to the earth's center, $R_c(\varphi)$ the distance to the y-axis of the ellipse along the ellipse's normal, ω the earth's angular velocity, and the empirical formula for the gravitational acceleration at ground:

$$g_0(\varphi) = 9.80616 \cdot (1 - 0.0026373 \cdot \cos 2\varphi + 0.0000059 \cdot \cos^2 2\varphi) \tag{3}$$

$$R_c(\varphi) = \frac{a}{\sqrt{1 - \left(1 - \frac{b^2}{a^2}\right) \sin^2\varphi}} \tag{4}$$

$$R_e(\varphi) = R_c(\varphi) \sqrt{1 - \left(1 - \frac{b^4}{a^4}\right) \sin^2\varphi}. \tag{5}$$

M_{air} is the molecular mass of air which is composed of the molecular masses and volume mixing ratios of H_2O, CO_2, N_2, O_2, O, and Ar:

$$\begin{aligned} M_{air}(z) \;=\; & M_{H_2O} VMR_{H_2O}(z) + M_{CO_2} VMR_{CO_2}(z) + M_{N_2} VMR_{N_2} + \\ & M_{O_2} VMR_{O_2} + M_O VMR_O + M_{Ar} VMR_{Ar}. \end{aligned} \tag{6}$$

The compressibility factor y is given by [1]:

$$y = 1 - \frac{p}{T}\left(a_0 + a_1 T + a_2 T^2 + (b_0 + b_1 T)\, VMR_{H_2O} + (c_0 + c_1 T)\, VMR_{H_2O}^2\right) + \left(\frac{p}{T}\right)^2 \left(d + e VMR_{H_2O}^2\right). \quad (7)$$

Where

$$\begin{aligned} a0 &= 1.58123 \times 10^{-6} KPa^{-1} \\ a1 &= -2.9331 \times 10^{-8} Pa^{-1} \\ a2 &= 1.1043 \times 10^{-10} K^{-1} Pa^{-1} \\ b0 &= 5.707 \times 10^{-6} KPa^{-1} \\ b1 &= -2.051 \times 10^{-8} Pa^{-1} \\ c0 &= 1.9898 \times 10^{-4} KPa^{-1} \\ c1 &= -2.376 \times 10^{-6} Pa^{-1} \\ d &= 1.83 \times 10^{-11} K^2 Pa^{-2} \\ e &= -0.765 \times 10^{-8} K^2 Pa^{-2}. \end{aligned}$$

2.2 Use of input profiles in give-functions

In the give-functions the input-profiles are interpolated to a position (indicated by latitude: lat, longitude: lon, and altitude: z) in the atmosphere.
The interpolation in altitude between two levels is performed linearly in temperature, non-LTE/LTE population ratio and volume mixing ratio and logarithmic in pressure.
The horizontal interpolation is computed in the following way:
at the position of the profile (lat_0, lon_0), also the local gradient profiles in south-direction and the gradients orthogonal to the south-direction are given in units of the atmospheric quantity per km. Then the local tangent plane at the point of the profile is constructed. The geolocation of the position to which the interpolation should be done (lat,lon) is projected (central projection outgoing from the earth's center) into this plane. In the tangent plane the interpolation of the profile to (lat,lon) at one distinct altitude z is done by:

$$\begin{aligned} y(lat, lon, z) = y(lat_0, lon_0, z) &+ dx(lat, lon) \cdot grad_x(lat_0, lon_0, z) \\ &+ dy(lat, lon) \cdot grad_y(lat_0, lon_0, z) \end{aligned} \quad (8)$$

With $grad_x$ and $grad_y$ the gradient profiles interpolated to the altitude z, $y(lat_0,lon_0,z)$ the profile value at z, and the (orthogonal) distances dx and dy:

$$\begin{aligned} dx &= \frac{R_e \arccos(z')}{\sqrt{x'^2 + y'^2}} x' \\ dy &= \frac{R_e \arccos(z')}{\sqrt{x'^2 + y'^2}} y' \end{aligned} \quad (9)$$

With the earth's radius R_e and the Cartesian co-ordinates x', y', z' of the position (lat,lon) relative to a co-ordinate system centered at the earth's center and z-axis through (lat_0,lon_0). This system is constructed outgoing from an earth centered system with z-axis through north pole and x-axis through (0°,0°): first rotate around z-axis so that the x-axis is now at (0°,lon_0) and then rotated around new y-axis

until z-axis is at (lat_0,lon_0):

$$\begin{aligned} x' &= \cos(lat) \cdot \cos(lon - lon_0) \cdot \sin(lat_0) - \sin(lat) \cdot \cos(lat_0) \\ y' &= \cos(lat) \cdot \sin(lon - lon_0) \\ z' &= \cos(lat) \cdot \cos(lon - lon_0) \cdot \cos(lat_0) + \sin(lat) \cdot \sin(lat_0) \end{aligned} \tag{10}$$

3 Layering

As stated before the layering of the atmosphere is very important for the performance (regarding quality of the results and computing time) of the code. The starting point for the layering are the so-called base-levels which are fixed. Their **n%baselev** altitudes are stored in variable **accu%basealt()** which is initialized in **input_main@input_m**. Depending on the switch **sw%baselev** :
0) the input-profile levels (**inprof()%alt**) from *kopra.inp (\$7.32)* are used exclusively, or
1) the input-profile levels (**inprof()%alt**) are used and additional levels are introduced in subroutine **make_modelgrid@modlev_m**, or
2) new base-levels are read from *kopra.inp (\$7.32)*, or
3) new base-levels are read from *kopra.inp (\$7.32)* and additional levels are introduced in subroutine **make_modelgrid@modlev_m**, or
4) the levels are set up automatically in **make_modelgrid@modlev_m**.

For cases 1), 3), and 4) starting from the base-levels KOPRA introduces additional layers depending on three criteria:
1. pressure variation between two levels,
2. temperature variation between two levels, and
3. a finer layering near tangent points.

The layering is performed by calling subroutine **make_modelgrid@modlev_m**. In detail the following procedure is adopted:

- in **grid_t_hw@ modlev_m** according to absolute temperature variation (maximum **accu%Tvar1, accu%Tvar2**) and relative pressure variation (maximum **accu%wvar**) additional levels are introduced in the middle between two successive levels in a recursive procedure. For levels below **accu%zTvar** km **accu%Tvar1** and above **accu%zTvar** km the maximum T-variation **accu%Tvar2** is used.
- in **grid_tang@modlev_m** up to **accu%upto** km above all tangent altitudes additional levels with maximum spacing of **accu%dif** km are inserted.
- in **min_distance@modlev_m** all levels which are no base-levels and which are less than **accu%dmin** km apart are deleted.
- the resulting **n%modlev** model level altitudes are stored in variable **modprof()%alt**.

Bibliography

[1] P. E. Ciddor, "Refractive index of air: new equations for the visible and near infrared", *Appl. Opt.* **35(9), 1996.**

Atmospheric raypath modeling for radiative transfer algorithms

F. Hase and M. Höpfner

Abstract: A general method for the determination of raypaths as well as resulting path segments and partial gas columns within a layered atmosphere is presented. Any singularity at the tangent point is avoided. No use is made of the gross spherical symmetry of the Earth's atmosphere. Hence, deviations from symmetry due to the Earth's oblate shape and atmospheric disturbances can be taken into account in a straightforward manner. It is additionally explained how KOPRA calculates integrated path quantities like Curtis-Godson values, path column amounts and their derivatives. The inclusion of horizontal gradients into this scheme is shown.

1 Introduction

Inference of atmospheric trace gas concentration profiles from observed infrared spectra relies on the comparison with synthetic spectra which are calculated from a set of relevant variables describing the state of the atmosphere. Their values are varied until the best possible agreement between the observed and the calculated spectra is achieved. The calculation of spectra requires first of all the determination of the refracted raypath through the atmosphere [1][2]. Since the usual set of variables consists of temperatures and gas concentrations at a number of discrete atmospheric levels, the path segments and partial gas column amounts for the enclosed layers have to be calculated.

2 Usual methods for raytracing in the Earth's atmosphere

A simple method is to refract the ray at the borders of discrete atmospheric layers. If the ray crosses the border, it is deviated according to Snell's law:

$$n_1 \times \sin\alpha_1 = n_2 \times \sin\alpha_2 \tag{1}$$

with n_1,n_2 denoting the refractive indices of the adjacent layers (assumed to be homogeneous) and α_1, α_2 denoting the angles of the light ray with the normal of the enclosed boundary. The method works reasonably well as long as the ray does not cross the boundary almost tangentially. This situation occurs in limb-sounding observations. In this case, the path segments in the layers near the tangent point become excessively large and the computational error increases. A further subdivision of the layers can be introduced around the tangent point, but this even increases the chance of just touching a layer near the tangent height under an angle of nearly 90° and, thus, undermining numerical stability. In any case, the region around the tangent point requires special attention. The usual solution is to insert a straight line around the tangent point.
It can be shown that in the case of spherical symmetry the construction of a constant of motion is possible [3][4]:

$$const = n(r) \times r \times \sin\alpha(r) \tag{2}$$

with $n(r)$ denoting the refractive index as a function of the radius and $\alpha(r)$ denoting the zenith angle. If $n(r)$ and the tangent height or any angle α to a certain value of r are given, the overall raypath is determined. In the vicinity of the tangent height care must be taken, because in the calculation of the distance traveled by the ray ds the quotient ds/dr diverges at the tangent point.
It is possible to take the ellipsoidal shape of the Earth approximately into account by using an appropriate radius of curvature determined by the tangent point's latitude and related azimuthal direction of view instead of the mean Earth radius. With the location of the tangent point given a priori, this causes no problems. However, in practice the given variables are the observer's position and the direction of view. The tangent point and related radius of curvature have to be sought iteratively by shooting back to the observer's position. If horizontal inhomogeneities of the atmosphere along the raypath are not negligible, the formulation's attraction disappears completely.
Therefore, a method free of numerical problems near the tangent point, avoiding symmetry requirements and capable to start raytracing at the observer's position with a given direction of sight is preferable.

3 Method of tangential displacement

Imagine a light ray propagating through a smoothly varying refractive index distribution $n(\vec{r})$ as represented in Fig. 1. A gradient $\nabla n(\vec{r})$ not parallel to the ray deviates it. The new direction of propagation $\vec{e}_{tn}$ lies in the plane defined by the old tangent vector $\vec{e}_t$ and $\nabla n(\vec{r})$. After traveling the infinitesimal distance ds, the angle $d\alpha$ between $\vec{e}_t$ and $\vec{e}_{tn}$ is

$$d\alpha = ds \times |\vec{e}_s \cdot \nabla n|/n \tag{3}$$

with $\vec{e}_s$ denoting the direction perpendicular to $\vec{e}_t$ in the plane defined by $\vec{e}_t$ and $\nabla n(\vec{r})$. This can be seen by constructing the tilt of a locally plane wavefront over

the infinitesimal distance of travel ds : $\vec{e}_s \cdot \nabla n$ is the component of the refractive index gradient along the wavefront and, thus, $|\vec{e}_s \cdot \nabla n|/n$ can be interpreted as the ratio of propagation velocities per unit length along the wavefront. Obviously, the wavefront is bent into the region where $n(\vec{r})$ becomes larger. The path of a light ray in any smooth distribution $n(\vec{r})$ can be followed by adding up together many sufficiently small displacements of this kind.
Due to the high expenditure needed when constructing the vector $\vec{e}_s$ for each displacement, we use the fact that $\nabla n(\vec{r})$ in the terrestrial atmosphere is an extremely small quantity. Then, $\vec{e}_{tn}$ can be constructed in the following way:
Calculate the auxiliary vector $\vec{e}_{ta}$ by

$$\vec{e}_{ta} = (n \times \vec{e}_t + ds \times \nabla n)/|n \times \vec{e}_t + ds \times \nabla n| \tag{4}$$

This vector can be used instead of $\vec{e}_{tn}$, as can be seen by multiplying equation (4) by $\vec{e}_s$:

$$\vec{e}_s \cdot \vec{e}_{ta} = ds \times (\vec{e}_s \cdot \nabla n)/|n \times \vec{e}_t + ds \times \nabla n| \tag{5}$$

The denominator is nearly $n(\vec{r})$ and, therefore, the value for the bending is in good agreement with the exact calculation. For a horizontal displacement of 100 m near the Earth's surface, the relative error in $d\alpha$ does not exceed 10^{-9}. For a nearly vertical displacement the relative error reaches 10^{-5}, but the absolute deviation becomes minute.

4 Implementation into a radiative transfer model

The starting point for the implementation of the method of tangential displacement into a radiative transfer model is the transformation of the observer coordinates (latitude, longitude, altitude) and the direction of view (elevation angle, azimuth) into Cartesian coordinates. All further raytracing operations will be performed in this coordinate system. Hence, given a point on the line of sight $\vec{r}_i = (x_i, y_i, z_i)$ and a tangent vector $\vec{e}_{t,i}$, the subsequent point $\vec{r}_{i+1}$ is:

$$\vec{r}_{i+1} = \vec{r}_i + 0.5 \times ds \times (\vec{e}_{t,i} + \vec{e}_{t,i+1}) \tag{6}$$

with:

$$\vec{e}_{t,i+1} = \frac{\vec{e}_{t,i} \times n(\vec{r}_i) + ds \times \nabla n(\vec{r}_i + 0.5 \times ds \times \vec{e}_{t,i})}{|\vec{e}_{t,i} \times n(\vec{r}_i) + ds \times \nabla n(\vec{r}_i + 0.5 \times ds \times \vec{e}_{t,i})|} \tag{7}$$

The gradient of the refractive index $n = 1 + \varepsilon$ at the location $\vec{r}_i + 0.5 \times ds \times \vec{e}_{t,i}$ is computed numerically by repeated use of a function $\varepsilon(x, y, z)$ giving the related ε for each coordinate (x, y, z).
Since radiative transfer models usually operate with predefined atmospheric layers, the crossing points of the ray with the level altitudes must be determined: Given a function $h(\vec{r})$ which returns for each Cartesian point $\vec{r}$ in the atmosphere the altitude h above the ground, the crossing point is determined by the linear approximation:

$$\vec{r}' = \vec{r}_i + (\vec{r}_{i+1} - \vec{r}_i) \times (h(\vec{r}_i) - H_j)/(h(\vec{r}_i) - h(\vec{r}_{i+1})) \tag{8}$$

with $h(\vec{r}_i) > H_j$ and $h(\vec{r}_{i+1}) < H_j$ or vice versa. H_j is the altitude of the atmospheric level j. Since $h(\vec{r})$ is nearly linear along the path over the distance ds, Eq. 8 should be a good approximation. Eq. 8 can be used iteratively (substituting $\vec{r}_{i+1}$

by $\vec{r}'$) to obtain a better estimate $\vec{r}''$, if necessary.
For limb sounding observations the determination of the tangent altitude is of interest as well. A first estimation of the tangent point is obviously $\vec{r}_{i,min}$ with $h(\vec{r}_{i,min})$ reaching a minimum. This result can be improved by parabolic interpolation taking the adjacent altitudes into account using ds as the horizontal distance between the three points. The minimum of the parabola is taken as the value for the tangent altitude.
Another important issue regarding the computation time is the choice of the increment ds. For satellite limb sounding experiments, $ds = 10$km is sufficient to obtain an accuracy of the trace gas amounts of better than 0.1%.

5 Curtis-Godson values, path column amounts and their derivatives

In this section we describe the calculation of integrated values of atmospheric state parameters for each path segment along the refracted line-of-sight. The following mean quantities are computed for each path j, each molecular species g, and in the case of non-LTE applications for each vibrational state n:

$$s_j = \int_{path_j} ds, \tag{9}$$

$$u_{jg} = \int_{path_j} \rho_g(s) ds, \tag{10}$$

$$T_{jg} = \frac{1}{u_{jg}} \int_{path_j} T(s)\rho_g(s) ds, \tag{11}$$

$$p_{jg} = \frac{1}{u_{jg}} \int_{path_j} p(s)\rho_g(s) ds, \tag{12}$$

$$r_{jgn} = \frac{1}{u_{jg}} \int_{path_j} r_{gn}(s)\rho_g(s) ds, \tag{13}$$

where s is the path length along the line-of-sight, u the partial column amount, T the kinetic temperature, p the pressure, $\rho \propto vmr_g p/T$ the number density (vmr = volume mixing ratio), and r the fractional population of each vibrational state n between non-LTE and LTE. r is connected to the vibrational temperature T_{vib} by

$$r_{gn} = f_{Q,g} \exp\left(\frac{hcE_{gn}}{k_B}\left[\frac{1}{T} - \frac{1}{T_{vib,gn}}\right]\right), \tag{14}$$

where E is the state energy, h Planck's constant, k_B Boltzmann's constant, and c the velocity of light. $f_{Q,g}$ is the non-LTE correction factor of the partition sum. Both, vibrational temperature or population ratio profiles are supported as input quantities. Since internally KOPRA only handles population ratios, vibrational temperatures will be converted at the beginning directly after read-in.
Additionally, the following derivatives of the integrated path values with respect to atmospheric retrieval parameters are determined:

$$\frac{du_{jg}}{dq_{vmr,gm}}, \frac{dT_j}{dq_{T,m}}, \frac{dp_j}{dq_{p,m}}, \frac{du_{jg}}{dq_{p,m}}, \frac{dr_{jgn}}{dq_{r,gnm}}, \tag{15}$$

where m is the index on the retrieval parameters, $q_{vmr,g}$ the retrieval parameters for vmr (or vmr gradients) of species g, q_T the retrieval parameters for temperature

(or temperature gradient), q_p the retrieval parameters for pressure, and $q_{r,gn}$ the retrieval parameters for the population ratios of state n for species g.
For the calculation of absorption coefficients and during radiative transfer in the following steps of KOPRA only the mean path quantities are needed. Therefore, by also having calculated the derivatives of these quantities with respect to the retrieval parameters the basis is provided for the flexible handling of the retrieval parameterization. I.e. that after radiative transfer the derivatives of the spectra with respect to the mean path values can be combined with (15) to construct the requested Jacobian matrix.

The integrals (9-13) are solved numerically when the end of each path segment j (i.e. the layer boundary) is reached. The steps are as following:

(a) Transformation of $\vec{r}_i = (x_i, y_i, z_i)$ into altitude $h(\vec{r}_i)$, latitude $lat(\vec{r}_i)$, and longitude $lon(\vec{r}_i)$ at each grid point i along the line-of-sight inside the actual path segment j.

(b) Calculation of atmospheric state parameters at each location $\vec{r}_i$ by calling the 'give'-functions. These functions perform the interpolation from discrete input-profiles to any location in the atmosphere.

- $T_i = give\_T(h(\vec{r}_i), lat(\vec{r}_i), lon(\vec{r}_i))$
- $p_i = give\_p(h(\vec{r}_i), lat(\vec{r}_i), lon(\vec{r}_i))$
- $vmr_{ig} = give\_vmr(h(\vec{r}_i), lat(\vec{r}_i), lon(\vec{r}_i), g)$
- $r_{ign} = give\_Tvib(h(\vec{r}_i), lat(\vec{r}_i), lon(\vec{r}_i), g, n)$

(c) Numerical integration using the grid values $T_i, p_i, vmr_{ig}, r_{ign}$ to obtain the integrated path values (9-13).

To obtain greatest flexibility in parameterization of retrieval quantities the derivatives of the integrated path quantities (15) are calculated numerically by: (for further details about derivative see Part XIII: 'Derivatives and interface to the retrieval')

(a) Change of each retrieval parameter by some increment.

(b) Recalculation of the integrated layer value with the changed parameter.

(c) Determination of the numerical derivative by using the previously determined path value for the undisturbed case.

If new parameterizations for the retrieval quantities should be implemented only the transformation rules in the 'give'-functions have to be changed and KOPRA automatically delivers the Jacobian matrix with respect to the new quantities.

6 Horizontal gradients

If the horizontal gradients are switched off (\$5.9 of the main input-file) the arguments *lat* and *lon* have no effects in the 'give'-functions and the interpolation is only performed for the altitude coordinate h.
If horizontal gradients are switched on, the 'give'-functions use the altitude profiles and the altitude gradient profiles to determine the value of the atmospheric state at the point $\vec{r}_i$ with altitude h, latitude *lat*, and longitude *lon* as described in Part III: 'Geophysical model and atmospheric layering'. Therefore, the integrated values for each path segment comprise the horizontal gradients.

azimuth	spherical Earth with gradient	elliptical Earth without gradient	elliptical Earth with gradient
airmass: $2.165 \times 10^{30}/m^2$			
0°	$+2.421\times 10^{-2}$	$+1.722\times 10^{-4}$	$+2.440\times 10^{-2}$
90°	$+5.789\times 10^{-4}$	$+1.728\times 10^{-3}$	$+2.311\times 10^{-3}$
180°	-2.386×10^{-2}	$+3.550\times 10^{-4}$	-2.352×10^{-2}
refraction angle: 0.219°			
0°	$+2.422\times 10^{-2}$	$+1.723\times 10^{-4}$	$+2.442\times 10^{-2}$
90°	$+5.799\times 10^{-4}$	$+1.729\times 10^{-3}$	$+2.316\times 10^{-3}$
180°	-2.387×10^{-2}	$+3.554\times 10^{-4}$	-2.353×10^{-2}

Table 1: Relative azimuthal variation of airmass and refraction angle along a ray starting tangentially at a height of 10km in 50° N. The azimuth is measured from south. The atmosphere is assumed to be isothermal (220 K) with a pressure at the tangent point of 265 hPa in a height of 10 km. For the elliptical Earth a minor axis of 6357 km and a major axis of 6378 km is assumed. In the spherical calculations a radius of 6370 km is used.

Derivatives of the integrated values with respect to the horizontal gradient parameters are determined exactly in the same way as the derivatives with respect to the altitude profile parameters by changing the (gradient)parameter-value, recalculating the integral and numerical differentiation.

7 Illustrative raytracing results

The presented method is especially suitable for raytracing in the presence of the ellipsoidal Earth shape and locally variable atmospheric state. To illustrate the effects, the azimuthal variation of airmass and refraction along a ray starting tangentially at a height of 10km were studied. The tangent point is located at 50° northern latitude and a southwards directed pressure gradient is constructed by vertical displacement of the isobars by a rate of 100 m/°. Such values can be found at the boundary of the polar vortex and near pronounced weather systems.
As can be seen from Table 1, the resulting effects are quite small in general. The relative variation in airmass and refraction angle are nearly the same. The variable atmosphere has a much stronger impact as compared to the Earth's oblateness. In the presence of both effects, the combined result is a linear superposition.

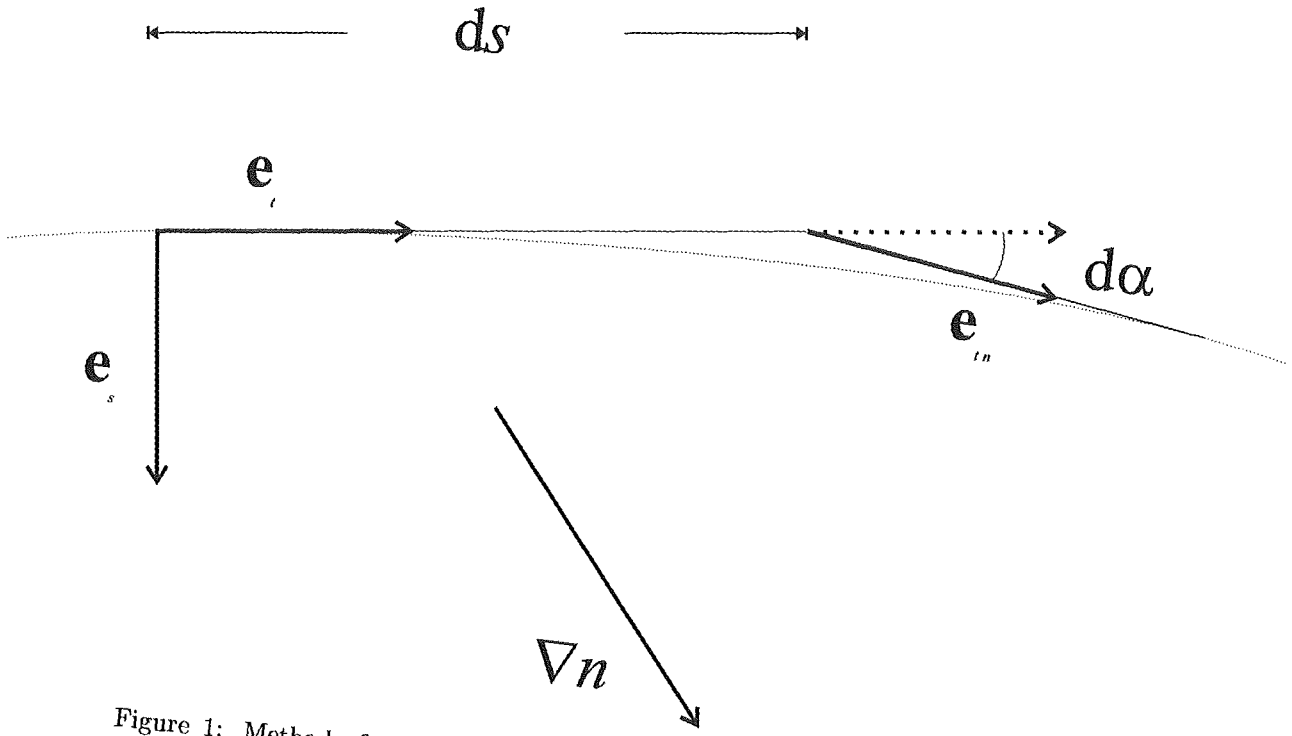


Figure 1: Method of tangential displacement: construction of the new tangent vector.

Bibliography

[1] A. I. Mahan, "Astronomical refraction - some history and theories," *Appl. Opt.* **1**, pp. 497–511, 1962.

[2] D. E. Snider, "Refractive effects in remote sensing of the atmosphere with infrared transmission spectroscopy," *J. Atmos. Sci.* **32**, pp. 2178–2184, 1975.

[3] M. Born and E. Wolf, *Principles of Optics*, Pergamon, New York, 1965.

[4] D. A. Thompson, T. J. Pepin, and F. W. Simon, "Ray tracing in a refracting spherically symmetric atmosphere" *J. Opt. Soc. Am.* **72**, pp. 1498–1501, 1982.

Absorption coefficients, line collection and frequency grid

M. Kuntz

Abstract: An efficient algorithm of evaluating absorption coefficients line-by-line for the purpose of integrating the radiative transfer equation with emphasis on the Earth's atmosphere is presented. The main problems associated with such an algorithm are concerned with line shape, the treatment of lines outside the spectral range of interest, and the way in which spectral sampling is performed. These are considered together with methods improving speed of computation in general.

1 Introduction

The absorption coefficient is a function of temperature, frequency (wavenumber), pressure, gas volume mixing ratio and constants associated with all contributing line transitions. The contribution to absorption due to a single transition l centered at ν_l can be written as the product of the number density N_s of the molecular species s to which the spectral line belongs, the line intensity S_l, and a line shape factor $f_l(\nu, \nu_l)$. The monochromatic absorption coefficient at frequency ν of a given species thus is

$$\sigma_{s,l}(\nu) = N_s \sum_l S_l \, f_l(\nu, \nu_l), \tag{1}$$

where the summation is performed over all relevant lines. (To simplify notation we do not include all dependent variables in this and other equations). Eq. 1 refers to the absorption coefficient under consideration of local thermodynamic equilibrium (LTE). Corrections to account for non-LTE are performed a posteriori within the radiative transfer modeling (see Part X:'Non-LTE and radiative transfer').
Note that the calculation of absorption coefficients is extremely time consuming, in particular if one tries to calculate the spectral absorption coefficient by summing directly the contribution of each line to absorption. In this case the task soon becomes prohibitive for three reasons: (a) hundreds of thousands of vibrational and rotational transitions contribute to absorption; (b) each transition defines a spectral line that can contribute to absorption over a wide range of frequency; and (c) absorption coefficients must be sampled at a frequency interval that is sufficiently small to resolve the thinnest lines of interest.
Rather than summing directly the contributions to absorption, the method used in KOPRA relies on establishing for each line a set of grid points at which the difference between the analytic value of such a contribution and its interpolated approximation exceeds a limiting small value to be specified by the user; line profiles are evaluated only at these frequencies. The interpolation routine itself is based on three-point Lagrangian interpolation. The line profiles are then added to summations of line profiles calculated with the same spectral resolution. Local absorption coefficients are constructed when the summations are complete.
An important feature of the method is that the frequencies at which each line profile is evaluated are determined dynamically. The spacing between such points increases continuously with distance from line center. Narrow lines are evaluated on a finer scale than broad lines of the same intensity. Lines of stronger intensity are generally evaluated on more points than those of weaker intensity.
The method of approximation is subject to the following three sections which are part of a paper published in JQSRT in 1999 [11]. Following these sections the implementation of the algorithm in Fortran90 will be described together with a depiction of the user interface.

2 Calculation of the line intensity

The line intensity S_l of line l is determined as

$$S_l(T_{kin}, \nu) = S_l(T_0, \nu) \cdot \frac{Q(T_0)}{Q(T_{kin})} \, \frac{e^{-\frac{E_l''}{k_B T_{kin}}}}{e^{-\frac{E_l''}{k_B T_0}}} \, \frac{1 - e^{-\frac{h c \nu}{k_B T_{kin}}}}{1 - e^{-\frac{h c \nu}{k_B T_0}}} \tag{2}$$

where

k_B	Boltzmann constant
E_l''	lower state energy of transition l
$Q(T)$	LTE total internal partition function evaluated at T
$Q(T_0)$	LTE total internal partition function at T_0
$A_l(T_0)$	line intensity at reference temperature T_0
T_{kin}	kinetic temperature
T_0	reference temperature (296 K)
ν	wavenumber, usually approximated as $\nu_{0,l}$
$\nu_{0,l}$	central wavenumber of transition l

The lower state energy E'' is calculated from the lower state energy e'' in units of wavenumbers as given in spectroscopic databases by:

$$E_l'' = c\, e_l''\, h \tag{3}$$

The total internal partition function is determined by use of Gamache's parameterization [7]:

$$Q(T) = a_0 + a_1 T + a_2 T^2 + a_3 T^3 \tag{4}$$

where a_0, a_1, a_2, and a_3 are pretabulated coefficients stored in the file `hitmol.dat` (`$1.3` in the main KOPRA input file).

3 Efficient calculation of the line shape factor

Different line shape factors may be required for different atmospheric conditions. In the lower atmosphere, the shape of spectral lines is dominated by pressure broadening and can be represented most simply by the Lorentz line shape factor:

$$f_{Ll}(\nu, \nu_l) = \frac{1}{\pi} \frac{\alpha_{L,l}}{(\nu - \nu_l)^2 + \alpha_{L,l}^2}, \tag{5}$$

where $\alpha_{L,l}$ is the Lorentz half-width at half maximum of the line. At high altitudes, the shape of spectral lines is governed by Doppler broadening. The corresponding line shape factor is

$$f_{D,s,l}(\nu, \nu_l) = \frac{1}{\alpha_{D,s,l}\sqrt{\pi}} \exp\left[-\frac{(\nu - \nu_l)^2}{\alpha_{D,s,l}^2}\right], \tag{6}$$

where $\alpha_{D,s,l}$ is the Doppler half-width at $1/e$ of the maximum for species s. At intermediate altitudes, both pressure and Doppler broadening are important. They can be modeled using the Voigt line shape factor, a convolution of the Lorentz and Doppler line shape factors:

$$f_{V,s,l}(\nu, \nu_l) = \frac{1}{\alpha_{D,s,l}\sqrt{\pi}} \underbrace{\frac{y}{\pi} \int_{-\infty}^{\infty} \frac{\exp(-t^2)}{y^2 + (x - t)^2} dt}_{K_{V,s,l}(x,y)}, \tag{7}$$

where $x = (\nu - \nu_l)/\alpha_{D,s,l}$ is the distance from line center in units of the Doppler half-widths, $y = \alpha_{L,l}/\alpha_{D,s,l}$ is the ratio of the Lorentzian half-width to the Doppler half-width, and $K_{V,s,l}(x, y)$ is the Voigt profile function. The Voigt line shape factor reduces to the Lorentz and Doppler line shape factors, respectively, in the appropriate limits.

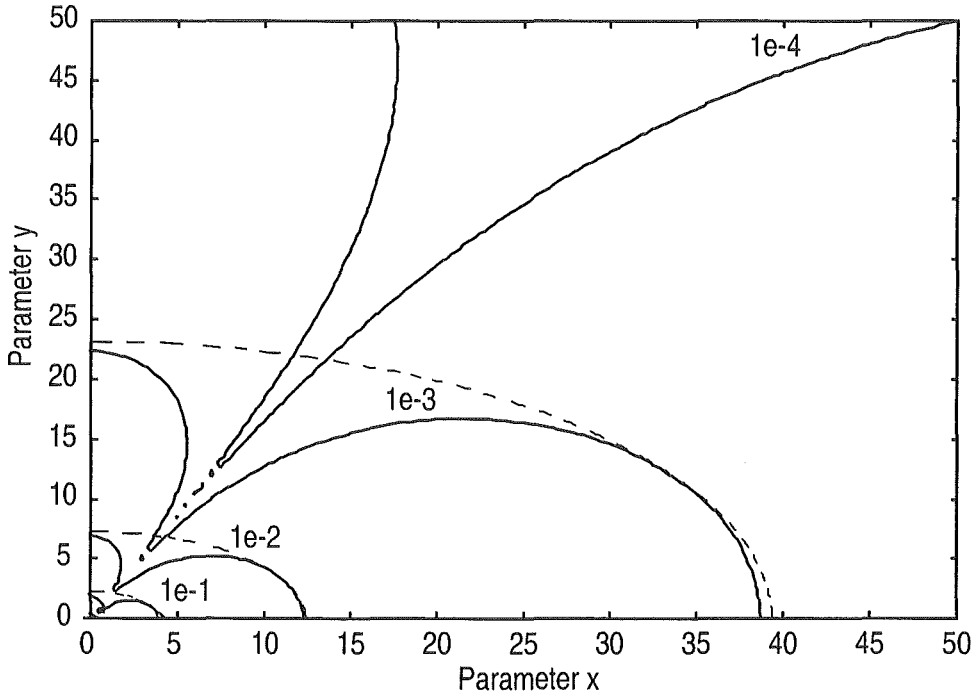


Figure 1: Contours of $|K_{L,s,l}(x,y) - K_{V,s,l}(x,y)|/K_V(x,y)$, i. e. the relative error obtained by approximating the Voigt profile function by the Lorentz profile function. Dashed lines mark the regions outside which the Voigt profile function should be approximated by the Lorentz profile function. Numbers indicate the desired relative accuracy $\mathcal{E}$.

Since the integral in (7) cannot be evaluated analytically, the Voigt profile function $K_{V,s,l}(x,y)$ must be calculated numerically. It is thus desirable to substitute the Voigt line shape factor by the numerically less expensive Lorentz and Doppler line shape factors wherever possible. For deriving an appropriate criterion it seems reasonable to rewrite the Lorentz and Doppler line shape factors (2) and (6) as

$$f_{L,l}(\nu,\nu_l) = \frac{1}{\alpha_{D,s,l}\sqrt{\pi}} \underbrace{\frac{1}{\sqrt{\pi}} \frac{y}{x^2+y^2}}_{K_{L,s,l}(x,y)} \tag{8}$$

and

$$f_{D,s,l}(\nu,\nu_l) = \frac{1}{\alpha_{D,s,l}\sqrt{\pi}} \underbrace{\exp[-x^2]}_{K_{D,s,l}(x,y)}, \tag{9}$$

respectively, where $K_{L,s,l}(x,y)$ and $K_{D,s,l}(x,y)$ are the analogues to the Voigt profile function $K_{V,s,l}(x,y)$ in (7). Since the profile functions $K_{L,s,l}(x,y)$, $K_{D,s,l}(x,y)$, and $K_{V,s,l}(x,y)$ depend only on x and y it is much easier to compare the line shape factors in this form. Figure 1 displays contours of the relative error $|K_{L,s,l} - K_{V,s,l}|/K_{V,s,l}$ on a plane with coordinates x and y. Obviously, the larger the distance from the line center frequency x and the larger the ratio of the Lorentz to the Doppler half-width y is, the smaller the corresponding approximation error becomes. We therefore approximate the Voigt profile function by the Lorentz profile function outside the elliptical regions bounded by the dashed lines in Figure 1. The appropriate criterion may be written as

$$x^2 > 1.52/\mathcal{E} - 2.84y^2,$$

Table 1: Number of floating point operations (Flops) within each region. Exponentiation is considered as 10 floating point operation equivalents.

	Lorentz	Voigt				Doppler
Flops	Region 0	Region 1	Region 2	Region 3	Region 4	Region 5
New	3	7	15	19	79	11
Ref. [10]	7^1	7	15	19	79	19^3 or 79^4
Ref. [9]	23^1	23	41	75	148	75^3 or 148^4

^{1,3,4} Numbers indicated correspond to the former regions 1, 3, and 4, respectively.

where $\mathcal{E}$ represents the desired relative accuracy. Analogous considerations apply for the derivation of an appropriate criterion for the Voigt and the Doppler line shape factor. Figure 2 shows contours of the relative error $|K_{D,s,l} - K_{V,s,l}|/K_{V,s,l}$ in the x-y plane. Note that the relative error is only small within a small region around $y = 0$ that is relatively complicated in shape. However, since the evaluation of the Voigt profile function for small values y is computationally expensive [9][?] (see also region 5 in Table 1), we use the Doppler line shape factor if

$$|x| < 2.15 - 2.53y/\mathcal{E},$$

as is indicated by the dashed lines in Figure 2. Using a more precise parameterization of the contours would be computationally too expensive.
In regions, where neither the Lorentz nor the Doppler line shape factors are appropriate, we use an accelerated implementation of the so-called Humlicek algorithm for the approximation of the Voigt line shape factor [10]. This algorithm involves dividing the x-y plane into four regions, where ratios of rational polynomials are used to achieve relative accuracy better than 1 part in 10^4. To perform the transition from the Voigt line shape factor to the Lorentz and Doppler line shape factors automatically, we add two further regions, region 0 and 5, to the original set of four regions. The borders of these regions are chosen according to the above criteria, and the profile function within these regions is evaluated using (8) and (9), respectively. The extent of saving of floating point operations within these two regions is indicated in Table 1.

4 The optimum set of sampling points

4.1 The method of approximation

For optimum speed, the rapid evaluation of the Voigt profile function must be augmented with a scheme that minimizes its use by reducing the resolution and therefore the number of function evaluations. Numerous strategies have been proposed to achieve this [2][4][5][13][14] . A common approach is to model the absorption coefficient spectra by using multiple frequency grids: fine grids are used to model the line shape factor near the line center frequency, while coarse grids are used for the line wings. However, reducing the resolution requires interpolation to higher resolution grids and introduces numerical errors, see Figure 3. The finer the frequency spacing Δx and the larger the distance from the line center frequency x is, the smaller the interpolation error $\delta K_{s,l}$ becomes.
The principle first proposed by Sparks [13] in order to keep these interpolation errors under control is to introduce an user-specified absolute accuracy coefficient $\delta\sigma$, to which the contribution of any line to the absorption coefficient (1) needs to

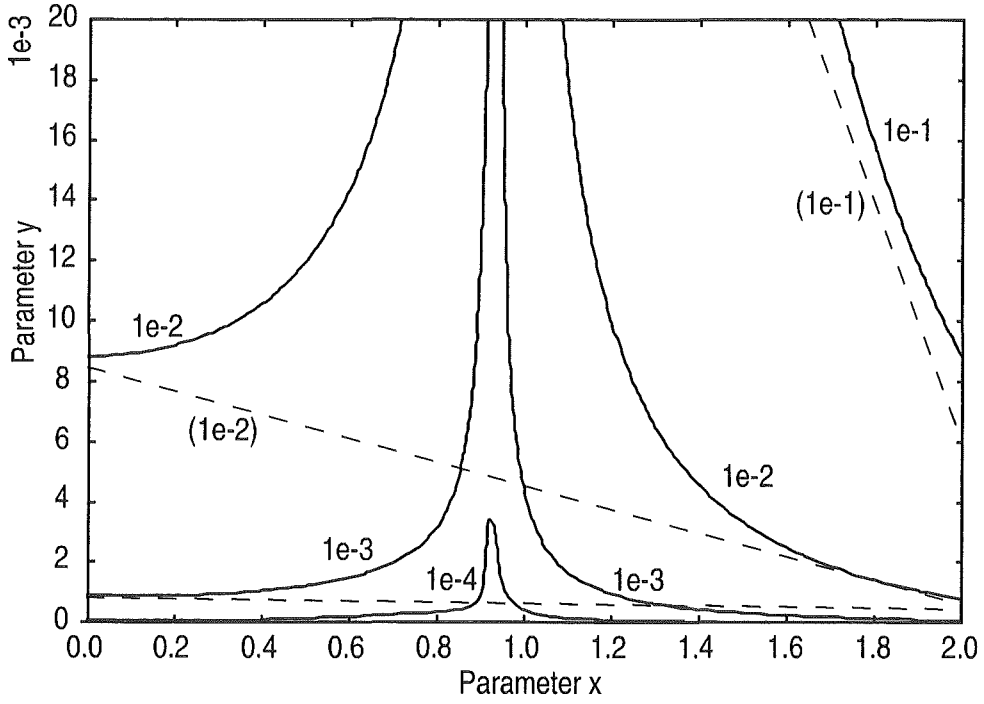


Figure 2: Same as Figure 1 except that the Lorentz profile function has been replaced by the Doppler profile function. Dashed lines mark the regions inside which the Voigt profile function should be approximated by the Doppler profile function.

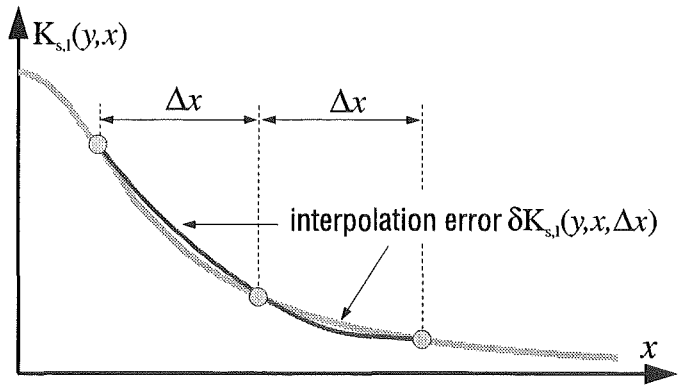


Figure 3: Approximating the profile function by a three-point Lagrangian interpolation.

be calculated to. Inserting Equations (1) and (7)-(9) this condition may be written as

$$\delta\sigma \geq \frac{N_s S_l}{\alpha_{D,s,l}\sqrt{\pi}}\delta K_{s,l},$$

or equivalently

$$\delta K_{s,l} \leq \frac{\alpha_{D,s,l}\sqrt{\pi}}{N_s S_l}\delta\sigma, \tag{10}$$

where $\delta K_{s,l}$ is the interpolation error that can be accepted for a given line. Note that this interpolation error is inversely proportional to the number density N_s of the molecular species s and the line intensity S_l. The stronger a given line is, the smaller the error bounds become within which the interpolation has to be performed to.

To determine the interpolation error associated with a given line and frequency spacing Sparks [14] suggests to approximate the profile function $K_{s,l}(x,y)$ by a simple three-point Lagrangian interpolation [1] :

$$\begin{aligned} K_{s,l}(x_0 + p\Delta x, y) \approx \\ &\frac{1}{2}p(p-1)K_{s,l}(x_0 - \Delta x, y) + \\ &(1-p^2)K_{s,l}(x_0, y) + \\ &\frac{1}{2}p(p+1)K_{s,l}(x_0 + \Delta x, y), \end{aligned}$$

or, setting $x = x_0 + p\Delta x$,

$$\begin{aligned} K_{s,l}(x, \Delta x, y) \approx \\ &\frac{1}{2}p(p-1)K_{s,l}(x - \Delta x - p\Delta x, y) + \\ &(1-p^2)K_{s,l}(x - p\Delta x, y) + \\ &\frac{1}{2}p(p+1)K_{s,l}(x + \Delta x - p\Delta x, y) \\ &\equiv \overline{K_{s,l}}(x, \Delta x, y). \end{aligned}$$

The interpolation error is then given by

$$\delta K_{s,l}(x, \Delta x, y) \equiv K_{s,l}(x, \Delta x, y) - \overline{K_{s,l}}(x, \Delta x, y).$$

The problem is to identify the frequencies and frequency spacings at which condition (10) is violated for a given line. Note that the maximum of the Voigt profile function $K_{s,l}(x = 0, y)$ and hence $\delta K_{s,l}(x = 0, \Delta x, y)$ is proportional to $1/y$ for $y \gg 1$ (Lorentz regime), while $\delta K_{s,l}(x = 0, \Delta x, y)$ remains almost constant for $y \ll 1$ (Doppler regime). We therefore multiply (10) on both sides with $y + 1 = (\alpha_{D,s,l} + \alpha_{L,l})/\alpha_{D,s,l}$ (this essentially corresponds to the Voigt half-width in units of the Doppler half-width) and consider the criterion

$$(y+1)\delta K_{s,l} \leq \frac{(\alpha_{D,s,l} + \alpha_{L,l})\sqrt{\pi}}{N_s S_l}\delta\sigma, \tag{11}$$

rather than (10). Condition (11) reduces to expressions (12) and (16) in the work of Sparks [14] in the appropriate limits. Figures 4-6 display contours of $\underset{-1\leq p\leq 1}{\text{Max}}\,(y+1)|\delta K_{s,l}(x, \Delta x, y)|$ on planes with coordinates $\log_2(\Delta x/(y+1))$ and $\log_2|x/\Delta x|$ for different ratios of the Lorentz to the Doppler half-width, y. Note

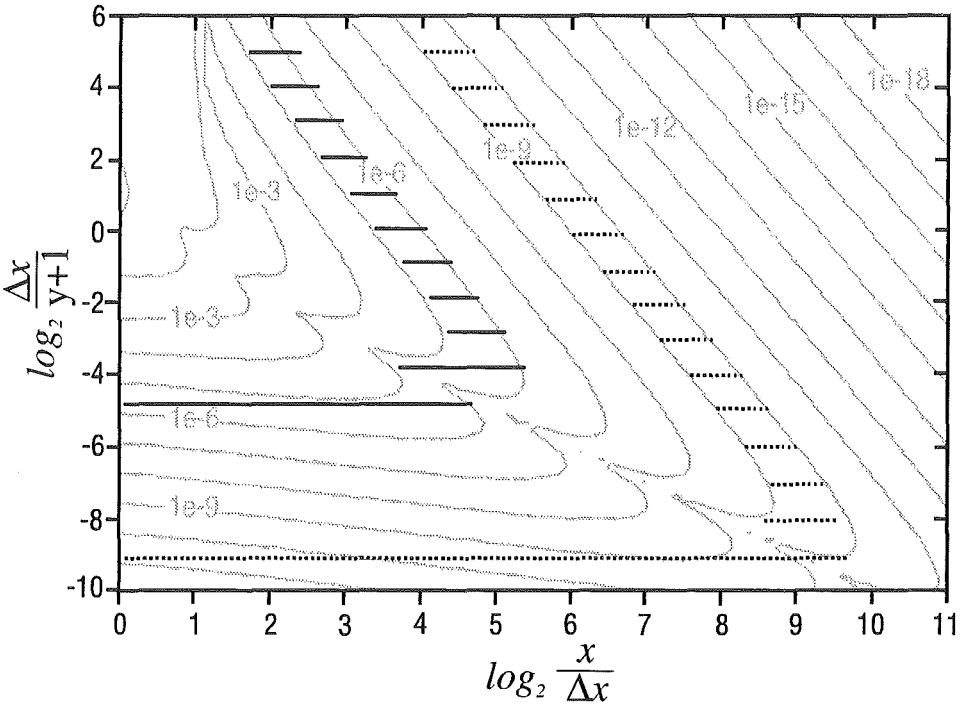


Figure 4: Contours of $\underset{-1\leq p\leq 1}{\text{Max}}\,(y+1)\delta K_{s,l}(x,\Delta x,y)$ for $y \gg 1$ (Lorentz regime). Also indicated is the optimum set of sampling points for a maximum acceptable interpolation error of $(y+1)\delta K_{s,l} \leq 10^{-5}$ and $\leq 10^{-9}$, respectively.

that $\Delta x/(y+1) = \Delta\nu/(\alpha_{D,s,l}+\alpha_{L,l})$ is the frequency spacing in units of the Voigt half-width and that $|x/\Delta x| = |\nu-\nu_l|/\Delta\nu$ is the distance from the line center frequency in units of frequency spacing; hence the two axes are not independent of each other. The logarithmic scale on the base two is chosen to account for the second order polynomial used for interpolation. Although this choice of axes seems rather complicated, it is in principle sufficient to note that the vertical axis essentially denotes frequency spacing while the horizontal axis indicates the number of sampling points that should be considered using a given frequency spacing. How this information can be used to determine the optimum set of sampling points for a given line is best illustrated considering examples.

4.2 Cutoff tables

Let us first examine the Lorentz regime, i.e. $y \gg 1$; the corresponding contours of $\underset{-1\leq p\leq 1}{\text{Max}}\,(y+1)\delta K_{s,l}(x,\Delta x,y)$ are shown in Figure (4). Let us further assume that a given line, indicated by the contour $\delta\sigma\sqrt{\pi}(\alpha_{D,s,l}+\alpha_{L,l})/N_sS_l$, needs to be interpolated with an interpolation error $\delta K_{s,l}$ so that $(y+1)\delta K_{s,l} \leq 10^{-5}$. This implies a frequency spacing of $\log_2(\Delta x/(y+1)) = \log_2(\Delta\nu/(\alpha_{D,s,l}+\alpha_{L,l})) \sim -5$ near the line center frequency and the transition to a coarser frequency grid in a distance of $\log_2|x/\Delta x| = \log_2(|\nu-\nu_l|/\Delta\nu) \sim 4.7$, i.e. ~ 26, sampling points (indicated by the solid lines in Figure 4). It would be unwise to start with a finer frequency grid from the beginning since the above value bounds the contour from below and any frequency spacing finer than $\Delta x = (y+1)2^{-5}$ or $\Delta\nu = (\alpha_{D,s,l}+\alpha_{L,l})2^{-5}$ would result in an interpolation error smaller than the one that can be accepted. On the coarser frequency grid one should evaluate $2^{5.5}-2^{3.8}$, i.e. ~ 32,

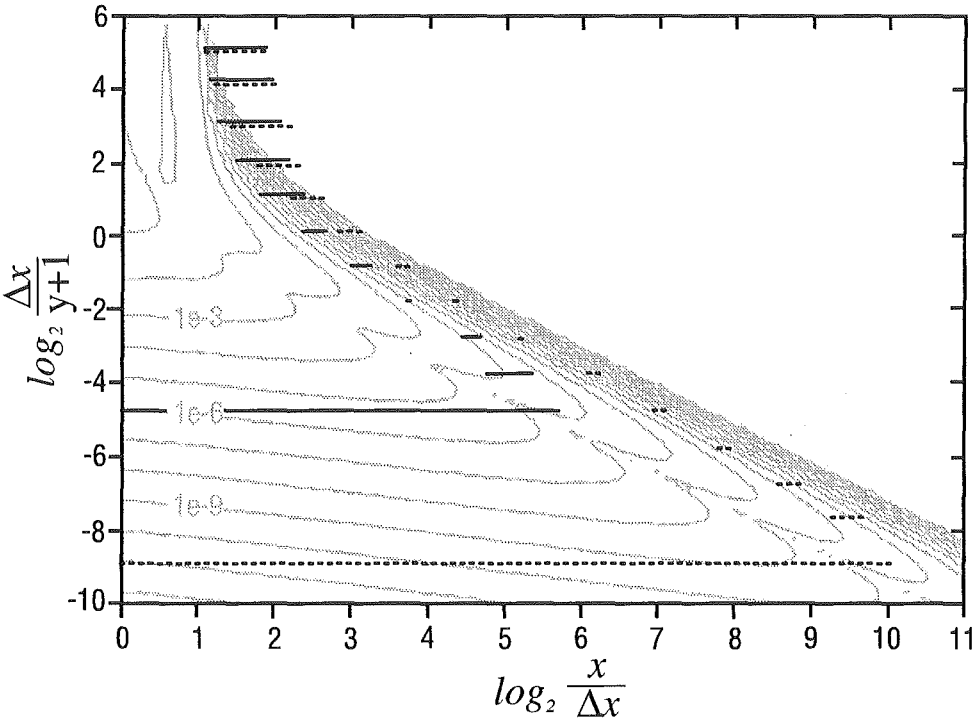


Figure 5: Same as Figure 4 except that the Lorentz regime has been replaced by the Doppler regime, i. e. $y \ll 1$.

sampling points before again proceeding to a coarser frequency grid and so on. The point is to proceed to a coarser frequency grid only if the interpolation error on the coarser grid falls below the maximum interpolation error that can be accepted for a given line. Note that due to the second order polynomial used for interpolation, the frequency spacing always increases by a factor of two.

Let us consider a stronger line that requires an interpolation error $\leq 10^{-9}$. In this case one would start with a much finer frequency grid from the beginning (corresponding to a frequency spacing of $\log_2(\Delta x/(y+1)) \sim -9$ near the line center frequency) and evaluate $\log_2 |x/\Delta x| \sim 10$, i.e. ~ 1000, sampling points before proceeding to a coarser frequency grid (indicated by the dotted lines in Figure 4).

The same procedure can in principle be applied to the Doppler regime, i.e. $y \ll 1$ (see Figure 5), so that tables of cutoffs may be constructed for both limits. These tables indicate over a wide range of possible interpolation errors and frequency spacings in which distance from the line center frequency (or equivalently after the consideration of how many sampling points using a given frequency spacing) one should proceed to a coarser frequency grid. For a given spectral line one can then evaluate the maximum interpolation error $\delta\sigma\sqrt{\pi}(\alpha_{D,s,l} + \alpha_{L,l})/N_s S_l$ and use these tables to define the appropriate Lorentz and Doppler cutoffs for any frequency spacing $\Delta x/(y+1) = \Delta\nu/(\alpha_{D,s,l} + \alpha_{L,l})$ desired. Note that either $\alpha_{D,s,l}$ or $\alpha_{L,l}$ are negligible in the appropriate limits. We will henceforth refer to these tables as the Lorentz and the Doppler cutoffs.

As we mentioned previously, in the real atmosphere both pressure broadening and Doppler broadening are important and can be modeled using the Voigt profile function. Since the latter, however, depends on both x and y, it is in principle necessary to establish a complete set of cutoff tables: one for each ratio of the Lorentz to the Doppler halfwidth y—an enormous waste of computer memory. Figure 6 shows a

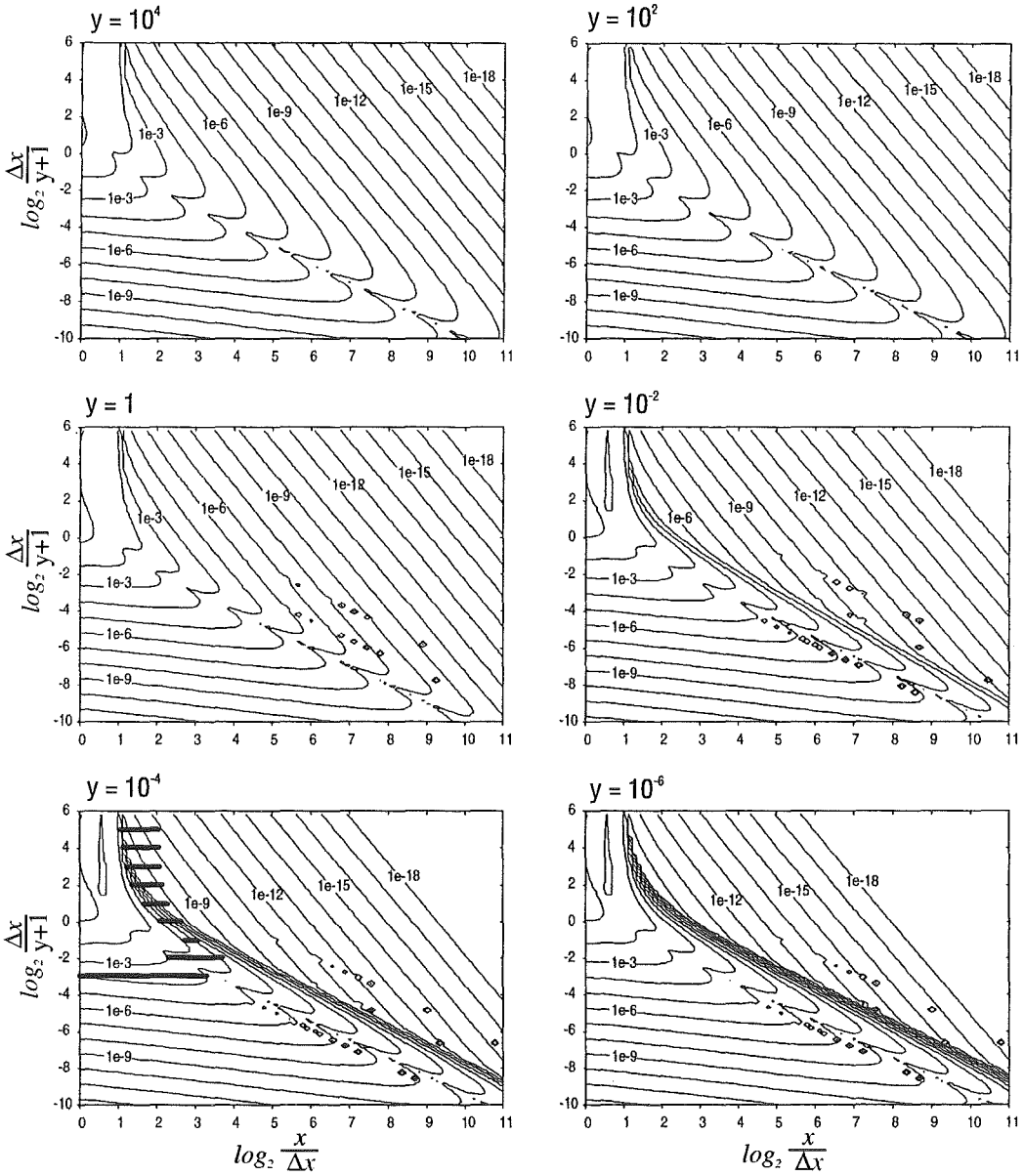


Figure 6: Contours of $\underset{-1\leq p\leq 1}{\text{Max}}\,(y+1)\delta K_{s,l}(x,\Delta x,y)$ for different ratios of the Lorentz to the Doppler half-width, y. Also shown is the optimum set of sampling points for a maximum permissible interpolation error of $(y+1)\delta K_{s,l} \leq 10^{-3}$ for $y = 10^{-4}$.

subset of six contourplots (cutoff tables) for a selected sample of ratios y. Note that the contourlines change from a more Lorentz-like course for large values of y to a more Doppler-like course as y successively decreases (compare Figures 4 and 5).
To overcome this waste of computer memory Sparks [14] suggests to neglect the contribution of a spectral line to absorption on a given frequency grid if the Lorentz cutoff is equal to zero; otherwise, he takes advantage of the fact that the Voigt profile function reduces to the Lorentz and the Doppler profile functions in the appropriate limits and uses the greater of the Lorentz and the Doppler cutoffs for the approximation of the Voigt cutoff. The corresponding contourlines are shown in Figure 7. This approximation works quite well in the Lorentz regime, i.e. for $y > 1$, but fails for $y < 1$ (compare the contours shown in Figures 6 and 7).
Consider for example an maximum interpolation error of 10^{-3} and a ratio of the Lorentz to the Doppler half-width of 10^{-4} (Doppler regime). According to Figure 6 this implies a frequency spacing of $\log_2(\Delta x/(y+1)) \sim -3$ near the line center frequency and the calculation of $\log_2 |x/\Delta x| \sim 2^{3.3}$, i.e. ~ 10, sampling points before proceeding to a coarser frequency grid. The approximation of Sparks [14], however, results in a much finer frequency grid from the beginning (corresponding to a frequency spacing of $\log_2(\Delta x/(y+1)) \sim -7$, i.e. by a factor 16 finer) and requires the evaluation of $\log_2 |x/\Delta x| \sim 2^{6.7}$, i.e. ~ 104, sampling points before proceeding to a coarser frequency grid. Altogether this approximation results in the evaluation of ~ 390 sampling points compared to ~ 30 really necessary according to Figure 6. We mentioned already that the evaluation of the Voigt profile function for small values y is computationally extremely expensive so that any reduction of the number function evaluations in the Doppler regime results in a noticeable gain of computational speed). This incongruity becomes even worse if one prefers to use the simpler but less restrictive criterion (20) in the work of Sparks [14] instead of refering to the appropriate cutoff table. It should be mentioned that this approximation is not a source of computational error—quite the reverse is true. Absorption coefficients in the Doppler regime are calculated much more accurately than necessary. The aim, however, is to consider each spectral line with the same absolute accuracy $\delta\sigma$.
Instead we take advantage of the fact that the interpolation error of the Voigt profile function $\delta K_{V,s,l}(x, y, \Delta x)$ can be well approximated by the greater of the interpolation errors of the Lorentz and the Doppler profile function weighted by the squares of the ratio of the Lorentz to the Voigt half-width $y/(y+1) = \alpha_L/(\alpha_L+\alpha_D)$ and the ratio of the Doppler to the Voigt half-width $1/(y+1) = \alpha_D/(\alpha_L+\alpha_D)$, i.e.

$$\begin{aligned}\delta K_{V,s,l}(x, y, \Delta x) \approx \\ \max\big(&y^2/(y+1)^2 \delta K_{L,s,l}(x_L, y, \Delta x_L), \\ &1/(y+1)^2 \delta K_{D,s,l}(x_D, y, \Delta x_D)\big).\end{aligned} \tag{12}$$

Note that the auxiliary variables Δx_L and Δx_D for a given frequency spacing can be calculated from $\Delta x_L = \Delta x\, y/(y+1) = \Delta\nu\, \alpha_L/\alpha_D(\alpha_L + \alpha_D)$ and $\Delta x_D = \Delta x\, y/(y+1) = \Delta\nu/(\alpha_L + \alpha D)$, respectively. Putting (12) into (11) results in the criteria

$$(y+1)\delta K_{L,s,l}(x_L, y, \Delta x_L) \le \frac{(\alpha_{D,s,l} + \alpha_{L,l})^2\sqrt{\pi}}{\alpha_L N_s S_l}\delta\sigma \tag{13}$$

and

$$(y+1)\delta K_{D,s,l}(x_D, y, \Delta x_D) \le \frac{(\alpha_{D,s,l} + \alpha_{L,l})^2\sqrt{\pi}}{\alpha_D N_s S_l}\delta\sigma \tag{14}$$

for the determination of the auxiliary cutoffs x_L and x_D from the precalculated Lorentz and Doppler cutoff-tables, respectively. The greater of x_L and x_D is finally used for the approximation of the Voigt cutoff.

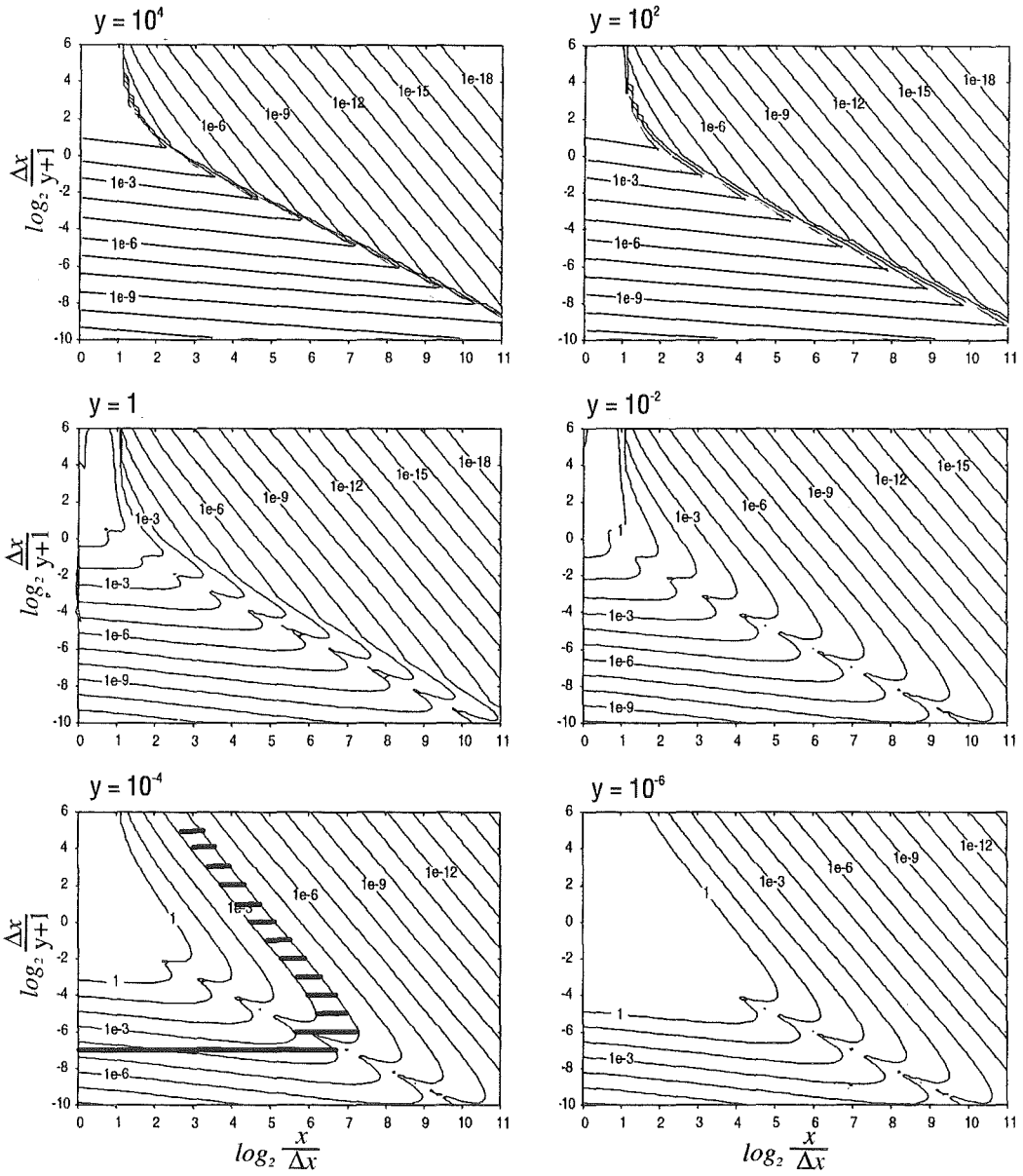


Figure 7: Sparks' [14] approximation for the determination of the Voigt cutoffs. Also shown is the resulting set of sampling points for $(y+1)\delta K_{s,l} \leq 10^{-3}$ and $y = 10^{-4}$.

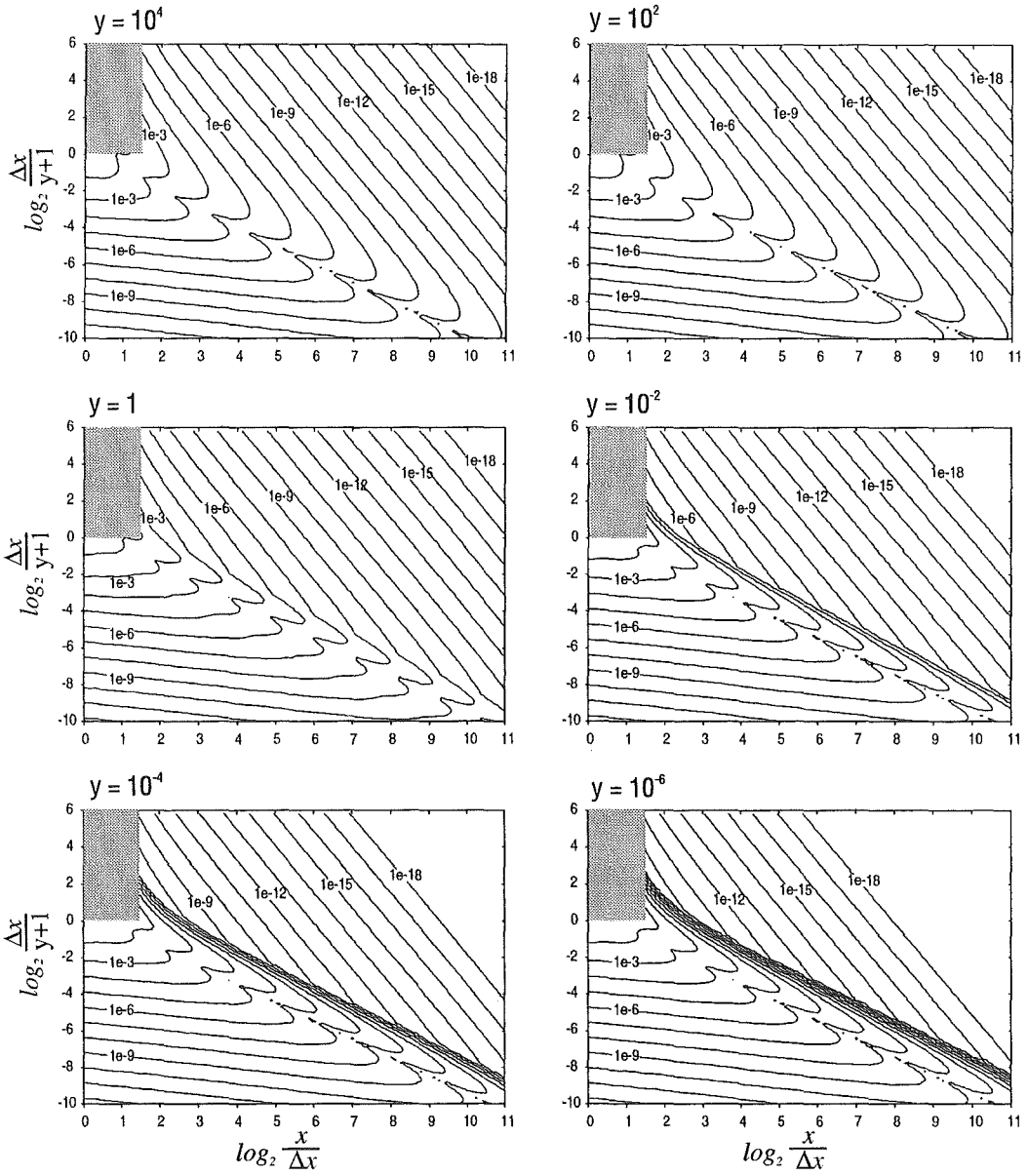


Figure 8: Proposed approximation for the determination of the Voigt-cutoffs. The light shaded regions indicate the domain of frequencies where lines are truncated.

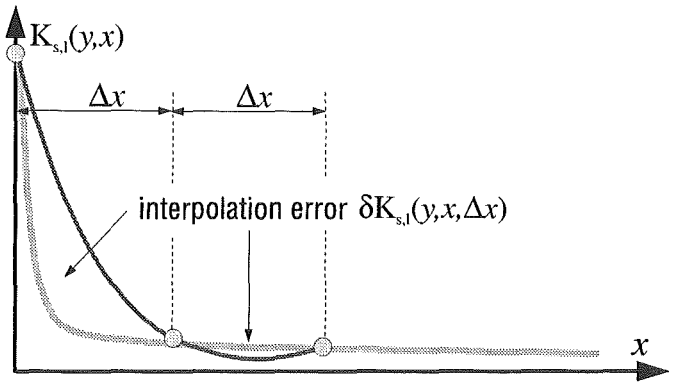


Figure 9: Approximating the profile function by a three-point Lagrangian interpolation for large frequency spacings Δx.

The quality of this new approximation is striking (compare the contours in Figure 8 with the true contours in Figure 6). Small deviations occurr only for $y \sim 1$. In addition, the evaluation of the new criterion requires only four extra floating point operations compared to the criterion proposed by Sparks [14]. This is certainly negligible compared to the achieved reduction of function evaluations. Furthermore, it should be mentioned that in the proposed algorithm the use of cutoff-tables is minimized so that all sampling points that have to be considered on a given frequency grid are determined with a single reference to the corresponding cutoff-tables, whereas Sparks [14] examines each sampling point individually.

4.3 Line rejection and truncation

Another question that has not yet been mentioned is which lines should be retained in the model and which lines may be rejected. Furthermore, we need a criterion to decide in which distance from the line center frequency spectral lines may be truncated without substantial loss of accuracy. This is especially true, if the far wings of lines are incorporated into models of the continuum and need not be taken into account explicitly [3]. Sparks [14] provides only two alternatives: to reject a spectral line completely, or to consider it over the entire frequency range of interest. In the proposed algorithm lines are only rejected, if $K_{s,l}(x = 0, y)$ is smaller than the maximum interpolation error that can be accepted for a given line or if the transition frequency of the line lies beyond a user-specified distance from the endpoints of the frequency range of interest. This distance is set equal to 25 cm^{-1}. Other lines may only be truncated.

A closer inspection of Figures 4-6 reveals that all contours converge to $\log_2(x/\Delta x) = 1$ for increasing values $\log_2(\Delta x/(y+1))$. This implies that the interpolation error becomes negligible only in a distance $x > 2\Delta x$, regardless how large the frequency spacing is. This can be understood if one recalls that each contour represents the maximum interpolation error within a region $\pm\Delta x$ around a given distance x from the line center frequency. The maximum interpolation error thus approaches $K_{s,l}(x = 0, y)$ for increasing frequency spacings Δx, as is shown in Figure 9. This also explains why the algorithm of Sparks [14] evaluates a contribution to absorption for each spectral line retained in the model over the entire frequency range of interest, rather than truncating it beyond a certain distance. In the proposed algorithm, we assume that the central part of each spectral line is considered on grids of finer frequency spacing so that spectral lines are truncated if $\log_2(x/\Delta x) \leq 1.5$

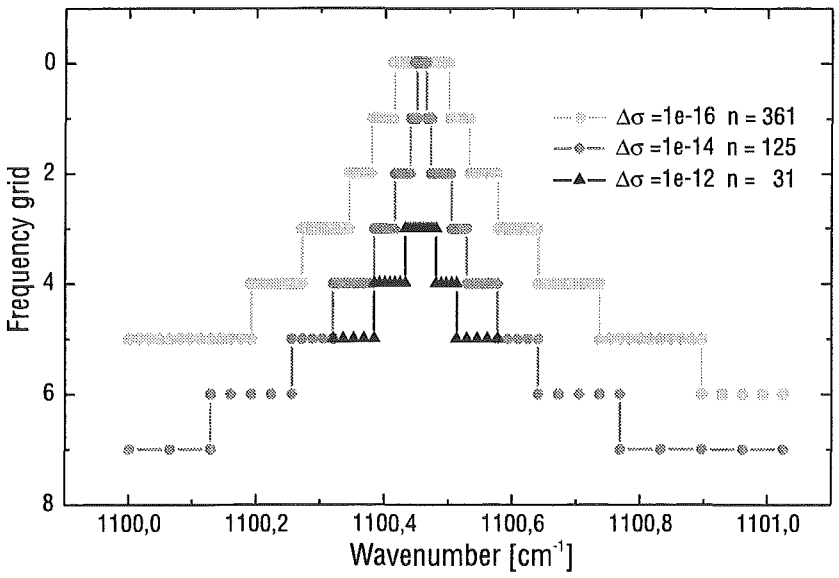


Figure 10: Representations of a given spectral line of maximum absorption $\sigma = 10^{-10}$ cm^{-1} for different absolute accuracies $\Delta\sigma$ on a set of successively finer frequency grids. The absolute accuracies $\Delta\sigma$ and the number of function evaluations are indicated.

and $\log_2(\Delta x/(y+1) \geq 0)$, as is indicated by the light-shaded regions in Figure (8). This assumption is strengthened by the fact that the frequency spacing of the finest frequency grid is usually chosen according to the Doppler half-width at high altitudes. Nevertheless, the truncation of lines may also be suppressed as an option so that each spectral line retained in the model is considered over the entire frequency range of interest on at least three sampling points. Test calculations revealed that the above criterion meets the requirement of negligible truncation errors quite well, especially if the frequency range of interest is small (microwindow approach) and the far wings are incorporated into models of the continuum.

Figure 10 displays different representations of a given spectral line with a maximum absorption coefficient of $\sigma = 10^{-10}$ cm^{-1} for different absolute accuracies $\Delta\sigma$. Note that the frequency spacing is finest near the line center frequency and successively increases with increasing distance from it. For the lowest accuracy $\delta\sigma = 10^{-12}$ cm^{-1}, 31 sampling points are adequate, while 361 sampling points are necessary to achieve an absolute accuracy $|\delta\sigma| \leq 10^{-16}$ cm^{-1}. It should be mentioned that for the lowest accuracy the spectral line has been truncated.

An example of the contribution of a single spectral line to absorption is shown in Figure 11 together with a reference spectrum calculated on an equidistant frequency grid of uniform intervals. The resulting interpolation error is shown below. The aim was an absolute accuracy $|\delta\sigma| \leq 10^{-16}$ cm^{-1} which could obviously be achieved over the entire frequency range of interest. Furthermore, it is interesting to note how the interpolation error always increases when the algorithm proceeds to a coarser frequency grid.

5 Efficient summation of spectral lines

The summation of spectral lines follows much of the methodology suggested by Fomin [5] and is performed on a set of successively finer frequency grids. Each of these grids is made up of intervals that consist of three sampling points due to the second order polynomial used for interpolation, see Figure 12. It should be

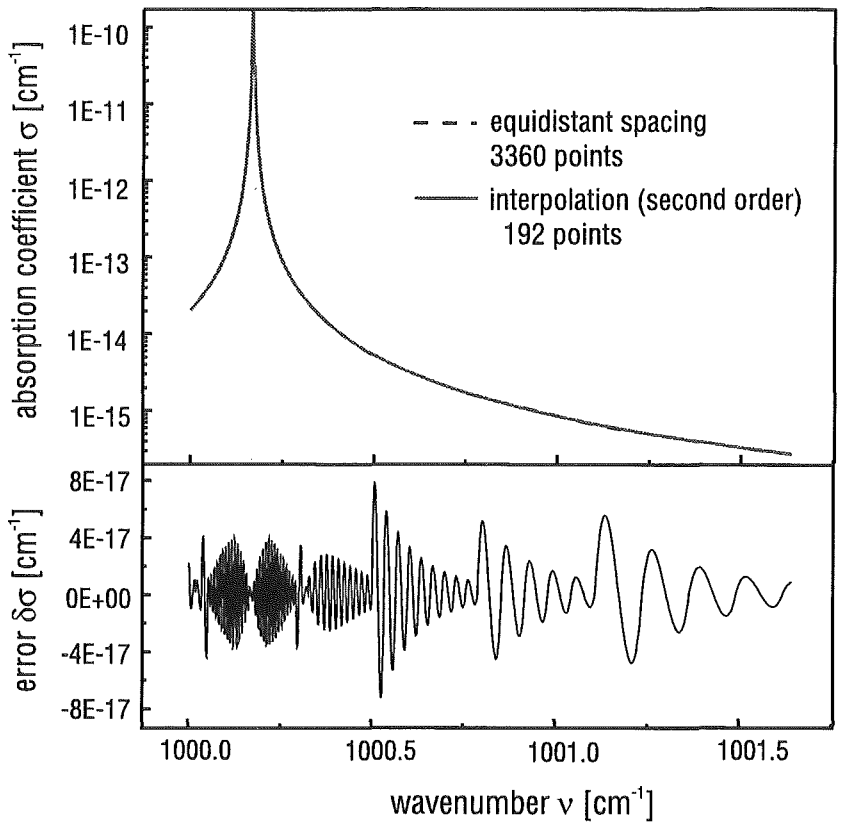


Figure 11: (top) Contribution of a single spectral line to absorption applying the new method of approximation together with the reference spectrum evaluated on an equidistant frequency grid of uniform intervals. (bottom) The resulting interpolation error.

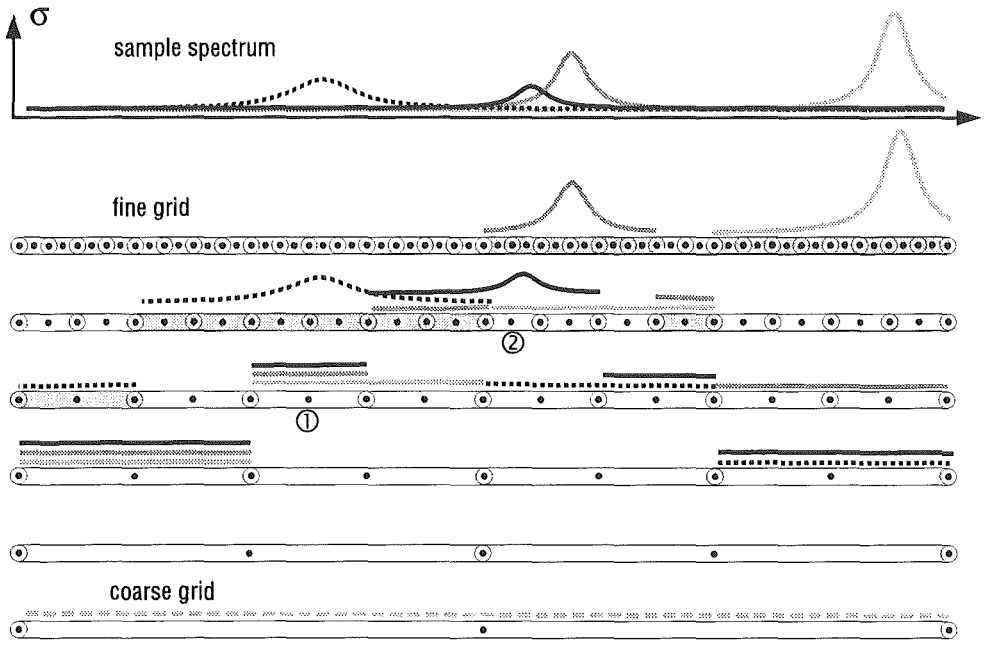


Figure 12: Illustration of the proposed interpolation technique on a set of successively finer frequency grids. The resulting quasi-uniform frequency grid is indicated by the light shaded intervals.

mentioned that the two endpoints must be considered individually for each interval (i.e. it takes two storage locations). This makes possible to consider each spectral line independently from the others so that each spectral line may be evaluated on its own optimum set of sampling points. The procedure is similar to the technique that has been applied in the widely used algorithm of Clough and Kneizys [2] or Gordley et al. [8] and in particular in the work of Fomin [5], except that the number of frequency grids for each line is determined dynamically.

The summation of spectral lines is performed in such a way that for each line the contribution to absorption is added up in the appropriate intervals. For example consider the interval marked by ① in Figure 12, to which three spectral lines of different strengths and half-widths contribute. Note that only those spectral lines, whose transition frequencies lie far outside the frequency range of interest, contribute to the interval residing on the coarsest frequency grid. Because each succeeding frequency grid is by a factor of two finer than the preceeding one, it is possible to move back to a coarser frequency grid only from every second interval. This may sometimes result in the calculation of an extra interval on a given frequency grid as for example for the light shaded spectral line in the intervals marked by ① and ②.

Having completed the summation of spectral lines, absorption coefficients are calculated in a straightforward manner starting on the coarsest frequency grid and moving progressively to grids of finer frequency spacing while interpolating and adding the coarser intervals to the appropriate intervals on the succeeding finer frequency grid. Note that by using an interpolation scheme with constant coefficients (e.g. Lagrangian interpolation) the contribution of each interval to the succeeding intervals can be calculated as a linear combination of the contributions to absorption evaluated on the current frequency grid. The interpolation step thus is computationally efficient, especially since it has to be performed only once after the summation over relevant lines has been completed.

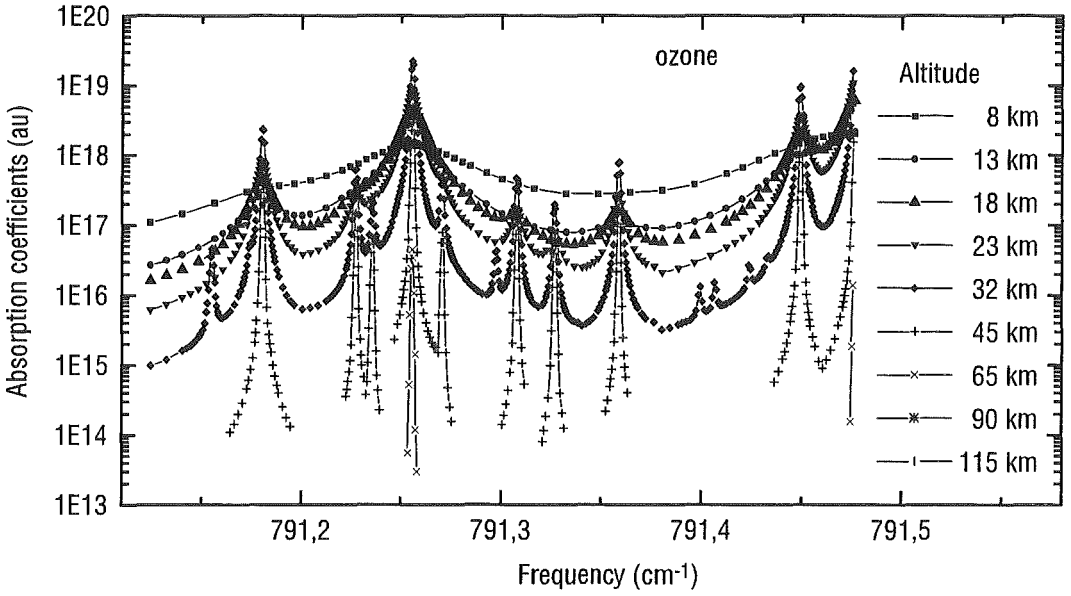


Figure 13: Absorption coefficients of ozone at different altitudes calculated for a limb sounding geometry and a tangent height of 8 km.

Rather than terminating the interpolation only if the finest frequency grid is reached, the interpolation may readily be terminated at the finest intervals for which a contribution to absorption has been evaluated. In this case one ends up with a nonuniform or nonequidistant frequency grid which is, however, based on an equidistant fine grid of uniform intervals, indicated by the light shaded intervals in Figure 12. Let us henceforth refer to this nonequidistant grid as "quasi-uniform" frequency grid. Compared to an equidistant frequency grid of uniform intervals this quasi-uniform grid saves valuable computer memory. Furthermore, the use of quasi-uniform frequency grids presents additional opportunities for optimizing the radiative transfer calculations, because the integration is performed only on the sampling points of the quasi-uniform frequency grid. Details of this integration are, however, beyond the scope of this section. A sequence of O_3 absorption coefficients calculated on quasi-uniform frequency grids at altitudes varying between 8 and 115 km is shown in Figure 13. For the calculation a limb sounding geometry with a tangent height of 8 km has been assumed. Note that the number of sampling points always decreases where the absorption coefficient varies only slowly with frequency. At high altitudes the contributions to absorption are extremely weak and so that spectral lines are truncated.

Figure 14 is a conceptional depiction of the basic steps involved in the proposed algorithm for calculating absorption coefficients. Firstly, the optimum set of sampling points for a given line is determined from the appropriate cutoff tables. Secondly, the sampling points are sorted into increasing order to allow for the rapid evaluation of the Voigt line shape factor. Thirdly, the contributions to absorption are collected into the appropriate intervals residing on a set of successively finer frequency grids. Finally, the coarser intervals are interpolated to the uniform or quasi-uniform frequency grid, respectively.

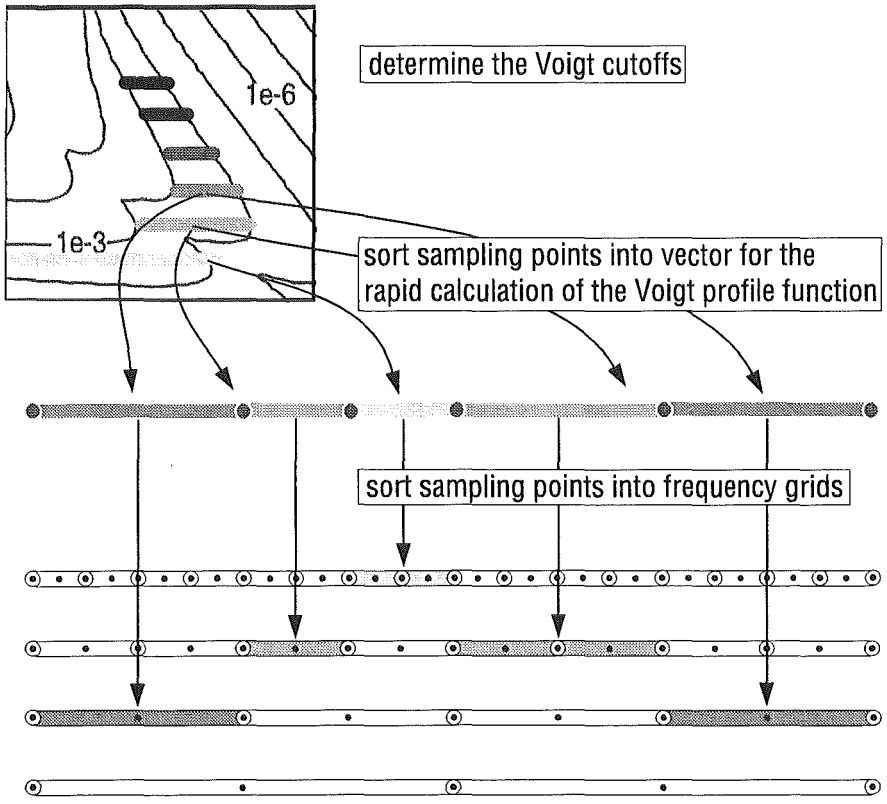


Figure 14: Conceptual depiction of the proposed technique for the efficient calculation of absorption coefficients line-by-line.

6 Implementation of the algorithm

The proposed algorithm has been implemented in ADDLIN, a Fortran90 module of $\sim$ 1000 lines of code. The objective in developing this module was to accommodate flexibility and simplicity in use without substantial loss of accuracy and efficiency. The result was a clear and modular approach that can easily be adapted as a research tool for specific needs separately from its specific use within KOPRA.
Besides the standard calculation of spectral absorption coefficients ADDLIN has been generalized to the application of more sophisticated line shape models. In general, using a different line shape model will require defining new tables of cutoffs analogous to the Lorentz and Doppler cutoffs discussed before. Considering line-mixing, for example, requires defining a separate table of cutoffs since the line shape factor decays asymptotically as $|\nu - \nu_l|^{-1}$ rather than $|\nu - \nu_l|^{-2}$. In practice, however, it may often prove sufficient to continue to use the Lorentz and Doppler cutoffs, especially if line mixing is combined with a line shape model that consists of a Lorentzian profile modified by a corrective factor $\chi_l(\nu, \nu_l)$ (often $\chi_l(\nu, \nu_l)$ takes the form of a decaying exponential with adjustable coefficients determined empirically). ADDLIN has been designed to support both line mixing and corrective $\chi_l(\nu, \nu_l)$-factors basing on the precalculated Lorentz and Doppler cutoffs. Moreover, considering line mixing requires the evaluation of the complex probability function which is closely related to the Voigt profile function. Let us henceforth refer to the complex probability function as $f_l(\nu, \nu_l)$; the Voigt profile function is referred to as real part of the complex probability function, $\mathrm{Re}(f_l(\nu, \nu_l))$, while the imaginary part is referred to as $\mathrm{Im}(f_l(\nu, \nu_l))$. For the calculation of the complex probability function ADDLIN uses the new implementation of the Humlicek algorithm described in section 3 except that the algorithm has been modified to provide also the imaginary part $\mathrm{Im}(f_l(\nu, \nu_l))$ as an option. The following paragraphs are a summary of how absorption coefficients are calculated within ADDLIN if line mixing and/or corrective $\chi_l(\nu, \nu_l)$-factors are taken into account.

Ordinary absorption coefficient: If neither line mixing nor corrective $\chi_l(\nu, \nu_l)$-factors are considered monochromatic absorption coefficient is calculated as

$$\sigma_{s,l}(\nu) = N_s \sum_l S_l \,\mathrm{Re}(f_l(\nu, \nu_l))\,. \tag{15}$$

This is the same expression as (1) on page 44.

Corrective $\chi_l(\nu, \nu_l)$-factors: If line profiles are to be modified by a corrective $\chi_l(\nu, \nu_l)$-factor, the absorption coefficients is evaluated as

$$\sigma_{s,l}(\nu) = N_s \sum_l S_l \,\chi_l(\nu, \nu_l)\,\mathrm{Re}(f_l(\nu, \nu_l))\,, \tag{16}$$

where the corrective $\chi_l(\nu, \nu_l)$-factor has to be provided by the user in form of an external subroutine (for details see the following interface section).

Line mixing: If line mixing is taken into account, the absorption coefficient is calculated as

$$\sigma_{s,l}(\nu) = N_s \sum_l S_l^* \,\mathrm{Re}(f_l(\nu, \nu_l^*)) + y_l^* \,\mathrm{Im}(f_l(\nu, \nu_l^*))\,, \tag{17}$$

where the summation is performed over all lines of the line-mixing branch. The y_l^* are coefficients that have to be provided by the user. KOPRA provides two different line mixing models for the calculation of these coefficients. Note, that depending

on the line mixing model used, the linestrengths, line center frequencies and line widths may also be modified. This is indicated by the asterisks in (17). For more details on line-mixing and the calculation of the y_l^* coefficients see Part VI: 'Line mixing' in this issue.

Corrective $\chi_l(\nu, \nu_l)$-factors and line mixing:
If both corrective $\chi_l(\nu, \nu_l)$-factors and line mixing are taken into account the calculation of absorption coefficients requires two steps. In a first step an averaged corrective $\tilde{\chi}(\nu)$-factor for the whole line-mixing branch is determined from

$$\tilde{\chi}(\nu) = \frac{\sum_l S_l^* \, \chi_l(\nu, \nu_l^*) \, \mathrm{Re}(f_l(\nu, \nu_l*))}{\sum_l S_l^* \, \mathrm{Re}(f_l(\nu, \nu_l^*))}.$$

Note, that the averaged corrective $\tilde{\chi}(\nu)$ may be determined either from the original or the modified line-strengths, line-widths, and line center frequencies (the latter marked by an asterisk). ADDLIN uses the modified line-strengths, line-widths, and line center frequencies as indicated in the above equation. In a second step the averaged corrective $\tilde{\chi}(\nu)$-factor then is multiplied with the result of the pure line-mixing absorption coefficient to give

$$\sigma_{s,l}(\nu) = N_s \, \tilde{\chi}(\nu) \sum_l S_l^* \, \mathrm{Re}(f_l(\nu, \nu_l*)) + y_l \, \mathrm{Im}(f_l(\nu, \nu_l^*)) \,. \tag{18}$$

Note that the two steps are performed automatically, if ADDLIN is invoked appropriately via its user-interface.

7 User interface

The calculation of absorption coefficients within ADDLIN is performed in an almost automated way which enables workers to perform computations quickly without detailed knowledge of the calculation methods involved. Workers may communicate with ADDLIN only via 9 subroutine invocations, which will be described in detail in the following:

`allocate_cutoff(cutdop, cutlor)` Allocates memory for a number of internal arrays and initializes some auxiliary variables. Reads the appropriate look-up-tables for the Doppler- and Lorentz limit, respectively, from the files specified in the strings `cutdop` and `cutlor`. This is the first subroutine to be called by the user. It should be invoked once before the calculation of absorption coefficients starts.

`deallocate_cutoff` This subroutine is the exact counterpart of `allocate_cutoff` and should be invoked once after the calculation of absorption coefficients has been completed.

`allocate_grid (fmin, fmax, fdel)` Determines the spectral range of interest (`fmin`, `fmax`) and the desired spectral resolution `fdel`. Allocates memory for the internal set of successively finer frequency grids. The subroutine should be invoked each time a new spectral range is considered. The units of `fmin`, `fmax`, and `fdel` should be consistent with the units of `alphaL`, `alphaD`, and `f0` in the subroutines `add_lines`, `add_lines_chi`, `add_lines_lm`, and `add_lines_chilm`.

`deallocate_grid` This subroutine is the counterpart of `allocate_grid` and deallocates the arrays specific to a given spectral range of interest. It should be

invoked each time after the calculation of absorption coefficients for a given spectral range has been completed.

`add_lines (nlines, iflag, alphaL, alphaD, ratio, f0, fac, S, dSdT)`
Calculates the absorption coefficient according to (15), i.e. `add_lines` performs the summation of the `nlines` spectral lines with Lorentz half-widths `alphaL`, Doppler half-widths `alphaD`, line center frequencies `f0`, and line strengths `S`. Note that `alphaL` and `alphaD` are defined as the half-width at 1/2 of the maximum of the profile (rather than $1/e$ of the maximum as in the definition of $\alpha_{D,s,l}$ in section 3). As an option, the user may also pass the temperature derivatives of the line strenghts `dSdT`, in which case `add_lines` also provides the temperature derivative of the spectral absorption coefficient as an option. It should be mentioned that the temperature derivative of the absorption coefficient is calculated by summing the temperature derivatives of the line strengths. Temperature dependences of the line-widths e.g. are not taken into account. Test calculations revealed, however, that this approximation is accurate within 5% or better.

`iflag` determines how the calculation of the absorption coefficient is performed in the detail.

- If `iflag = 0`, weak spectral lines may be rejected and those lines retained in the model may be truncated at their line wings.
- If `iflag = 1`, all spectral will be retained in the model and will be considered over the entire frequency range of interest on at least three sampling points. The truncation of lines is suppressed.

`ratio` determines the accuracy to which spectral lines are considered in the model:

- If `ratio` is chosen to be negative, the spectral absorption coefficient is calculated by summing directly the contribution of each line to absorption at each frequency. Note that the calculation of this reference absorption coefficient may be extremely time consuming.
- If `ratio` is chosen to be ratio of the desired absolute accuracy $\delta\sigma$ to which the contribution to absorption of a given line is to be calculated to the number density of the species N_s, i.e. `ratio` $= \delta\sigma/N_s$, the absorption coefficient is calculated in such a way that the absolute accuracy of the contribution to absorption of each spectral line is better than $\delta\sigma$.
- `Ratio` may also be chosen to be the ratio of the desired absolute accuracy $\delta\tau$ to which the contribution of a single line to the optical depth of a given atmospheric layer is to be calculated to the column density $N_s\Delta s$, i.e. `ratio` $= \delta\tau/(N_s\Delta s)$, where Δs is the optical path-length through the layer. In this case the absorption coefficient is determined in such a way that the absolute accuracy to which the contribution of each spectral line is calculated times the optical path-length Δs yields $\delta\tau$.

`fac` is a factor the absorption coefficient is multiplied with.

- If `fac` is set equal to the volume mixing ratio of a given species, the result will be the absorption coefficient itself.
- If `fac` is set equal to the column density, the result will be the optical depth of the given atmospheric layer.
- If `fac` is set equal to 1, the result will be the absorption coefficient divided by the column density.

```
interface                                          ! interface to the user-defined
  subroutine sub(x,n,T,y,alphaL,alphaD)            !   external subroutine SUB
    use precis_m                                   ! definition of precision
    integer, intent(in)  :: n                      ! number of sampling points
    real(dp), intent(in) :: T                      ! temperature
    real(dp), optional   :: alphaD,alphaL          ! Doppler and Lorentz half widths
    real(dp), dimension(*), intent(in)   :: x ! distance from line center
    real(dp), dimension(*), intent(inout):: y ! ratio Lorentz to Doppl. halfwidth
  end subroutine sub
end interface
```

Figure 15: Subroutine interface for the corrective $\chi_l(\nu, \nu_l)$ factor to be provided by the user. For the definition of x and y see section 3.

Note, that the units of `ratio` (e.g. cm^2) and `alphaL`, `alphaD`, and `f0` (e.g. GHz) must be consistent with the units of the line strength `S` (e.g. GHz cm^2) and the units of `fmin`, `fmax`, and `fdel` (e.g. GHz) in `allocate_grid`. Furthermore, the user should take care that if `add_lines` is invoked several times in succession, the optional parameter `dSdT` is either specified or not specified in all subroutine invocations.

`add_lines_chi (nlines, iflag, alphaL, alphaD, ratio, f0, fac, S, sub T, dSdT)` Same as `add_lines` except that each spectral line is multiplied by a corrective $\chi_l(\nu, \nu_l)$-factor according to (16). This corrective $\chi_l(\nu, \nu_l)$-factor) has to be provided by the user in form of an external subroutine, which then is used as an argument (`sub`) in the invocation of `add_lines_chi` and should be declared according to the interface block shown in Figure 15. The input variable `T` is the temperature in K and has to be provided because the user-defined $\chi_l(\nu, \nu_l)$-factor may be temperature-dependent. If the optional parameter `dSdT` is passed, `add_lines_chi` calculates temperature derivatives as an option.

`add_lines_lm (nlines, iflag, alphaL, alphaD, ratio, f0, fac, S, ycoef, dSdT)` Same as `add_lines` or `add_lines_chi` except that line mixing is taken into account according to (17). For this purpose the user has to provide a dimensionless y_l^* coefficient (`ycoef`) for each spectral line of the line-mixing branch (for the calculation of these y_l^* coefficients see Part VI: 'Line mixing' in this issue or [*Funke et al.* 1997]). As an option `add_line_lm` may also calculate the T-derivative of the absorption coefficient depending on whether the optional argument `dSdT` is passed or not.

`add_lines_chilm (nlines, iflag, alphaL, alphaD, ratio, f0, fac, S, sub T, ycoef, dSdT)` takes into account both line mixing and a corrective $\chi_l(\nu, \nu_l)$ factor according to (18). Note, that `add_lines_chilm` performs the two necessary steps, i.e. the calculation of the averaged corrective $\chi_l(\nu, \nu_l)$ factor and its following multiplication with the line mixing absorption coefficient automatically. This enables workers to perform computations quickly without detailed knowledge of the calculation methods involved. Workers only have to make sure, that each invocation of `add_lines_chilm` comprises all spectral lines of a given line-mixing branch. I.e. a given line-mixing branch may not be split into bundles of lines and `add_lines_chilm` invoked several times for each of these bundles. Again, as an option `add_lines_chilm` calculates the temperature derivative of the absorption coefficient, if the optional argument `dSdT` is passed.

`interpolate_grid (iabcomx, iabco, abco, dabcodT, equidistant)` performs the final interpolation of all contributions to absorption calculated on the set

of successively finer frequency grids either to the quasi-uniform frequency grid or the uniform fine grid, depending on whether the optional logical input parameter `equidistant` is passed and true or not. The output variable `iabcomx` indicates how many sampling points are necessary to store the absorption coefficient while `iabco` is a pointer on an integer vector that contains the sampling points. Note, that all sampling point are stored as integers and that the real frequency of the `i`th sampling point is `fmin+iabco(i)*fdel`. The output variables `abco` and `dabcodT` are pointers on the spectral absorption coefficient and its temperature derivative, respectively. The temperature derivative is only calculated if the optional variable `dabcodT` is passed and if the `dSdT` has been passed on in the preceding invocations of `add_lines`, `add_lines_chi`, `add_lines_lm`, or `add_lines_chilm`.

Bibliography

[1] Abramowitz, M. A. and I. A. Stegun, Eds., Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55, 1968.

[2] Clough, S. A. and F. X. Kneizys, Convolution Algorithm for the Lorentz function, Appl. Optics, 18, 2329-2333, 1979.

[3] Clough, S. A., F. X. Kneizys, and R. W. Davies, Line shape and the water vapor continuum, Atmospheric Research, 23, 229-241, 1989.

[4] Edwards, D. P., Modeling of the Atmosphere, SPIE Modeling Atmos., 928, 94-116, 1988.

[5] Fomin, B. A., Effective interpolation technique for line-by-line calculations of radiation absorption in gases, J. Quant. Spectrosc. Radiat. Transfer, 53, 663-669, 1995.

[6] Funke, B., G. P. Stiller, T. von Clarmann, G. Echle, and H. Fischer, CO_2 line mixing in MIPAS limb emission spectra and its influence on retrieval of atmospheric parameters, J. Quant. Spectrosc. Radiat. Transfer, 59, 215-230, 1998.

[7] Gamache, R.R., R.L. Hawkins, L.S. Rothman, Total internal partition sums for atmospheric molecules in the temperature range 70-2005 K: Atmospheric linear molecules, J. Mol. Spectrosc. 142, 205-219, 1990.

[8] Gordley, L. L., B. T. Marshall, and D. A. Chu, LINEPAK: Algorithms for modeling spectral transmittance and radiance, J. Quant. Spectrosc. Radiat. Transfer, 52, 563-580, 1994.

[9] Humlicek, J., Optimized Computation of the Voigt and complex probability fuction, J. Quant. Spectrosc. Radiat. Transfer, 27, 437-444, 1982.

[10] Kuntz, M., A new implementation of the Humlicek algorithm for the calculation of the Voigt profile function, J. Quant. Spectrosc. Radiat. Transfer, 75, 819-824, 1997.

[11] Kuntz, M. and M. Höpfner, Efficient line-by-line calculation of absorption coefficients, J. Quant. Spectrosc. Radiat. Transfer, 63, 97-114, 1999.

[12] Schreier, F., The Voigt and complex error function: A comparison of computational methods, J. Quant. Spectrosc. Radiat. Transfer, 48, 743-762, 1992.

[13] Sparks, L., Accelerated line-by-line calculation of spectral absorption coefficients with high numerical accuracy, in Optical Remote Sensing of the Atmosphere, 2, 1995 OSA Technical Digest Series, 68-70, Optical Society of America, Washington DC, 1995.

[14] Sparks, L., Efficient line-by-line calculation of absorption coefficients to high numerical accuracy, J. Quant. Spectrosc. Radiat. Transfer, 57, 631-649, 1997.

Line mixing

B. Funke

Abstract: The treatment of line mixing within the algorithm is briefly described. KOPRA supports two different approaches, first, the exact treatment by means of a direct diagonalization method (DND) and second, Rosenkranz's fist order approximation. In all cases, a parameterization of the relaxation matrix is calculated externally and read in by the code. A description of the input data format is given.

1 Introduction

The significantly higher spectral resolution of modern instruments leads to an increased sensitivity to the spectral line shapes so that physical effects affecting line shapes such as line mixing (also called collisional narrowing) will be detectable. Especially CO_2 Q-branch line mixing will affect radiance spectra in the spectral region covered because of the high CO_2 volume mixing ratio and narrow line spacing. The effect of CO_2 Q-branch line mixing in the ν_2 fundamental band has already been reported by Strow and Reuter [1]. Line mixing was found to lower atmospheric brightness temperature by as much as 3 K. Another investigation on the impact of line mixing on the retrieval of CCl_4 volume mixing ratio in a spectral range around 790 cm^{-1} including the 11101 $\leftarrow$ 10002 CO_2 Q-branch has shown that the neglection of line mixing maps into an error of volume mixing ratio in the order of 30 - 50 %[2]. The effect is not only observed in CO_2 Q-branches, but also in P- and R-branches of the strong ν_3 fundamental band of CO_2 as well as in Q-branches of other gases such as N_2O and CH_4.
Line mixing occurs when collisions between a radiating molecule and broadening gas leads to a population transfer between the ro-vibrational molecular states and a redistribution of spectral intensity within a band. The intensity of the effect increases with pressure since it is proportional to the broadening gas density. KOPRA offers two different treatments of line shape calculation under consideration of line mixing. The direct diagonalization method (DND) enables exact line mixing calculation while the less time consuming Rosenkranz approximation is only valid for low atmospheric pressures. Therefore the DND approach is recommended for optical pathes through low altitude regions with high pressures such as for ground based observational geometries. The determination of the collisional induced population transfer between the ro-vibrational molecular states expressed by the so-called relaxation matrix is not performed within the code. A parametrisation of relaxation matrix data for the most prominent CO_2 bands is calculated by an external code and read in as input data. Furthermore, the line mixing treatment of KOPRA can easily be extended to other molecules by providing the corresponding input data. KOPRA also supports the calculation of P- and R-branch coupling, provided that input data exists.

2 Theory

2.1 The relaxation matrix W

Within the impact approximation, the absorption coefficient taking account of line mixing depends on the frequency-independent complex relaxation matrix **W** introduced by Ben-Reuven [3]. Its diagonal elements are related to pressure broadening, while nondiagonal elements are related to line mixing. **W** could be calculated using the EPGL (Exponential Power Gap Law) model introduced by Strow et al. [1],[4] and Tobin [5]. It is assumed that the relaxation matrix elements have the same functional form as the rotational state-to-state cross sections within a single vibrational state $\mathbf{K}_{j,k}$ which are modeled using an empirical energy-gap scaling law,

$$\mathbf{K}_{j,k} = a_1 \left(\frac{|\Delta E_{j,k}|}{B_0} \right)^{a_2} exp\left(-a_3 \frac{|\Delta E_{j,k}|}{k_B T} \right) \tag{1}$$

for $j > k$. $\Delta E_{j,k} = E_j - E_k$ is the energy gap between the rotational states j and k, B_0 the rotational constant, k_B the Boltzmann constant and T the kinetic temperature. Detailed balance gives the rates for energetically downward transitions $j < k$. The temperature dependent parameters a_1,a_2 and a_3 of this exponential

power gap law are determined by a least squares fit to the following sum rule:

$$\mathbf{W}_{jj} \approx -\frac{1}{2}\left[\sum_{k\neq j}^{v_i}\beta(v_i,j,k)\mathbf{K}_{j,k} + \sum_{k\neq j}^{v_f}\beta(v_f,j,k)\mathbf{K}_{j,k}\right] \tag{2}$$

where $\mathbf{W}_{jj}$ is given by the Lorentzian half width. The sums are over all rotational states of the initial vibrational level denoted by v_i and the final vibrational level denoted by v_f. β is a symmetry factor which takes into account the propensity differences between $e,f \leftarrow e,f$ and $e,f \leftarrow f,e$ state to state transitions and depends on the vibrational excitation and the angular momentum exchange of the involved states. The values of β were determined empirically by Strow et al. [4]. The nondiagonal elements of W are given by $\mathbf{W}_{j,k} = -\epsilon\mathbf{K}_{j,k}$. $\epsilon = \beta(v_i,j,k)\beta(v_f,j,k)$ depends on the β-factors of both initial and final vibrational level.

2.2 Calculation of the absorption coefficients within the DND approach

The exact calculation of the volume absorption coefficients $\sigma_a^{Vol}(\nu)$ can be performed using a numerical diagonalization procedure [6] (DND) which leads to the following expression for $\sigma_a^{Vol}(\nu)$:

$$\sigma_a^{Vol}(\nu) = \frac{N_g}{\pi}\sum_n \frac{A_n}{\rho_n d_n^2}\frac{Re\Lambda_{nn}}{\nu_n}\frac{1-\exp\frac{-hc\Lambda_{nn}}{kT}}{1-\exp\frac{-hc\nu}{kT}}\frac{Im\Lambda_{nn}Re\Omega_{nn} + Im\Omega_{nn}(\nu - Re\Lambda_{nn})}{(\nu - Re\Lambda_{nn})^2 + (Im\Lambda_{nn})^2} \tag{3}$$

$$\Omega_{nn} = \sum_j d_j \mathbf{T}_{jn}\sum_k \mathbf{T}^{-1}_{nk}\rho_k d_k \tag{4}$$

Here ν_n is the center wavenumber of line n, ν is the wavenumber, N_g the number density of the absorbing gas g, A_n the HITRAN line intensity corrected for temperature T, ρ_k the density of the initial state of the transition k and d_n, d_j the dipole matrix elements of the transitions n and j, respectively. The transformation matrix $\mathbf{T}$ and the diagonal matrix of eigenvalues Λ are determined by the relation

$$\Lambda = \mathbf{T}^{-1}(\vec{\nu_0} - iP\mathbf{W})\mathbf{T} \tag{5}$$

with the vector of line center wavenumbers $\vec{\nu_0}$ and the total pressure P. Since $\mathbf{T}$ and Λ depend on the relaxation matrix $\mathbf{W}$ which is temperature dependent and on the total pressure P the calculation of $\mathbf{W}$ and the determination of $\mathbf{T}$ and Λ is necessary for each atmospheric layer. A polynomal temperature parameterization of the parameters a_1,a_2 and a_3 of equation (1) is used for the implementation in the forward code, based on the precalculated parameters α_i, β_i, γ_i and δ_i:

$$a_i(T) = \begin{cases} \left(\frac{T_0}{T}\right)^{0.75}\left[\alpha_i + \beta_i(T-T_0) + \gamma_i(T-T_0)^2 + \delta_i(T-T_0)^3\right] & \text{for } i = 1 \\ \left[\alpha_i + \beta_i(T-T_0) + \gamma_i(T-T_0)^2 + \delta_i(T-T_0)^3\right] & \text{for } i = 2,3 \end{cases} \tag{6}$$

with $T_0 = 200K$. These parameters are read from the file 'linemix.dat' ($1.4 in the main kopra input file).

2.3 Calculation of the absorption coefficients within the Rosenkranz approximation

The Rosenkranz approximation [7] (RK) which only considers first order line mixing errors requires less computational effort. Within this approximation $\sigma_a^{Vol}(\nu)$ can be written as

$$\sigma_a^{Vol}(\nu) = \frac{N_g}{\pi} \sum_n A_n \left(\frac{\alpha_{L_n} + (\nu - \nu_n) p Y_n}{(\nu - \nu_n)^2 + (\alpha_{L_n})^2} \right) \tag{7}$$

Here ν_n is the center wavenumber of line n, α_{L_n} are the Lorentzian half widths and A_n the line strengths. The first-order line mixing coefficients Y_n are given by

$$Y_n = 2 \sum_{j \neq n} \frac{d_j}{d_n} \frac{\mathbf{W}_{j,n}}{\nu_n - \nu_j} \tag{8}$$

with the dipole matrix elements d_n and d_j. The temperature dependences of Y_n are parameterized similar to Eq. 6 using the precalculated parameters a_n,b_n,c_n and d_n provided by the file 'linemix.dat':

$$Y_n(T) = \left(\frac{200K}{T} \right)^{0.75} \left[a_n + b_n(T - 200K) + c_n(T - 200K)^2 + d_n(T - 200K)^3 \right] \tag{9}$$

The Rosenkranz approximation is only accurate for relatively low pressures. Errors of 2 % in the absorption coefficients compared to calculations using the DND approach occur at Q-branch centers for pressures of 100 hPa and increase up to 12 % at 500 hPa. For this reason KOPRA calculates line-mixing within the Rosenkranz approximation only at atmospheric layers with a pressure below 100 hPa while at layers with a pressure above 100 hPa the exact DND approach is used by default.

2.4 Convolution of line mixing line shape with a Doppler line profile

For atmospheric radiative transfer applications both line shapes must be convolved with a Doppler shape to obtain

$$\sigma_a^{Vol}(\nu) = \frac{N_g}{\pi} \sum_n \frac{\tilde{A}_n}{\alpha_{D_n}} \left[ReW(x_n, y_n) + \tilde{Y}_n ImW(x_n, y_n) \right] \tag{10}$$

$$\tilde{A}_n = A_n, \quad x_n = \frac{\nu - \nu_n}{\alpha_{D_n}}, \quad y_n = \frac{\alpha_{L_n}}{\alpha_{D_n}}, \quad \tilde{Y}_n = pY_n \quad \text{for RK} \tag{11}$$

$$\tilde{A}_n = \frac{A_n}{\rho_n d_n^2} \frac{Re\Lambda_{nn}}{\nu_n} \frac{1 - \exp \frac{-h\Lambda_{nn}}{kT}}{1 - \exp \frac{-h\nu}{kT}} Re\Omega_{nn},$$

$$x_n = \frac{\nu - Re\Lambda_{nn}}{\alpha_{D_n}}, \quad y_n = \frac{Im\Lambda_{nn}}{\alpha_{D_n}}, \quad \tilde{Y}_n = \frac{Im\Omega_{nn}}{Re\Omega_{nn}} \quad \text{for DND} \tag{12}$$

where α_{D_n} is the Doppler half width and $W(x_n, y_n)$ is the complex probability function which is resolved numerically by means of an accelerated Humlicek algorithm [8].

3 Detailed balance requirements

It can be shown that detailed balance requirements on **W** forces

$$\sum_n \tilde{A}_n \tilde{Y}_n = 0 \tag{13}$$

Any violation of this sum rule – produced by numerical errors in line strength calculation or neglection of lines – may result in large errors of the absorption coefficient in the far wing region of the band. For this reason all lines of a given branch for which relaxation matrix elements have been calculated are considered in the forward calculation even if some of them are located outside the spectral interval under consideration. However, an additional correction routine is implemented in the code which calculates $\sum_n \tilde{A}_n \tilde{Y}_n = \Delta$ and recalculates $\tilde{Y}_n^{corr} = \tilde{Y}_n - \Delta$ in order to guarantee Eq. 13.

4 Combined treatment of line mixing and far wing effects

In the case of Q-branch coupling, line mixing is disregarded for microwindows with boundaries farther away than 10 cm^{-1} from all lines of the branch under consideration because of far wing effects caused by the finite duration of collisions. An empirical χ-factor as introduced by Cousin et al. [9][10][11] (see also Part II: 'Analytical expressions') is used instead for all CO_2 lines in order to consider the sub-Lorentzian behavior of the line wings. For P- and R- line mixing, a different approach is chosen since those branches cover spectral intervals much bigger than the region where the impact approximation is valid and hence, both, line mixing and far wing effects have to be considered. In this case, far wing effects are considered by applying a branch correction factor calculated within the code (see also section 1.5 of Part V: 'Absorption coefficients, line collection and frequency grid') to the Rosenkranz or DND line mixing absorption coefficient in the whole spectral region under consideration.

5 The input data file 'linemix.dat'

The input file 'linemix.dat provides line mixing data which is needed for both DND and RK line mixing calculation. The format of the data is specified within the file header. For each branch the following data is repeated:

branch specific data:

Record nb.	Data items
1	IG,VU,VL,BRAN,LI,B0,VMIN,VMAX, BETALF,BETAUF,BETALS,BETAUS
2	AS1A,AS1B,AS1C,AS1D
3	AS2A,AS2B,AS2C,AS2D
4	AS3A,AS3B,AS3C,AS3D
5	AS4A,AS4B,AS4C,AS4D
6	AF1A,AF1B,AF1C,AF1D
7	AF2A,AF2B,AF2C,AF2D
8	AF3A,AF3B,AF3C,AF3D
9	AF4A,AF4B,AF4C,AF4D

IG	HITRAN Molecule-/Isotope number
VU	HITRAN upper global quanta index
VL	HITRAN lower global quanta index
BRAN	branch specifier (P,Q,R)
LI	number of considered lines
B0	rotational constant of isotope
VMIN	wavenumber of first line in branch
VMAX	wavenumber of last line in branch
BETA-L,U-F,S	β-values of lower state(L) and upper state (U) for foreign- (F) and self- (S) broadening
A-S,F-1,2,3,4-A,B,C,D	parameter for **W**-matrix calculation as defined in Eq. 6. S,F = self-/foreign broadening; A,B,C,D = $\alpha, \beta, \gamma, \delta$

line specific data (repeated for each line of the branch):

Record nb.	Data items
1	J,S,YSA,YSB,YSC,YSD,YFA,YFB,YFC,YFD
2	full HITRAN record of this line

J	rotational quantum number of ground state
S	symmetry of lower and upper state, respectively
Y-S,F-1,2,3	parameter for Y-coefficient calculation as defined in Eq. 9. S,F = self-/foreign broadening

Bibliography

[1] L. L. Strow and D. Reuter, "Effect of line mixing on atmospheric brightness temperatures near 15 μm," *Appl. Opt.* **27**(5), pp. 872–878, 1988.

[2] B. Funke, "Modellierung des line–mixing Effekts in Q–Zweigen von CO_2–Banden für die Berechnung von atmosphärischen Infrarotspektren," Master's thesis, Institut für Meteorologie und Klimaforschung, Universität Karlsruhe, Forschungszentrum Karlsruhe, 1995.

[3] A. Ben-Reuven, "Impact broadening of microwave spectra," *Phys. Rev.* **145**(1), pp. 7–22, 1966.

[4] L. L. Strow, D. C. Tobin, and S. E. Hannon, "A compilation of first–order line–mixing coefficients for CO_2 Q–branches," *J. Quant. Spectrosc. Radiat. Transfer* **52**(3/4), pp. 281–294, 1994.

[5] D. C. Tobin, "Carbon dioxide lineshapes in $\Pi \leftarrow \Sigma$ and $\Pi \leftarrow \Delta$ vibrational transitions," Master's thesis, University of Maryland Baltimore County, 1993.

[6] R. Gordon and R. P. McGinnis, "Line shapes in molecular spectra," *J. Chem. Phys.* **49**, p. 2455ff, 1968.

[7] P. W. Rosenkranz, "Shape of the 5 mm oxygen band in the atmosphere," *IEEE Transactions on Antennas and Propagation* **AP–23**(4), pp. 498–506, 1975.

[8] M. Kuntz, "A new implementation of the Humlicek algorithm for the calculation of the Voigt profile function," *J. Quant. Spectrosc. Radiat. Transfer* **57**(6), pp. 819–824, 1997.

[9] R. Le Doucen, C. Cousin, C. Boulet, and A. Henry, "Temperature dependence of the absorption in the region beyond the 4.3–μm band head of CO_2. 1: Pure CO_2 case," *Appl. Opt.* **24**(6), pp. 897–906, 1985.

[10] V. Menoux, R. Le Doucen, and C. Boulet, "Line shape in the low–frequency wing of self–broadened CO_2 lines," *Appl. Opt.* **26**(3), pp. 554–562, 1987.

[11] V. Menoux, R. Le Doucen, J. Boissoles, and C. Boulet, "Line shape in the low frequency wing of self– and N_2–broadened ν_3 CO_2 lines: temperature dependence of the asymmetry," *Appl. Opt.* **30**(3), pp. 281–286, 1991.

Cross-sections of heavy molecules and pseudo-lines

S. Zorn, T. von Clarmann, M. Höpfner, G. P. Stiller, N. Glatthor and A. Linden

Abstract: Absorption coefficients of heavy molecules such as CFC-11, CFC-12, CFC-14, CFC-113, CFC-114, HCFC-22, HCFC-123, HCFC-124, HCFC-141b, HCFC-142b, CCl_4, SF_6, $ClONO_2$, N_2O_5, CH_3COCH_3 (acetone), $CH_3CO_3NO_2$ (peroxyacetylnitrate, PAN), CH_3OH (methanol), CH_3CCl_3 (methylchloroform), C_5H_8 (isoprene), C_3H_6 (propene), DMS (dimethylsulfide), H_2CO (formaldehyde), CH_3Br (methylbromide), and C_6H_6 (benzene) are calculated by interpolation of reference cross-section spectra rather than line-by-line calculation. For most species interpolation in pressure and temperature follows a summation of selected reference data weighted by the inverse quadratic distance from the target point in the pressure temperature plane. For $ClONO_2$ two cross-section sources are implemented: for those by *Ballard et al.*[1] no pressure-dependence is considered, and temperature-dependence of the absorption is scaled following an 1/temperature law. Pressure and temperature dependent cross-sections of $ClONO_2$ by *M. Birk (pers. com., 2000)* are also supported. Additionally, for many of the heavy molecules a pseudo-line treatment is implemented in KOPRA.

1 Cross-sections

1.1 Introduction

Heavy molecules give raise to very dense, often continuum-like spectra of which single rotational transitions are generally not resolved. This applies to CFC-11 (CCl_3F), CFC-12 (CCl_2F_2), CFC-14 (CF_4), CFC-113 ($C_2Cl_3F_3$), CFC-114 ($C_2Cl_2F_4$), HCFC-22 ($CHClF_2$), HCFC-123 (CF_3CHCl_2), HCFC-124 (CF_3CHFCl), HCFC-141 ($CFCl_2CH_3$), HCFC-142 (CF_2ClCH_3), CCl_4, SF_6, $ClONO_2$, N_2O_5, acetone (CH_3COCH_3), $CH_3CO_3NO_2$ (peroxyacetylnitrate, PAN), CH_3OH (methanol), CH_3CCl_3 (methylchloroform), C_5H_8 (isoprene), C_3H_6 (propene), DMS (dimethylsulfide), H_2CO (formaldehyde), CH_3Br (methylbromide), and benzene (C_6H_6). Since spectroscopic data are not available for single transitions, and for reasons of efficient computing, absorption coefficients of these species for given pressure and temperature are generated by two-dimensional interpolation of pretabulated reference cross section spectra in pressure and temperature rather than line-by-line calculations.

1.2 Interpolation Procedure

Laboratory measurements of absorption cross sections of CFC's, HCFC's, and other heavy molecules at various pressures and temperatures are provided by *Varanasi et al.*[2, 3, 4, 5, 6], *Li et al.*[7], *Massie et al*[8], *McDaniel et al.*[9], *Clerbaux et al.*[10], *Hanst et al.*[11], and others. They cover typical stratospheric temperatures and pressures (Table 1).
As a first step, the availability of reference spectra in the vicinity of the target pressure and temperature are checked: There are four cases to be distinguished, dependent on how in many quadrants in the pressure temperature plane reference cross section data are available:

(a) **1 Quadrant**: The target pressure and temperature point is ouside the range of both temperatures and pressures covered by reference measurements. See Fig. 1 A.

(b) **2 Quadrants**: The target pressure and temperature point is outside the range of values covered by the reference data either for pressure or for temperature

(c) **3 Quadrants**: The target pressure and temperature point is situated near a concave boundary of the aerea covered by the reference measurements

(d) **4 Quadrants**: The target pressure and temperature point is situated in the aerea covered by reference measurements.

At a next step from each "occupied" quadrant one reference measurement is selected. The selection criterion is

$$d = \sqrt{(T_{ref} - T_{target})^2 + (100 \times \frac{p_{ref} - p_{target}}{p_{target}})^2} = min. \quad (1)$$

The selected reference cross section spectra then are interpolated linearily onto a common wavenumber grid. The interpolation in the pressure and temperature plane is depends on the number of quadrants in the pressure temperature plane where reference spectra are available.

1.2.1 One Quadrant

If reference spectra are available in one quadrant in the pressure temperature plane (see Fig. 1 A), the reference cross-section spectrum selected by means of Eq. 1 is used. No extrapolation is performed. This case applies also to species where only one measurement is available.

1.2.2 Two Quadrants

If reference spectra are available in two quadrant in the pressure temperature plane, two subcases have to be considered (see Figs. 1 B and E). In the case of occupation of two adjacent quadrants (Fig. 1 B), linear interpolation is performed in the quantity where one value higher and one value lower than the target value is available, while for the other quantity no extrapolation is performed. This typically applies to situations where the reference cross-sections do not cover the pressure or temperature range needed, or species where only temperature but no pressure dependent cross-sections are available (CFC-113, CFC-114, $ClONO_2$, N_2O_5). In the case of occupation of two opposite quadrants (Fig. 1 E), the same scheme as for the case of three or four quadrants is used.

Molecule	Spectral Intervals [cm^{-1}]	Temperatures [K]	Pressures [torr]	References
CFC-11	810–880 1050–1120	201–296 201–296	40–760 40–760	Li and Varanasi, 1994[7]
CFC-12	800–950 1040–1200	216–296 216–296	170–760 170–760	Varanasi and Nemtchinov, 1994[6]
CFC-14	1250–1289	180–296	22–760	Varanasi, 1997 pers. com. Massie et al.,1991[8] McDaniel et al., 1991[9] Husson et al., 1992[12] Husson et al., 1994[13]
CFC-113	780–995 1005.5–1232	203–293 203–293		Husson et al., 1992[12] Husson et al., 1994[13] Massie et al.,1991[8] McDaniel et al., 1991[9]
CFC-114	815–860 870–960 1030–1067 1095–1285	203–293		Husson et al., 1992[12] Husson et al., 1994[13] Massie et al., 1991[8] McDaniel et al., 1991[9]
HCFC-22	750–870 1060–1209 1275–1380	216–294 216–294 216–294	40–760 40–760 40–760	Varanasi et al., 1994[5] Clerbaux et al., 1993[10]
HCFC-123	740–900 1080–1450	253, 270, 287		Clerbaux et al., 1993 [10]
HCFC-124	675–715 790–920 1035–1430	287 (pure vapor)		Clerbaux et al., 1993 [10]
HCFC-141b	710–790 990–1210 1325–1470	253, 270, 287		Clerbaux et al., 1993 [10]
HCFC-142b	650–705 875–1030 1075–1265 1360–1475	253, 270, 287		Clerbaux et al., 1993 [10]
CCl_4	770–810	170–310		Orlando et al., 1992[14]
SF_6	925–956	216–295	25–760	Varanasi et al., 1994[5]
$ClONO_2$	740–840 1240–1340 1680–1790	213 and 296		Ballard et al., 1988[1]
$ClONO_2$	690–1330	190–297	0–113	M. Birk, 2000 pers. com.
N_2O_5	550–600 720–765 1210–1275 1680–1765	233–293		Massie et al., 1985[15]
CH_3COCH_3	700–2000	298		Remedios, 2000 pers. com. from EPA-webside
CH_3COCH_3	500–3700	298	760	Hanst, 2000[11]
C_6H_6	600–1800	295		Remedios, 2000 priv. com. measured by P. Sinclair
$CH_3CO_3NO_2$	531–3653	298	760	Hanst, 2000[11]
CH_3OH	500–3740	298	760	Hanst, 2000[11]
CH_3CCl_3	500–3700	298	760	Hanst, 2000[11]
C_5H_8	500–3700	298	760	Hanst, 2000[11]
C_3H_6	500–3700	298	760	Hanst, 2000[11]
DMS	500–3700	298	760	Hanst, 2000[11]
H_2CO	500–3700	298	760	Hanst, 2000[11]
CH_3Br	500–3700	298	760	Hanst, 2000[11]

Table 1: Molecules handled by tabulated cross section data, and their data sources

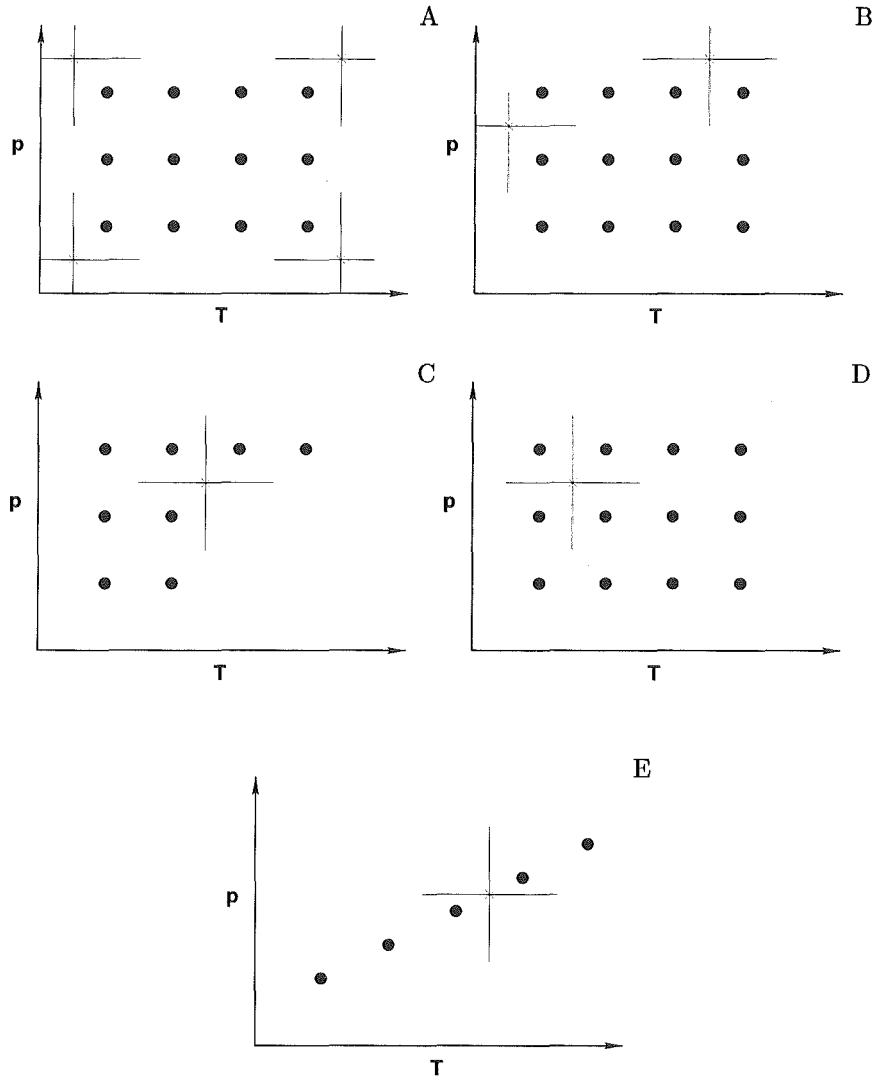


Figure 1: Definition of the cases 1.2(a) to 1.2(d)

1.2.3 Three and Four Quadrants

The interpolation in the pressure temperature plane is performed by summing up the selected reference data from each occupied quadrant weighted by the inverse quadratic distance from the target point. The distance d_n of the target point to the reference point in the pressure temperature plane is defined as

$$d_n = \sqrt{(\frac{T_{target} - T_{reference}}{w_T})^2 + (\frac{p_{target} - p_{reference}}{p_{target} \times w_p})^2} \qquad (2)$$

where w_T and w_p are factors used for weighting temperature against pressure when transforming both quantities to the dimensionsless, which is necessary, because a scalar distance is not defined for quantities of different dimensions, and the result would depend on the physical units applied. The weighting factor is directly determined from the pressure and temperature dependence of reference laboratory

cross-sections. The calculation of cross-section spectrum $X(p_{target}, T_{target})$ is performed as follows:

$$X(p_{target}, T_{target}) = \frac{1}{\sum_{n=1}^{N} \frac{1}{d_n^2}} (\sum \frac{X_n}{d_n^2}) \tag{3}$$

where N is the number of occupied quadrants, and X_n is the cross-section spectrum in the nth occupied quadrant.

1.2.4 Special Case: $ClONO_2$

(a) **Ballard's data:**

Formally $ClONO_2$ cross-sections are handled as described in the standard two-quadrants case. The only difference is that cross-section reference spectra for various temperatures are no lab spectra but have been precalculated from reference data as

$$X_m(T) = \frac{a_m}{T} + b_m \tag{4}$$

where m denotes the spectral gridpoint, and coefficients a_m and b_m have been chosen to fit the reference data provided by *Ballard et al.*[1] for 213 and 296 K.

(b) **Birk's data:**

M. Birk (pers. com., 2000) measured temperature and pressure dependent cross-sections of $ClONO_2$. These were parameterized and delivered together with an p-T interpolation scheme. Making use of this routine the cross-sections were calculated at 7 temperatures between 189K and 297K and at 8 pressures between 0 and 71torr. KOPRA uses these pre-calculated cross-sections to perform it's own 2-d interpolation as described above.

2 Pseudo-lines

2.1 Introduction

In order to enable the handling of cross-section in line-by-line codes and to perform the interpolation procedure more 'physically', pseudo-linelists were produced by *G. Toon (pers. com., 1997)*. For this purpose pseudo-lines were made by fitting line-parameters to the laboratory cross-section measurements. In the following we first quote the description of the pseudo-linelists and then describe the implementation into KOPRA.

2.1.1 Pseudo-linelists

The 'readme' file which was distributed together with the pseudo-linelists reads:

> "... Each pseudo-linelist was derived by fitting all of the relevant laboratory spectra simultaneously while solving for the 296K strength and the Ground State Energy (E") of each pseudo-line. The pressure-broadened half-width (PBHW) and its temperature dependence were determined "manually", by trying various values and selecting the ones that gave the best overall fit. Generally, for gases without sharp absorption features, the goodness of fit was insensitive to the choice of PBHW, whereas for gases like CFC-12 and HCFC-22 which have sharp Q-branches, the right

choice of PBHW is important. Note that for some gases (e.g. CFC-12) the resulting value for the temperature- dependence of the PBHW (0.0) is well outside the normal range (0.5 to 0.8). All lines in a given absorption band were assumed to have the same PBHW and temperature dependence.

The idea of using pseudo-lines to represent broad featureless absorption bands is not new. However, whereas previously workers minimized the number of lines needed by ascribing them an exaggerated PBHW, we have achieved the same goal by giving each pseudo-line an exaggerated Doppler width. The advantage of this latter approach is that it allows the correct PBHW to be employed, so that a realistic pressure-dependence can still be simulated, even in cases when all of the laboratory spectra were measured at low pressure (e.g. CF_4).

These lists are not intended to supplant proper quantum-mechanically-based linelists. They were derived primarily as a convenient means of interpolating (and extrapolating) the laboratory cross-sections to temperatures and pressures where actual measurements are unavailable (I could not think of a realistic way of doing this directly from the cross-sections). However, in deriving and using these pseudo-linelists, several additional advantages became apparent:

(a) Since the pseudo-linelists are in the HITRAN format, they can be accessed in exactly the same manner as all the regular gases, avoiding special code to read the raw cross-section spectra and interpolate them to the desired temperatures and pressures.

(b) Fitting a physically-based function to the laboratory spectra also serves as a quality control measure: Since we are typically trying to determine just two unknowns (S & E") from 4-30 spectra, the problem is over-determined and so performing the fit provides an assessment of the consistency of the various laboratory spectra. This makes it possible to identify and reject any laboratory spectra which are inconsistent with the others, or even to quantify biases between different sets of laboratory spectra, perhaps measured under very different conditions. Furthermore, the retrieval of unphysical (i.e. -ve) values of S and E" provides a warning that serious problems exist.

(c) The laboratory cross-sections are always convolved with the Instrument Line Shape (ILS) of the laboratory spectrometer. In making a pseudo-linelist, the effects of this ILS is removed, since it is included in the forward model which calculates the cross-sections from the pseudo-lines. This is particularly important if the atmospheric spectra are measured at a better resolution than the laboratory spectra.

(d) Several different laboratory data-sets, even with widely different measurement conditions and spectral resolutions, can easily be assimilated into a single pseudo-linelist.

(e) At the end of the fitting process, the pseudo list can be checked by comparing the forward model calculation (which uses the pseudo-lines) with the measured laboratory spectra. Of course, the agreement will not be perfect since the fit was overconstrained, but the difference are usually $< 1\%$.

(f) Since all the pseudo-lines in a given band are assumed to have the same PBHW and Doppler widths, only one evaluation of the Voigt lineshape is necessary to compute the absorption spectrum from all the psuedo-lines (provided that this lineshape is stored). Thus, the speed of using the pseudo-linelists is competitive with 2-D interpolation in the raw cross-sections (assuming one knew a good way of doing this).

Spacing of pseudo-lines:

The choice of line spacing for the pseudo-lines was somewhat arbitrary. We tried to make it as wide as possible to minimize the total number of lines, yet still resolve any structure observed in the laboratory spectra. Typically, the line spacing was chosen to be similar to the resolution of the laboratory spectra. Note that the positions and spacing of the pseudo-lines are completely inde- pendent of the spectral frequencies in the laboratory spectra. This fact makes it possible to simultaneously fit different sets of laboratory spectra.

Most of the pseudo-linelists are spaced at 0.01 cm^{-1}, which is ten times larger than an actual Doppler widths of most heavy gases. While this would not be a problem in the troposphere where the pressure broadening would cause the pseudo- lines to overlap, in the upper stratosphere a high resolution computed spectrum would show narrow lines with large gaps between. To avoid this problem one must artificially increase the Doppler width until it approximately matches the line spacing. A convenient way of doing this is to set the molecular weight to an artificially small value (e.g. 1), However, this has drawbacks if one wants to use pseudo-lines, together with real quantum-mechanical lines, of the same gas in the same interval (e.g for $ClONO_2$ it is probably advisable to replaced the 760-798 cm^{-1} section of the pseudo-line list with Bell's linelist). Therefore, in all of the pseudo linelists, we have defined the isotope number to be zero. This allows pseudo lines to be easily distinguished from real lines (which have isotope numbers 1-9), and could allow the line-by-line code to explitly set the Doppler width equal to the line spacing whenever it encounters pseudo-lines, avoiding the need to fudge the molecular weight. This is especially helpful for gases like $ClONO_2$ for which a proper linelist (requiring the actual molecular weight) exists for the region around the 780 cm^{-1} Q-branch, but pseudo lines must be used for other regions.

Line strength:

The following expression for line strength was assumed in the derivation of the pseudo-linelists:

$$S(T) = S(296)\left(\frac{296}{T}\right)^{tdrpf} \frac{Qvib(T)}{Qvib(296)} \frac{SE(T)}{SE(296)} \exp\left(hcE''(1/296 - 1/T)\right) \tag{5}$$

where $Qvib(T) = \prod[1 - \exp(-hc\nu_j/kT)]$ is the vibrational partition function and the product is performed over all the vibrational frequencies, ν_j. $SE(T) = [1 - \exp(-hc\nu_i/kT)]$ is the correction for the Stimulated Emission, ν_i being the center frequency of the line in question. The term $(296/T)^{tdrpf}$ is commonly known as the rotational partition function and tdrpf is usually (1.0, 1.5, or 2.0).

File	GAS	Interval	Spacing	Lines	Error	Measurer
cf4.h92	CF4	1250-1295	0.005	9001	4%	Varanasi
f12.h92	CFC-12	850- 950	0.010	10000	2%	Varanasi
	CFC-12	1050-1200	0.010	15000	1%	Varanasi
f11.h92	CFC-11	810- 880	0.010	7000	7%	Varanasi
	CFC-11	1050-1120	0.010	7000	6%	Varanasi
ccl4.h92	CCl4	770- 810	0.010	4001	2%	Orlando (omitted 170K)
f22.h92	CHF2Cl	776- 850	0.00742	9977	5%	Varanasi & McDaniel
	CHF2Cl	1080-1150	0.010	7001	2%	McDaniel
	CHF2Cl	1290-1335	0.010	4501	2%	McDaniel
f113.h92	CFC-113	786- 990	0.500	408	8%	McDaniel (omitted 203K)
sf6.h92	SF6	925- 955	0.010	3001	2%	Varanasi
f142b.h92	HCFC-142b	870-1270	0.010	40000	4%	Newnham
clno3.h92	ClONO2	750- 831	0.00964	8401	0%	Ballard (only 2 spectra)
n2o5.h92	N2O5	547- 610	0.160	373	4%	NCAR
	N2O5	709- 775	0.210	315	2%	NCAR
	N2O5	1194-1281	0.350	266	2%	NCAR
	N2O5	1663-1793	0.480	271	2%	NCAR

Table 2: Computed pseudo-linelists

The term $\exp(hcE"(1/296 - 1/T))$ is simply the Boltzmann factor, E" being the ground-state energy.

Most forward models should already have the code to compute the above expression because they are needed for the lighter gases. So it should be a simple matter to extend this capability to the heavy gases. Note that the same expression for S(T) was used for the fitting of the laboratory spectra, so the derived values of S(296) and E" values will only correctly reproduce the laboratory spectra provided that the user employs the same expressions.

Finally, we want to make it clear that one should not expect the forward calculation made using these pseudo-lines to agree perfectly (except $ClONO_2$) with individual laboratory spectra, since the pseudo-lines were derived from an over-determined fit to ALL of the laboratory spectra. Differences will arise from noise on the laboratory spectra, and uncertainties in the measurement conditions (T,P,vmr), in addition to inadequacies in the pseudo-line approach.

In the table 2 we summarize the gases and spectral intervals for which we have computed pseudo-linelists. We estimated the maximum error in absorber found in re-fitting the laboratory spectra using the final pseudo-linelist. Note that at specific frequencies, the error in the computed absorption coefficient may well exceed these tabulated values.

Please direct any questions or comments to:
Geoff Toon 818 354 8258 toon@mark4sun.jpl.nasa.gov
or Bhaswar Sen 818 354 1165 sen@mark4sun.jpl.nasa.gov"

2.1.2 KOPRA-implementation of pseudo-linelists

The gas-numbering of the pseudo-line species in KOPRA is actually from 251-260 (see the gas-list in the KOPRA main input file in section 'KOPRA installation' for the detailed numbers).
The pseudo-lines are integrated in the KOPRA spectroscopic line list with their KOPRA numbering (the record length is therefore 101 instead of 100 in the HITRAN case).
In `line_parameters@abco_m` the Doppler width is set to the line-spacing for pseudo-lines.
For calculating the partition functions (vibrational times rotational) KOPRA only uses the polynomial parameterization of the temperature dependence by Gamache. Therefore, for the pseudo-line molecules these coefficients were determined by a fit to the temperature dependence resulting from the explicit formulas as given in the previous section.

Bibliography

[1] J. Ballard, W. B. Johnston, M. R. Gunson, and P. T. Wassell, "Absolute absorption coefficients of $ClONO_2$ infrared bands at stratospheric temperatures," *J. Geophys. Res.* **93**(D2), pp. 1659–1665, 1988.

[2] P. Varanasi, "Absorption coefficients of CFC–11 and CFC–12 needed for atmospheric remote sensing and global warming studies," *J. Quant. Spectrosc. Radiat. Transfer* **48**(2), pp. 205–219, 1992.

[3] P. Varanasi, "Absorption spectra of HCFC–22 around 829 cm^{-1} at atmospheric conditions," *J. Quant. Spectrosc. Radiat. Transfer* **47**(4), pp. 251–255, 1992.

[4] P. Varanasi, A. Gopalan, and J. F. Brannon, Jr., "Infrared absorption–coefficient data on SF_6 applicable to atmospheric remote sensing," *J. Quant. Spectrosc. Radiat. Transfer* **48**(2), pp. 141–145, 1992.

[5] P. Varanasi, Z. Li, V. Nemtchinov, and A. Cherukuri, "Spectral absorption–coefficient data on HCFC–22 and SF_6 for remote–sensing applications," *J. Quant. Spectrosc. Radiat. Transfer* **52**(3/4), pp. 323–332, 1994.

[6] P. Varanasi and V. Nemtchinov, "Thermal infrared absorption coefficients of CFC–12 at atmospheric conditions," *J. Quant. Spectrosc. Radiat. Transfer* **51**(5), pp. 679–687, 1994.

[7] Z. Li and P. Varanasi, "Measurement of the absorption cross–sections of CFC–11 at conditions representing various model atmospheres," *J. Quant. Spectrosc. Radiat. Transfer* **52**(2), pp. 137–144, 1994.

[8] S. T. Massie, A. Goldman, A. H. McDaniel, C. A. Cantrell, J. A. Davidson, R. E. Shetter, and J. G. Calvert, "Temperature dependent infrared cross sections for CFC–11, CFC–12, CFC–13, CFC–14, CFC–22, CFC–113, CFC–114, and CFC–115," tech. rep., NCAR, 1991. Technical Note, TN–358+STR.

[9] A. H. McDaniel, C. A. Cantrell, J. A. Davidson, R. E. Shetter, and J. G. Calvert, "The temperature dependent infrared cross sections for CFC–11, CFC–12, CFC–13, CFC–14, CFC–22, CFC–113, CFC–114, and CFC–115," *J. Atmos. Chem.* **12**, pp. 211–227, 1991.

[10] C. Clerbaux, R. Colin, P. C. Simon, and C. Granier, "Infrared cross-sections and global warming potentials of 10 alternative hydrohalocarbons," *J. Geophys. Res.* **98**(D6), pp. 10491–10497, 1993.

[11] L. Hanst, *Infrared Analysis, Inc.*, http://www.Infrared-Analysis.com, 2000.

[12] N. Husson, B. Bonnet, N. A. Scott, and A. Chedin, "Management and study of spectroscopic information: The GEISA program," *J. Quant. Spectrosc. Radiat. Transfer* **48**(5–6), pp. 509–518, 1992.

[13] N. Husson, B. Bonnet, A. Chédin, N. A. Scott, A. A. Chursin, V. F. Golovko, and V. G. Tyuterev, "The GEISA data bank in 1993. A PC/AT compatible computers new version," *J. Quant. Spectrosc. Radiat. Transfer* **52**, pp. 425–438, 1994.

[14] J. J. Orlando, G. S. Tyndall, A. Huang, and J. G. Calvert, "Temperature dependence of the infrared absorption cross sections of carbon tetrachloride," *Geophys. Res. Lett.* **19**(10), pp. 1005–1008, 1992.

[15] S. T. Massie, A. Goldman, D. G. Murcray, and J. C. Gille, "Approximate absorption cross sections of F12, F11, $ClONO_2$, N_2O_5, HNO_3, CCl_4, CF_4, F21, F113, F114, and HNO_4," *Appl. Opt.* **24**, pp. 3426–3427, 1985.

Parameterization of continua caused by gaseous constituents

G. Echle and M. Höpfner

Abstract: The calculation of broadband continua caused by atmospheric trace gases like O_2, N_2, H_2O, CO_2 is described. The collision induced bands of O_2 and N_2, due to interaction of molecule pairs, are simulated by empirical models from literature. For H_2O the well know CKD continuum is used where a special treatment of the line wing calculation is necessary. For CO_2 an own continuum was determined, taking into account contributions more than $25cm^{-1}$ apart from the line-center.

1 Introduction

The emission of the terrestrial atmosphere in the infrared spectral region is dominated by vibrational-rotational bands of a large number of atmospheric trace species such as H_2O, CO_2, O_3, CH_4, N_2O. In the case of measuring their emission with high spectral resolution the individual vibrational-rotational lines of most trace species can be resolved. In addition to the molecular lines, absorption features extending over wide spectral regions and varying slowly with frequency are present in atmospheric infrared emission spectra. This so-called *continuum* radiance has its origin in different physical mechanisms:

- collision-induced absorption bands due to the interaction of molecule pairs like N_2-N_2, N_2-O_2, O_2-O_2,
- superposition of line wings from strongly absorbing bands such as the 6.3 μm ν_2 band of H_2O and the 4.3 μm ν_3 band of CO_2

The different gaseous continua are calculated in the module **gascon_m** which include calls of the individual subroutines for the different continua.

2 O_2 Continuum

The O_2 Continuum is calculated by the subroutine **calc_o2cont@gascon_m**. The implementation is based on the empirical model by [1]. They combined laboratory measurements of normalized binary absorption coefficients $B_{O_2-O_2}$ and $B_{N_2-O_2}$ to the normalized absorption coefficient B_{O_2-air} assuming a composition of 21% O_2 and 79% N_2. The temperature and wavenumber dependence of the normalized absorption coefficient for O_2-air is modeled using a simple empirical law:

$$B_{O_2-air}(\nu,T) = B^0_{O_2-air}(\nu)\exp\left[\beta^0_{O_2-air}(\nu)\left(\frac{1}{T_0}-\frac{1}{T}\right)\right] \tag{1}$$

with the reference temperature $T_0 = 296$ K and the normalized absorption coefficient $B^0_{O_2-air}$ and an empirically determined exponent for the temperature dependence $\beta^0_{O_2-air}$ at reference temperature, both tabulated as a function of wavenumber in the wavenumber region 1365 cm^{-1} to 1800 cm^{-1} in steps of 5 cm^{-1}. The normalized absorption coefficient $B^0_{O_2-air}$ and the exponent for the temperature dependence $\beta^0_{O_2-air}$ are given in units of [cm^{-1} Am^{-2}] and [K], respectively. For calculation of the O_2 continuum at arbitrary wavenumbers these coefficients are linearly interpolated. The absorption cross-section σ_{O_2} of the O_2 continuum in units of [cm^2/molec] can then be calculated by:

$$\sigma_{O_2}(\nu,\rho_{O_2},T) = \frac{\rho_{O_2}\times B^0_{O_2-air}(\nu)}{(0.21\ Am)^2}\times\exp\left[\beta^0_{O_2-air}(\nu)\left(\frac{1}{T_0}-\frac{1}{T}\right)\right] \tag{2}$$

where ρ_{O_2} is the density of O_2 in units of [molec/cm^3].
The subroutine **calc_o2cont@gascon_m** also supports the calculation of the derivative of the O_2 absorption cross section with respect to temperature given by:

$$\begin{aligned}\frac{d\sigma_{O_2}}{dT}(\nu,\rho_{O_2},T) &= \frac{\rho_{O_2}\times B^0_{O_2-air}(\nu)}{(0.21\ Am)^2}\times\frac{\beta^0_{O_2-air}(\nu)/T-1}{T} \\ &\quad\times\exp\left[\beta^0_{O_2-air}(\nu)\left(\frac{1}{T_0}-\frac{1}{T}\right)\right]\end{aligned} \tag{3}$$

3 N_2 Continuum

The N_2 Continuum is calculated by the subroutine **calc_n2cont@gascon_m**. The implementation is based on the empirical model by [2] The temperature and wavenumber dependence of the normalized absorption coefficient $B_{N_2-N_2}$ for pure N_2 is modeled using a simple empirical law:

$$B_{N_2-N_2}(\nu,T) = B^0_{N_2-N_2}(\nu)\exp\left[\beta^0_{N_2-N_2}(\nu)\left(\frac{1}{T_0}-\frac{1}{T}\right)\right] \tag{4}$$

with the reference temperature $T_0 = 296$ K and the normalized absorption coefficient $B^0_{N_2-N_2}$ and an empirically determined exponent for the temperature dependence $\beta^0_{N_2-N_2}$ at reference temperature, both tabulated as a function of wavenumber in the wavenumber region 2125 cm^{-1} to 2600 cm^{-1} in steps of 5 cm^{-1}. The normalized absorption coefficient $B^0_{N_2-N_2}$ and the exponent for the temperature dependence $\beta^0_{N_2-N_2}$ are given in units of [cm^{-1} Am^{-2}] and [K], respectively. For calculation of the N_2 continuum at arbitrary wavenumbers these coefficients are linearly interpolated. The relative efficiency $E^{N_2}_{O_2/N_2}$ of collisions of N_2 with O_2 and N_2 molecules can be modeled by the following wavenumber-independent expression:

$$E^{N_2}_{O_2/N_2}(T) = 1.294 - 0.4545(T/T_0) \tag{5}$$

The N_2-air collision-induced spectra in the Earth's atmosphere from the pure N_2 data can then be calculated by:

$$B_{N_2-air}(\nu,T) = [0.79 + 0.21 \times E^{N_2}_{O_2/N_2}(T)] \times B_{N_2-N_2}(\nu,T) \tag{6}$$

The absorption cross-section σ_{N_2} of the N_2 continuum in units of [cm^2/molec] considering the composition of air by 79% N_2 and 21% O_2 can then be calculated by:

$$\begin{aligned}\sigma_{N_2}(\nu,\rho_{N_2},T) &= \frac{\rho_{N_2} \times B^0_{N_2-N_2}(\nu)}{(0.79\ Am)^2} \times [0.8387 - 0.0754\frac{T}{T_0}] \\ &\quad \times \exp\left[\beta^0_{N_2-N_2}(\nu)\left(\frac{1}{T_0} - \frac{1}{T}\right)\right]\end{aligned} \tag{7}$$

where ρ_{N_2} is the density of N_2 in units of [molec/cm^3].
The subroutine **calc_n2cont@gascon_m** also supports the calculation of the derivative of the N_2 absorption cross section with respect to temperature given by:

$$\begin{aligned}\frac{d\sigma_{N_2}}{dT}(\nu,\rho_{N_2},T) &= \frac{\rho_{N_2} \times B^0_{N_2-N_2}(\nu)}{0.79\ Am^2} \times \exp\left[\beta^0_{N_2-N_2}(\nu)\left(\frac{1}{T_0} - \frac{1}{T}\right)\right] \\ &\quad \times \left[\frac{0.8387 - 0.0754\frac{T}{T_0}}{T}\left(\frac{\beta^0_{N_2-N_2}(\nu)}{T} - 1\right) - \frac{0.0754}{T_0}\right]\end{aligned} \tag{8}$$

4 H_2O Continuum

The H_2O Continuum is calculated by the subroutine **calc_h2ocont@gascon_m**. The implementation is based on the CKDv2.2 model by [3] This model introduces an empirical lineshape correction factor (χ-factor) which accounts for the non-Lorentzian behaviour of H_2O lines. The specific realization of Clough's continuum parameterization is coupled with the definition of the continuum: The absorption of all water vapor transitions is separated into the *local absorption* and the *continuum absorption*. The local absorption is defined as a Lorentzian lineshape out to $\pm$ 25 cm^{-1} apart form the line center, minus the Lorentz value at 25 cm^{-1}. The continuum is then simply defined by any observed absorption not attributable to the local absorption.
Application of this definition of the H_2O continuum to KOPRA calculations requires the subtraction of the so-called *basement*, which is the Lorentz value at 25 cm^{-1}, from the absorption coefficient of each individual H_2O line out to $\pm$ 25 cm^{-1} apart form the line center. Furthermore, calculation of absorption coefficient of H_2O lines is restricted to the region inside of $\pm$ 25 cm^{-1} apart form the line center. Therefore, a switch is introduced into the subroutine **add_lines_chilm@addlin_m** to perform this restriction and subtraction for H_2O lines in the case the H_2O continuum is included.
The original CKD model was developed by fitting the parameters in analytical χ-factors to reach agreement between the calculated continuum and laboratory measurements. For pragmatic reasons these χ-factors were used to calculate continuum coefficients $C^0_s(\nu,T)$ for the self-broadening component and $C^0_f(\nu)$ for the foreign-broadening component at reference pressure p_{ref} such that the continuum absorption cross section in units of [cm^2/molec] is given by:

$$\begin{aligned}\sigma_c(\nu,T,vmr_{H_2O}) &= \nu \tanh\left(\frac{hc\nu}{2kT}\right)\left(\frac{p}{p_{ref}}\right)\left(\frac{T_{ref}}{T}\right) \\ &\quad \times [vmr_{H_2O}\ C^0_s(\nu,T) + (1 - vmr_{H_2O})\ C^0_f(\nu)]\end{aligned} \tag{9}$$

Values of C_s^0 are tabulated for temperatures of 296 K and 260 K and C_f^0 for a temperature of 296 K in the wavenumber range from -20 cm^{-1} up to 20000 cm^{-1} in steps of 10 cm^{-1}. For calculation of the water vapor continuum at arbitrary wavenumbers these coefficients are interpolated linearly in wavenumbers. In the case of C_s^0 an exponential interpolation to arbitrary temperatures is performed.
The present CKDv2.2 model includes modifications of the original CKDv0 model to maintain better agreement with laboratory ansd atmospheric continuum measurements. These modifications are performed by correction factors to the continuum coefficients to reduce deviations with measurements in specific spectral regions.
Derivatives of the water vapor continuum with respect to temperatures are implemented by calculating the differential quotient for a temperature increment of dT = 0.1 K.

5 CO_2 Continuum

The CO_2 continuum absorption arises from the superposition of the extreme wings of many distant lines. To account for this CO_2 continuum in accurate line-by-line calculations it would be required to include CO_2 lines over a very large spectral region. This method would increase the required computation time dramatically. Therefore, CO_2 continuum coefficients are precalculated for different temperatures as a function of wavenumber.
The generation of CO_2 continuum coefficients is performed by a stand-alone program **create_co2coe.f90**. This program calculates the CO_2 absorption cross section at user-defined spectral positions and user-defined temperatures by superimposing all spectral lines out of a spectral region which also can be specified by the user. The spectral line shape used for the calculation of the continuum coefficients is the Lorentzian line shape modified by so-called χ-factors which account for the non-Lorentzian behavior of CO_2 lines. We used the asymmetric χ-factors of Menoux et al. (1987, 1991). As the temperature dependence of the asymmetric χ-factors for the O_2 broadening has not been investigated and, on the other hand, the temperature dependence of of the symmetric χ-factors indicate significant differences for N_2- and O_2 broadening we applied for O_2 broadening at low temperatures symmetric χ-factors.[4]These χ-factors are interpolated and extrapolated linearly in temperature. In cases where extrapolation leads to negative χ-factors they are set to zero.
The continuum cross-section includes all contributions of linewings out of $\pm$ 25 cm^{-1} apart from the line center plus the χ-modified Lorentz value at 25 cm^{-1} of each CO_2 line. Application of these continuum cross-section to KOPRA line-by-line calculations requires the subtraction of the χ-modified Lorentz value at 25 cm^{-1} from the absorption coefficient of each individual CO_2 line out to $\pm$ 25 cm^{-1}. Furthermore, calculation of absorption coefficient of CO_2 lines is restricted to the region inside of $\pm$ 25 cm^{-1} apart form the line center. Therefore, a switch is introduced into the subroutine **add_lines@addlin_m** to perform this restriction and subtraction for CO_2 lines in the case the CO_2 continuum is included.
The CO_2 continuum coefficients to be used by KOPRA are stored in the module **co2coe_m**. They were calculated for 6 different temperatures covering the range from 150 K to 350 K and for wavenumbers from 0 to 10000 cm^{-1} with 5 cm^{-1} spacing. The continuum coefficients can be assumed to be linear in pressure because the Lorentz line shape is only applied to regions more than 25 cm^{-1} apart from the line center where the air broadening half width is negligible to the distance from the line center. Therefore, the continuum coefficients stored in **co2coe_m** have units of $[\frac{cm^2}{molec\ hPa}]$.

In the subroutine **calc_co2cont@gascon_m** these continuum coefficients are interpolated linearly to the actual wavenumber and interpolated by the Steffen Interpolation scheme (subroutine **intp_steffen@varsub_m**) to the actual temperature. In order to obtain the continuum absorption cross-sections in units of $[\frac{cm^2}{molec}]$ the interpolated coefficients are multiplied by the actual pressure.
Derivatives of the CO_2 continuum with respect to temperatures are implemented by calculating the differential quotient for a temperature increment of dT = 0.1 K.

Bibliography

[1] F. Thibault, V. Menoux, R. Le Doucen, L. Rosenmann, J.-M. Hartmann, and C. Boulet, "Infrared collision-induced absorption by O_2 near 6.4 μm for atmospheric applications: measurements and empirical modeling," *Appl. Opt.* **36**(3), pp. 563–567, 1997.

[2] W. J. Lafferty, A. M. Solodov, A. Weber, W. B. Olson, and J.-M. Hartmann, "Infrared collision-induced absorption by N_2 near 4.3 μm for atmospheric applications: measurements and empirical modeling," *Appl. Opt.* **35**(30), pp. 5911–5917, 1996.

[3] S. A. Clough, "The water vapor continuum and its role in remote sensing," in *Optical Remote Sensing of the Atmosphere*, vol. 2 of *OSA Tech. Dig. Ser.*, pp. 76–78, Optical Society of America, Washington, D.C., 1995.

[4] C. Cousin, R. Le Doucen, C. Boulet, and A. Henry, "Temperature dependence of the absorption in the region beyond the 4.3-μm band head of CO_2. 2: N_2 and O_2 broadening," *Appl. Opt.* **24**(22), pp. 3899–3907, 1985.

The broadband continuum implementation

M. Höpfner and G. Echle

Abstract: Two possibilities for the simulation of a broadband continuum are described: the direct input of cross-sections and number density profiles or the calculation of cross-sections by a KORA-internal Mie model with aerosol size distribution parameters and refraction indices as input. Analytical derivatives with respect to various parameters are supported.

1 Forward calculation

Two different possibilities are implemented in KOPRA to consider the effects of a broadband continuum:

- Direct use of profiles of wavenumber dependent absorption and extinction cross sections and a profile of particle number density.
- Use of profiles of wavenumber dependent refraction indices (real and imaginary parts), of a profile of particle number density and profiles for parameters describing the particle distribution in a KOPRA-internal Mie-model to calculate the absorption and extinction cross sections.

1.1 Cross-section input

Input necessary for this mode of operation:

- Altitude profiles of the aerosol number density [particles/cm^3]
- Altitude profiles of absorption and extincition cross sections [cm^2/particle]. These are wavenumber depedent and can be given on an arbitrary wavenumber grid. Directly after reading, the cross sections are interpolated for each microwindow to a 5 cm^{-1} grid. (It is asumed here that 5 cm^{-1} are sufficient to describe wavenumber dependence of aerosols.)

In the next step, during ray-tracing, particle columns [particles/cm^2] for each atmospheric path are determined from the number density profiles (assuming linear interpolation between two altitude levels.) Further, Curtis-Godson values for each path are calculated for the wavenumber dependent absorption and extinction cross sections (using the number density for weighting).
During radiative transfer the cross sections for each path are multiplied with the particle columns to get the absorption and extinction optical depths. During this procedure it is necessary to interpolate the cross sections to the non-equidistant fine grid on which the radiative transfer is performed. This interpolation is performed linearly.

1.2 Mie-model input

Input necessary for this mode of operation:

- Altitude profiles of the aerosol number density N_{tot}[particles/cm^3].
- Number of modes for the particle distribution *nmode* (1- or 2- modal size distributions are supported).
- Altitude profiles of refraction indices (real and imaginary part). These are wavenumber depedent and can be given on an arbitrary wavenumber grid. Directly after reading, the refraction indices are interpolated for each microwindow to a 5 cm^{-1} grid. (It is asumed here that 5 cm^{-1} are sufficient to describe the detailed wavenumber dependence of aerosols.)
- Altitude profiles for mode radius r_i [μm], mode width s_i, and ratio of particles in mode 1 to the total number of particles v_N. (Number density and particle distribution parameters are given on the same altitude grid.)

It is assumed that the aerosols are spherical and homogeneous. The formula for the 1 or 2-modal log-normal size distribution density which is supported by KOPRA reads:

$$n(r) = \sum_{i=1}^{nmode} \frac{N_i}{\sqrt{2\pi} r s_i} \exp - \frac{(\ln \frac{r}{r_i})^2}{2 s_i^2}, \tag{1}$$

where r is the aerosol radius, $N_1 = N_{tot}$ for $nmode = 1$, and $N_1 = v_N N_{tot}, N_2 = (1 - v_N) N_{tot}$ for $nmode = 2$.
The integral over the size distribution density for all radii is the total aerosol number density

$$N_{tot} = \int_0^\infty n(r) dr. \tag{2}$$

During ray-tracing, particle columns [particles/cm^2] for each path are determined from the number density profiles (assuming linear interpolation between two altitude levels.) Further, Curtis-Godson values for each path are calculated for the particle distribution parameters and the wavenumber dependent refraction indices (using the number density for weighting) .
An essential difference to the previous section is that there the cross-sections are an input while in this section they have to be calculated from the Curtis-Godson path values of the particle distribution parameters and the refraction indices. This is performed by a Mie model which is called in the module where the absorption cross sections for the other species are also determined (Mie_call@abco_m).
Input for the Mie-model are the particle size distribution parameters and refraction indices. The model is described in detail in [1] and [2] . However, it was not possible to use this program unchanged since the integration over the particle distribution was rather slow. Therefore, the integration was optimized for the log-normal size distributions which are supported by KOPRA. During the Mie-calulations integrals of the form $\int_0^\infty f(r)n(r)dr$ are determined. By substituting $z^2 = (\ln r/r_i)^2/2s_i^2$ they are transformed into $\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(r(z)) \exp -z^2 dz$. This integrals can very efficiently be solved by using the Gauss-Hermite quadrature formula [3].
Output of the Mie-model are the absorption and extinction cross sections on the same wavenumber grid as the refraction indices.
As in the previous section during radiative transfer these cross sections for each path are multiplied with the particle columns and linearly interpolated to the wavenumber fine-grid to get the absorption and extinction optical depths.

2 Derivative calculation

KOPRA calculates analytical derivatives with respect to various parameters determinining the continuum contribution in the spectra.

2.1 Cross-section input

In this case the derivatives of the spectrum with respect to the aerosol number density is determined.
During the determination of path column values of aerosol particles in the raytracing module the derivatives of these column amounts with respect to the altitude retrieval-parameters of the number density are calculated. Afterwards, during radiative transfer first the derivatives of the spectrum with respect to the aerosol path column values are determined. Then, the post-derivation is performed by multiplication with the previously calculated derivatives of the column with respect to the altitude parameters of the number density.

2.2 Mie-model input

In addition to the derivatives with respect to number density which was already described in the previous section, derivative spectra with respect to the aerosol size distribution parameters (r_i, s_i, v_N) are produced.
As for number density, during the Curtis-Godson calculation the derivatives of the path values for r, s and v_N with respect to their altitude retrieval-parameterization are calculated. When the Mie-model is called with the Curtis-Godson layer values as input, parallel to the determination of the absorption and extinciton cross sections their derivatives with respect to the size distribution parameters are computed. (By analytic derivative of the log-normal size distribution functions with respect to r_i, s_i and v_N.)

During radiative transfer the following 3 steps are performed to get the final derivative spectra with respect to the size distribution altitude parametrization:

- Derivative of the radiance with respect to the absorption and extinction cross sections
- Multiplication by the derivative of these cross sections with respect to the Curtis-Godson layer values.
- Multiplication by the derivative of the Curtis-Godson layer values with respect to the altitude retrieval-parameter values

Bibliography

[1] H. Quenzel and H. Müller, Optical properties of single Mie particles: diagrams of intensity- extinction- scattering- and absorption efficiencies, Universität München, Meteorologisches Institut, Wissenschaftliche Mitteilung Nr. 34, 1978.

[2] G. Metzig, Optima: Berechnung der optischen Eigenschaften einzlener Mie-Partikeln mit homogener oder schalenförmiger Struktur, Kernforschungszentrum Karlsruhe (KfK), KfK Report 3678, 1984.

[3] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran, 2nd edition, Cambridge University Press, p. 147 1992.

Non-LTE and radiative transfer

B. Funke and M. Höpfner

Abstract: In this part, we describe the radiative transfer modeling for LTE and non-LTE situations. The necessary extensions for consideration of vibrational and rotational non-LTE are discussed in detail. Finally, some aspects concerning the technical implementation are described.

1 Introduction

The calculation of radiative transfer within KOPRA is performed exclusively in the module radtra_m. Since also the main part of the simulation of non-LTE effects is done in this part of the code in the following we describe first the general LTE-radiative transfer and then the extensions for non-LTE. At the end we will make some remarks about the technical implementation, especially about the interpolation on the wavenumber axis.

2 The radiative transfer in LTE

In contrast to the analytic formulation of radiative transfer in Part II: 'Analytical expressions for radiative transfer modeling of instrumental effects in KOPRA' the discrete formula reads

$$S = B(T_{back}) \prod_{l=1}^{L} \tau_l + \sum_{l=1}^{L} \tilde{J}_l (1 - \tau_l) \prod_{l'=l+1}^{L} \tau_{l'} \tag{1}$$

where l is the index for the layers with layer 1 the farthest and layer L the nearest with respect to the observer. Mind that in this formulation only mean (Curtis-Godson) values of each layer along the ray-path are used. The index on wavenumber is omitted for clarity. T_{back} is the temperature of the background. $\tilde{J}_l$ is the source function for layer l and τ_l the single layer transmission:

$$\tau_l = \exp\left[-\delta_l\right] \tag{2}$$

δ_l is the optical depth for layer l:

$$\delta_l = \delta_{e,Aerosol,l} + \sum_g \sigma_{a,gl} m_{gl} \tag{3}$$

where $\delta_{e,Aerosol,l}$ is the optical depth of aerosol extinction, $\sigma_{a,gl}$ the absorption coefficient, and m_{gl} the partial column density of gas g in layer l.
For the most simple case

$$\tilde{J}_l = B(T_{kin,l}) = \frac{2hc^2\nu^3}{\exp\left(\frac{hc\nu}{k_B T_{kin,l}}\right) - 1}, \tag{4}$$

i.e. the Planck function for the Curtis-Godson temperature of air for layer l (with c: velocity of light, h: Planck's constant, k_B: Boltzmann's constant). The rigorous formulation (to be used for largely different Curtis-Godson temperatures for each gas, for different aerosol absorption and emission coefficients or in the case of non-LTE (see below)) reads

$$\tilde{J}_l = \frac{\hat{J}_l}{\delta_l} = \frac{B(T_{kin,l})\delta_{a,Aerosol,l} + \sum_g \left[J_{gl}\sigma_{a,gl}\right] m_{gl}}{\delta_l}, \tag{5}$$

where $\delta_{a,Aerosol,l}$ is the optical depth of aerosol absorption, $J_{gl} = B(T_{kin,gl})$ the Planck function for the Curtis-Godson temperature of gas g for layer l. Note that in this formulation the aerosol emission is determined by the absorption coefficient of aerosol. This means that only the emission of light by aerosols and not scattering of light into the light path is described in the source term. The scattering of photons out of the light path is considered in the optical depth δ_l by use of the aerosol extinction. For $\delta_{a,Aerosol} = \delta_{e,Aerosol}$ no scattering is described. While Eq. 1 at first glance seems to erroneously neglect the distinction between absorption and extinction the weighting by δ in Eq. 5 compensates for this.

3 Non-LTE

3.1 Introduction

Under conditions of local thermodynamic equilibrium (LTE) the relative populations of the lower and upper state of a ro-vibrational transition are related to the

local kinetic temperature by Boltzmann's law. This implies that the source function of a given vibrational-rotational emission is equal to the Planck function at the local kinetic temperature. However, the assumption of LTE for the vibrational degree of freedom is not valid for many infrared emitting gases at and above mesospheric altitudes (CO_2, O_3, CO, etc.), and, in some cases (NO, NO_2, O_3, etc.) even in the stratosphere. LTE breaks down if non-thermal excitation mechanisms such as photochemical processes, solar pumping, and absorption of upwelling radiation cannot be compensated by collisional relaxation. The resulting non-LTE emissions can affect limb radiances even at tangent heights much lower than the breakdown of LTE occurs. In order to consider the effect of non-LTE within radiative transfer calculations the non-LTE state distribution has to be known. Dedicated non-LTE models for numerous infrared emitting gases calculate the non-LTE vibrational state distribution by solving the statistical equilibrium equation under consideration of collisional, chemical, and radiative processes. In each layer l, the vibrational state distribution is then expressed by the vibrational temperature for each vibrational level m of specie g, or

$$T_{vib,gml} = -hE_{vib,gm}\left[k_B \ln\left(\frac{n_{gml}}{n_{g0l}}\right)\right]^{-1}, \tag{6}$$

where n_{gml} and n_{g0l} are the fractional populations of the state m and the ground state, respectively. Alternatively, the non-LTE state distribution can be expressed by the ratio r_{gml} of the fractional populations of the vibrational level m in non-LTE and in LTE:

$$r_{gml} = \frac{n_{gml}^{NLTE}}{n_{gml}^{LTE}}. \tag{7}$$

The latter representation of non-LTE populations is used internally in the code. Either $T_{vib,gml}$ or r_{gml} profiles are necessary input data for KOPRA non-LTE calculations. The treatment of radiative transfer under non-LTE conditions requires some modifications with respect to LTE regarding a) the non-LTE partition sum, b) the non-LTE source function, and c) the non-LTE absorption coefficient.

3.2 The non-LTE correction to the vibrational partition sum

If vibrational temperatures are provided by the user the conversion of $T_{vib,gml}$ to r_{gml} requires the calculation of the non-LTE vibrational partition sum. The LTE vibrational partition sum, can be expressed according to Boltzmann's law:

$$Q_{gl}^{LTE} = \sum_{\eta} g_{g\eta} \exp\left(-\frac{E_{g\eta}}{k_B T_{kin,l}}\right), \tag{8}$$

where the sum is over all accessible vibrational-rotational states η with the state energy $E_{g\eta}$ and the degeneracy factor $g_{g\eta}$. If vibrational LTE breaks down, the vibrational states are not distributed according to Boltzmann's law which raises the necessity for non-LTE corrections to the partition sum. Under assumption of rotational LTE (which is valid for most atmospheric species below approx. 120 km) and a negligible vibration-rotation interaction, the non-LTE partition sum can be expressed according to *Edwards et al.* [1] by $Q_{gl}^{NLTE} = f_{Q,gl} Q_{gl}^{LTE}$ with the correction factor

$$f_{Q,gl} = \frac{\sum_m g_{gm} \exp\left(-\frac{E_{vib,gm}}{k_B T_{kin,l}}\right)}{\sum_m g_{gm} \exp\left(-\frac{E_{vib,gm}}{k_B T_{vib,gml}}\right)}. \tag{9}$$

The summation is performed within the code over all vibrational states m for which vibrational temperature profiles are provided. The relation between $T_{vib,gml}$ and r_{gml} is then given by

$$r_{gml} = f_{Q,gl} \frac{\exp\left(-E_{vib,gm}/k_B T_{vib,gml}\right)}{\exp\left(-E_{vib,gm}/k_B T_{kin,l}\right)}. \tag{10}$$

3.3 The non-LTE absorption coefficient

The absorption coefficient $\sigma_{a,gl}$ for a given species g is defined as

$$\sigma_{a,gl}(\nu) = \sum_n \frac{h\nu_n}{c} B_{\eta_1\eta_2} n_{\eta_1 l} \left(1 - \frac{g_{\eta_1} n_{\eta_2}}{g_{\eta_2} n_{\eta_1}}\right) \Phi_{nl}(\nu), \tag{11}$$

where the sum is over all transitions n between the ro-vibrational states η_1 and η_2. $B_{\eta_1\eta_2}$ is the Einstein coefficient for absorption and $\Phi_{nl}(\nu)$ is the profile function. Under conditions of LTE, the ratio $(g_{\eta_1} n_{\eta_2})/(g_{\eta_2} n_{\eta_1})$ is given by the Boltzmann factor Γ at the local kinetic temperature $T_{kin,l}$,

$$\Gamma = \exp\left(-\frac{hc\nu}{k_B T_{kin,l}}\right). \tag{12}$$

Under vibrational non-LTE, the absorption coefficient can then be related to the LTE band absorption coefficients $\sigma^{LTE}_{a,gkl}$ of all vibrational bands k between the levels m_1 and m_2 by

$$\sigma^{NLTE}_{a,gl}(\nu) = \sum_k \alpha_{gkl} \sigma^{LTE}_{a,gkl} \tag{13}$$

where α_{gkl} is defined as

$$\alpha_{gkl} = \frac{r_{gm_1 l} - r_{gm_2 l}\Gamma}{1 - \Gamma}. \tag{14}$$

3.4 The non-LTE source function

A general form of the source function for a single ro-vibrational transition can be written

$$J_{nl} = 2hc^2\nu^3 \left(\frac{g_{\eta_1} n_{\eta_2}}{g_{\eta_2} n_{\eta_1}} \exp\left(\frac{hc\nu}{k_B T_{kin,l}}\right) - 1\right)^{-1}, \tag{15}$$

which is equal to the Planck function for LTE conditions. Under vibrational non-LTE, the source function for the vibrational band k can be expressed by $r_{gm_1 l}$ and $r_{gm_2 l}$:

$$J^{NLTE}_{gkl} = 2hc^2\nu^3 \left(\frac{r_{gm_1 l}}{r_{gm_2 l}} \exp\left(\frac{hc\nu}{k_B T_{kin,l}}\right) - 1\right)^{-1}. \tag{16}$$

The term $J_{gl}\sigma_{a,gl}$ in Eq. 5 then has to be substituted by

$$J_{gl}\sigma_{a,gl} = \sum_k J^{NLTE}_{gkl} \sigma^{NLTE}_{a,gkl}. \tag{17}$$

3.5 Rotational non-LTE

Rotational non-LTE occurs at high altitudes, where collisional thermalization of the rotational state distribution is slower than spontaneous emission. The inclusion of rotational non-LTE in atmospheric radiance calculation becomes necessary for species with high concentrations in the upper atmosphere (above 120 km), such as NO [2]. The radiative transfer equation under conditions of rotational non-LTE can be derived from the vibrational non-LTE case by substitution of the vibrational levels m by ro-vibrational states m, j and vibrational bands k by ro-vibrational lines n. Thus, under conditions of rotational non-LTE, the expressions for absorption coefficient and source function are given by

$$\sigma_{a,gl}^{NLTE}(\nu) = \sum_n \alpha_{gnl}\sigma_{a,gnl}^{LTE}\,,\ \alpha_{gnl} = \frac{r_{gm_1j_1l} - r_{gm_2j_2l}\Gamma}{1-\Gamma}, \tag{18}$$

$$J_{nl}^{NLTE} = 2hc^2\nu^3 \left(\frac{r_{gm_1j_1l}}{r_{gm_2j_2l}} \exp\left(\frac{hc\nu}{k_B T_{kin,l}}\right) - 1\right)^{-1}, \tag{19}$$

$$J_{gl}\sigma_{a,gl} = \sum_n J_{gnl}^{NLTE}\sigma_{a,gnl}^{NLTE}, \tag{20}$$

where r_{gmjl} is defined according to Eq. 7 as the the ratio of the non-LTE populations and LTE populations of the ro-vibrational state m, j, respectively.

4 Wavenumber interpolation

The absorption coefficients are calculated in the modules abco_m and addlin_m for each pair of Curtis-Godson temperatures and pressures (i.e. for each layer of the forward model) on a layer-specific wavenumber grid. This means, however, that during the radiative transfer where the atmospheric layers are linked an interpolation in wavenumber has to be performed.
At the beginning of the radiative transfer the common wavenumber grid for all layers of the line-of-sight is determined. Then an interpolation to this grid is performed only once for each absorption coefficient in each layer and the subsequent calculations for radiative transfer and derivative calculation are performed exclusively on the common grid. By this scheme the frequent interpolation of two different grids to their common grid (and the determination of this grid) is avoided (avoid 'if's to search the common grid inside the loop over wavenumbers).

Bibliography

[1] D. P. Edwards, M. López-Puertas, and R. R. Gamache, "The non–LTE correction to the vibrational component of the internal partition sum for atmospheric calculation," *J. Quant. Spectrosc. Radiat. Transfer* **59**(3–5), pp. 423–436, 1998.

[2] B. Funke and M. López-Puertas, "Nonlocal thermodynamic equilibrium vibrational, rotational, and spin state distribution of NO(ν= 0,1,2) under quiescent atmospheric conditions," *J. Geophys. Res.* **105**(D4), pp. 4409–4426, 2000.

The Voigt profile function and the Planck function

M. Kuntz and M. Höpfner

Abstract: A new implementation of Humlicek's algorithm for approximating the Voigt profile function is applied in KOPRA and compared with several other implementations with respect to computational speed. On scalar computers this new implementation is considerably faster than other implementations by more than a factor 3.3 on the average. However, on the vector computer the acceleration between the new implementation and a fully vectorized implementation as described by Schreier [3] is only between 1.2 and 1.8, depending on the region of the x, y-space under consideration.
KOPRA is especially designed for the calculation of spectral microwindows. Therefore, the Planck function, the non-LTE source function and the non-LTE correction factor for absorption cross sections are calculated in such a way that a given relative accuracy over the microwindow range is reached.

1 Voigt profile

1.1 Introduction

The Voigt profile function is the convolution of a Gaussian and a Lorentzian function. It is expressed as

$$K(x,y) = \frac{y}{\pi}\int_{-\infty}^{+\infty}\frac{\exp(-t^2)}{(x-t)^2+y^2}dt, \tag{1}$$

where x is the distance from the line center in units of Doppler halfwidths and y is the ratio of the Doppler halfwidth to the Lorentzian halfwidth. The Voigt profile function is used in a wide range of contexts and there are—relevant to practical numerical algorithms—numerous ways it can be computed [1]. Many papers have been published describing routines for evaluating the Voigt profile function based upon various numerical expansions in different regions of the x,y space. Some of them—including Humlicek's algorithm [2]—take advantage of the fact that the Voigt profile function can be manipulated into an expression in terms of the complementary error function of complex argument, erfc(z), as

$$K(x,y) = \mathrm{Re}\left[\exp(z^2)\mathrm{erfc}(z)\right], \qquad z = y - ix. \tag{2}$$

This relation between the Voigt profile function $K(x,y)$ and erfc(z) allows for approximations of the error function erf(z) to be used also for K, since $\mathrm{erfc}(z) = 1-\mathrm{erf}(z)$. Taking advantage of this fact Humlicek's algorithm divides the x,y space into four regions, see Fig. 1. It then approximates the complex expression on the right hand side of Eq. (2), for each region, by appropriate rational polynomials. These polynomials are chosen to optimize the combination of accuracy and speed of computation. The Humlicek algorithm thus simultaneously calculates both the real and imaginary parts of Eq. (2). However, in optical spectroscopy only the real part is needed, especially if line coupling effects need not be taken into account. The purpose of this paper is to present a highly efficient algorithm for calculating the real part of the right hand side of Eq. (2).
In addition to its speed this new algorithm is still very accurate, since Humlicek's complex rational approximations are substituted by real ones. As test calculations revealed, the relative error compared with Humlicek's original implementation is less than $2\,10^{-6}$ throughout the x,y space.

1.2 Acceleration of Humlicek's algorithm

The first modification we propose refers to overall organization of the algorithm with respect to the nesting of DO-loops and IF-inquiries. Most computer programs that use the Voigt profile function apply it to a set of regularly spaced x values for a given y value rather than to selected x,y points. This is especially true if a complete atomic or rotational line is calculated. While the original Humlicek algorithm determines the region (and thus the related polynomials) for each x,y pair individually, we propose to take advantage of the regularity of the grid points in x. The regions then need only be determined at their end points, see Fig. 1. This results in an acceleration of Humlicek's algorithm without any loss of accuracy, only by reducing the number of IF-inquiries which are necessary to assign each x value to its appropriate region. The following sequence of operations, which defines a recursive algorithm, avoids repeated IF-inquiries and thus is well suited for evaluation of the Voigt profile function for a large set of x values. In order to apply this algorithm the x values have to be arranged in increasing order. For the sake of simplicity we shall further assume that x takes on only positive values. The extension to negative values can be easily performed.

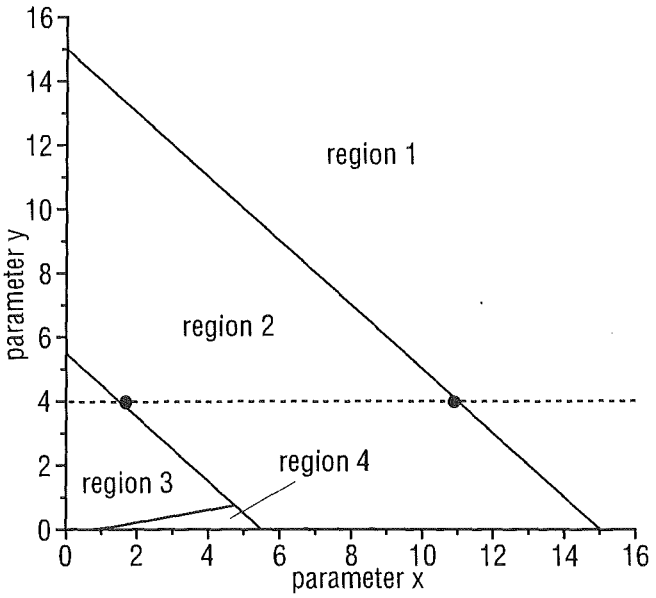


Figure 1: The four regions in the x,y space relevant for Humlicek's approximation of the Voigt profile function.

(a) If the left and right grid point of the current succession of x values belong to the same region (which implies, that all x values of the current succession also belong to this region), then store the indices of the extreme left and right grid points of this region so far.

(b) Else subdivide the current succession of x values into a left and right succession of approximately the same size and pop the right succession onto the stack. (In programming languages which do not support recursion the stack has to be provided by the user). Continue with the left succession at step 1.

(c) Push the next succession of x values from the stack and continue with step 1. If the stack is empty, finish the recursive algorithm.

As a consequence the number of IF-inquiries which are necessary to assign each x value to its appropriate region can be reduced from originally being equal to four times the number n of x values to less than $4\ln(n)/\ln(2)$. In addition the indices of the extreme left and right grid points of each region are known. This allows to substitute the former DO-loop containing conditional IF-branches (running over all x values) by four DO-loops (running over all x values lying within the same region) which do not contain IF-inquiries any longer. The new implementation can thus be better vectorized.
The second second modification we propose makes use of the fact that for most applications only the real part of Eq. (2) is needed. We thus substitute the complex rational polynomials Humlicek uses for approximation within each region by real ones. The latter can be calculated from the original approximations by separating them into a real and imaginary part, neglecting the imaginary one. In view of the intricacy of Humlicek's complex approximations all calculations were performed using a computer algebra system. A complete printout of the calculated real approximations for the Voigt profile function is given in the appendix. This leads to a considerable reduction of the number of floating point operations which have to be performed for each grid point in x individually. The reason is that those parts of the

calculations which concern only the fixed parameter y can be drawn forward and already be calculated before the DO-loops. The extent of the reduction of floating point operations within each region is indicated in Tab. 1.

Table 1: Achieved reduction of floating point operations in each of Humlicek's four regions. (In the case of Humlicek's complex rational approximation of the Voigt profile function all necessary complex floating operations have been expressed as real floating point operation equivalents.)

region	complex			real		
	+/−	*/÷	exp	+/−	*/÷	exp
1	7	16	0	3	4	0
2	17	24	0	7	8	0
3	37	38	0	9	10	0
4	56	62	3	29	30	2

1.3 Comparison of computing times

Most applications require the Voigt profile function for an array of x values at some selected y values, rather than for a limited set of x, y points. We thus follow a proposal of Schreier [3] and consider two sets of y values, $0 < y < 1$ and $1 < y < 10$, for comparison corresponding to Doppler and Lorentzian "dominance". For each of 50 y values in these intervals, we calculate the Voigt profile function for 1000 grid points in the interval $0 \leq x_i \leq x_{\max}$ with $x_{\max} = 10\,x_{\text{half}}$. For the halfwidth x_{half} of the Voigt profile function we use an empirical approximation $x_{\text{half}}(y) = \frac{1}{2}(y + \sqrt{y^2 + 4\ln 2})$ proposed by Whiting [4]. In agreement with Schreier no significant difference in computer time was found for smaller and larger values of y ($0 < y < 0,1$ or $10 < y < 100$) or for larger $x_{\text{xmax}} = 20\,x_{\text{half}}$. Furthermore, computing times varied linearly with the number of grid points in the x, y-plane.

The tests were performed on a 90 MHz Pentium PC under DOS, using WATCOM's nonoptimizing WATFOR87 compiler (16 bit) and WATCOM's FORTRAN 77^{32} optimizing compiler (32 bit), respectively. The SUN SPARC20 workstation ran under Solaris3.1 operating system with a SunSoft FORTRAN 77 4.0 compiler while the CRAYJ90 (CMOS) used UNICOS 8.0.4.2 with a CF77 compiler. Highest optimization and vectorization level has been applied where possible.

Times required for 1000 × 50 evaluations are listed in Tab. 2. Besides Humlicek's original [2] and our new implementation two additional implementations have been included in our comparison of computational speed. The first one is part of GENLIN [5], a computer program for modeling the atmospheric radiative transfer line-by-line; the second one has been suggested by Schreier [3], who has drawn much attention to the vectorizability of his implementation. By far the fastest program on all scalar computers was our new implementation of Humlicek's algorithm, which on the average led to an acceleration factor of more than 3.3. On the SPARC workstation the acceleration even exceeded a factor 5.5. The other implementations were all nearly equally slow with slight advantages for the one or other depending on the computer and region under examination. However, on the vector computer Schreier's implementation was nearly as fast as our new implementation, with slight advantages for the latter in the region $1 < y < 10$ (factor 1.8). Both implementations were superior to the other implementations with respect to computational speed by at least a factor 10 due to their higher vectorizability.

Table 2: Computing times in seconds for various implementations of Humlicek's algorithm (1000×50 points and $0 < x < 10\,x_{\mathrm{half}}$).

Time (sec)	$0 < y < 1$			
	Humlicek	GENLIN	Schreier	new
PC 16 bit	2,447	2,070	1,771	1,224
PC 32 bit	2,410	1,538	1,494	0,517
SPARC20	0,359	0,204	0,216	0,058
CRAYJ90	0,220	0,387	0,046	0,025

Time (sec)	$1 < y < 10$			
	Humlicek	GENLIN	Schreier	new
PC 16bit	1,202	1,426	1,078	0,682
PC 32bit	0,647	0,547	0,548	0,203
SPARC20	0,108	0,095	0,089	0,015
CRAYJ90	0,138	0,184	0,016	0,013

In summary, both, the implementation of Schreier as well as our new implementation, represent good alternatives for a relatively fast calculation of the Voigt profile function on vector computers. On scalar computers our new implementation seems to be superior to the other implementations by more than a factor 3.3 on the average and a factor 5.5 for the SPARC20 workstation.

1.4 Conclusion on the Voigt profile implementation

We have proposed a new implementation of Humlicek's algorithm for approximating the Voigt profile function and have compared it with several other implementations with respect to accuracy and computational speed. While the accuracy of our new implementation can be regarded as very satisfactory, its computational speed is considerably higher than for the other implementations on scalar computers. On the vector computer our new implementation is only slightly faster than the implementation of Schreier (the acceleration factor varies between 1.2 and 1.8 depending on the region of the x,y space under examination) but it still exceeds the implementations of Humlicek and in GENLIN by at least a factor 10.

2 Planck function, non-LTE source function and non-LTE correction for absorption cross-sections

2.1 Introduction

For radiative transfer the Planck function B and in case of non-LTE the source function J^{NLTE} and the correction factor for absorption cross-sections α must be determined for each spectral grid point ν_i under consideration:

$$B(T_{kin,li}) = 2hc^2\nu_i^3 \left(\exp\left(\frac{hc\nu_i}{k_B T_{kin,l}}\right) - 1\right)^{-1}, \tag{3}$$

$$J_{gkli}^{NLTE} = 2hc^2\nu_i^3 \left(\frac{r_{gm_1l}}{r_{gm_2l}} \exp\left(-\frac{hc\nu_i}{k_B T_{kin,l}}\right) - 1\right)^{-1}, \tag{4}$$

$$\alpha_{gkli} = \frac{r_{gm_1l} - r_{gm_2l} \exp\left(-\frac{hc\nu_i}{k_B T_{kin,l}}\right)}{1 - \exp\left(-\frac{hc\nu_i}{k_B T_{kin,l}}\right)}. \tag{5}$$

(For a detailed description of all variables see the illustration of radiative transfer in Part X: 'Non-LTE and radiative transfer')
For the determination of analytical derivatives the derivatives of these three functions with respect to the kinetic temperature T_{kin} and the non-LTE/LTE population ratios r are calculated.
All these functions are, compared with the spectral line structure of the absorption cross sections, smoothly varying with wavenumber. Therefore, it is not necessary to recalculate them for each spectral grid point ν_i which would be very time consuming due to the exponentials.

2.2 Optimized implementation in KOPRA

The implementation in KOPRA is optimized particularly with regard to the calculation of spectral microwindows. Four steps with increasing complexity of wavenumber interpolation and microwindow subdivision are tested:

(a) *Constant value:* the function is calculated exactly at the end points and one point in the middle of the microwindow. If the maximum difference is less than a certain relative threshold (10^{-5}) the function value of the middle point is used for all grid points.

(b) *Linear interpolation:* If the previous criterion is not fulfilled it is checked if the mean of the function values at the microwindow boarders differ less than the relative threshold from the function value in the middle of the microwindow. Then, linear interpolation to all other grid points is performed.

(c) *Quadratic interpolation:* In case 1. and 2. are not valid new exact function values in the middle of the two previous intervals are determined and with these it is tested if a quadratic interpolation (using the end points and the middle of the microwindow) is sufficient.

(d) *Quadratic interpolation in sub-intervals:* If even 3. is not sufficient the microwindow is iteratively subdivided into smaller intervals until the quadratic interpolation reaches the relative error limit.

This scheme is used in the following KOPRA routines:

- `planckn@radtra_m`: Planck function and derivative with respect to kinetic temperature
- `sourcen@radtra_m`: non-LTE source function and derivative with respect to kinetic temperature
- `alphan@radtra_m`: non-LTE correction factor for cross-sections and derivative with respect to kinetic temperature
- `dsourcedTvibn@radtra_m`: derivative of the non-LTE source function with respect to the non-LTE/LTE population ratios
- `dalphadTvibn@radtra_m`: derivative of the non-LTE source function with respect to the non-LTE/LTE population ratios

Appendix A

Printout of used polynominals

Printout of the rational polynominals used for the approximation of the Voigt profile function. **region 1:** $|x| + y > 15$

$$
\begin{aligned}
a_1 &= 0.2820948y + 0.5641896y^3 \\
b_1 &= 0.5641896y \\
a_2 &= 0.25 + y^2 + y^4 \\
b_2 &= -1 + 2y^2
\end{aligned}
$$

$$
K(x,y) = \frac{a_1 + b_1 x^2}{a_2 + b_2 x^2 + x^4}
$$

region 2: $5.5 < |x| + y < 15$

$$
\begin{aligned}
a_3 &= 1.05786y + 4.65456y^3 + 3.10304y^5 + 0.56419y^7 \\
b_3 &= 2.962y + 0.56419y^3 + 1.69257y^5 \\
c_3 &= 1.69257y - 2.53885y^3 \\
d_3 &= 0.56419y \\
a_4 &= 0.5625 + 4.5y^2 + 10.5y^4 + 6y^6 + y^8 \\
b_4 &= -4.5 + 9y^2 + 6y^4 + 4y^6 \\
c_4 &= 10.5 - 6y^2 + 6y^4 \\
d_4 &= -6 + 4y^2
\end{aligned}
$$

$$
K(x,y) = \frac{a_3 + b_3 x^2 + c_3 x^4 + d_3 x^6}{a_4 + b_4 x^2 + c_4 x^4 + d_4 x^6 + x^8}
$$

region 3: $|x| + y < 5.5$ and $y > 0.195|x| - 0.176$

$$
\begin{aligned}
a_5 &= 272.102 + 973.778y + 1629.76y^2 + 1678.33y^3 + 1174.8y^4 \\
&\quad +581.746y^5 + 204.501y^6 + 49.5213y^7 + 7.55895y^8 + 0.564224y^9 \\
b_5 &= -60.5644 - 2.34403y + 220.843y^2 + 336.364y^3 + 247.198y^4 \\
&\quad +100.705y^5 + 22.6778y^6 + 2.25689y^7 \\
c_5 &= 4.58029 + 18.546y + 42.5683y^2 + 52.8454y^3 + 22.6798y^4 \\
&\quad +3.38534y^5 \\
d_5 &= -0.128922 + 1.66203y + 7.56186y^2 + 2.25689y^3 \\
e_5 &= 0.000971457 + 0.564224y \\
a_6 &= 272.102 + 1280.83y + 2802.87y^2 + 3764.97y^3 + 3447.63y^4 \\
&\quad +2256.98y^5 + 1074.41y^6 + 369.199y^7 + 88.2674y^8 + 13.3988y^9 \\
&\quad +y^{10} \\
b_6 &= 211.678 + 902.306y + 1758.34y^2 + 2037.31y^3 + 1549.68y^4 \\
&\quad +793.427y^5 + 266.299y^6 + 53.5952y^7 + 5y^8 \\
c_6 &= 78.866 + 308.186y + 497.302y^2 + 479.258y^3 + 269.292y^4 \\
&\quad +80.3928y^5 + 10y^6 \\
d_6 &= 22.0353 + 55.0293y + 92.7568y^2 + 53.5952y^3 + 10y^4 \\
e_6 &= 1.49645 + 13.3988y + 5y^2
\end{aligned}
$$

$$
K(x,y) = \frac{a_5 + b_5x^2 + c_5x^4 + d_5x^6 + e_5x^8}{a_6 + b_6x^2 + c_6x^4 + d_6x^6 + e_6x^8 + x^{10}}
$$

region 4: $|x| + y < 5.5$ and $y < 0.195|x| - 0.176$

$$
\begin{aligned}
a_7 &= 1.16028e9y - 9.86604e8y^3 + 4.56662e8y^5 - 1.53575e8y^7 + 4.08168e7y^9 \\
&\quad -9.69463e6y^{11} + 1.6841e6y^{13} - 320772y^{15} + 40649.2y^{17} - 5860.68y^{19} \\
&\quad +571.687y^{21} - 72.9359y^{23} + 2.35944y^{25} - 0.56419y^{27} \\
b_7 &= -5.60505e8y - 9.85386e8y^3 + 8.06985e8y^5 - 2.91876e8y^7 + 8.64829e7y^9 \\
&\quad -7.72359e6y^{11} + 3.59915e6y^{13} - 234417y^{15} + 45251.3y^{17} - 2269.19y^{19} \\
&\quad -234.143y^{21} + 23.0312y^{23} - 7.33447y^{25} \\
c_7 &= -6.51523e8y + 2.47157e8y^3 + 2.94262e8y^5 - 2.04467e8y^7 + 2.29302e7y^9 \\
&\quad -2.3818e7y^{11} + 576054y^{13} + 98079.1y^{15} - 25338.3y^{17} + 1097.77y^{19} \\
&\quad +97.6203y^{21} - 44.0068y^{23} \\
d_7 &= -2.63894e8y + 2.70167e8y^3 - 9.96224e7y^5 - 4.15013e7y^7 + 3.83112e7y^9 \\
&\quad +2.2404e6y^{11} - 303569y^{13} - 66431.2y^{15} + 8381.97y^{17} + 228.563y^{19} \\
&\quad -161.358y^{21} \\
e_7 &= -6.31771e7y + 1.40677e8y^3 + 5.56965e6y^5 + 2.46201e7y^7 + 468142y^9 \\
&\quad -1.003e6y^{11} - 66212.1y^{13} + 23507.6y^{15} + 296.38y^{17} - 403.396y^{19} \\
f_7 &= -1.69846e7y + 4.07382e6y^3 - 3.32896e7y^5 - 1.93114e6y^7 - 934717y^9 \\
&\quad +8820.94y^{11} + 37544.8y^{13} + 125.591y^{15} - 726.113y^{17} \\
g_7 &= -1.23165e6y + 7.52883e6y^3 - 900010y^5 - 186682y^7 + 79902.5y^9 \\
&\quad +37371.9y^{11} - 260.198y^{13} - 968.15y^{15} \\
h_7 &= -610622y + 86407.6y^3 + 153468y^5 + 72520.9y^7 + 23137.1y^9 \\
&\quad -571.645y^{11} - 968.15y^{13} \\
o_7 &= -23586.5y + 49883.8y^3 + 26538.5y^5 + 8073.15y^7 - 575.164y^9 \\
&\quad -726.113y^{11} \\
p_7 &= -8009.1y + 2198.86y^3 + 953.655y^5 - 352.467y^7 - 403.396y^9 \\
q_7 &= -622.056y - 271.202y^3 - 134.792y^5 - 161.358y^7 \\
r_7 &= -77.0535y - 29.7896y^3 - 44.0068y^5 \\
s_7 &= -2.92264y - 7.33447y^3 \\
t_7 &= -0.56419y \\
a_8 &= 1.02827e9 - 1.5599e9y^2 + 1.17022e9y^4 - 5.79099e8y^6 + 2.11107e8y^8 \\
&\quad -6.11148e7y^{10} + 1.44647e7y^{12} - 2.85721e6y^{14} + 483737y^{16} - 70946.1y^{18} \\
&\quad +9504.65y^{20} - 955.194y^{22} + 126.532y^{24} - 3.68288y^{26} + y^{28} \\
b_8 &= 1.5599e9 - 2.28855e9y^2 + 1.66421e9y^4 - 7.53828e8y^6 + 2.89676e8y^8 \\
&\quad -7.01358e7y^{10} + 1.39465e7y^{12} - 2.84954e6y^{14} + 498334y^{16} - 55600y^{18} \\
&\quad +3058.26y^{20} + 533.254y^{22} - 40.5117y^{24} + 14y^{26} \\
c_8 &= 1.17022e9 - 1.66421e9y^2 + 1.06002e9y^4 - 6.60078e8y^6 + 6.33496e7y^8 \\
&\quad -4.60396e7y^{10} + 1.4841e7y^{12} - 1.06352e6y^{14} - 217801y^{16} + 48153.3y^{18} \\
&\quad -1500.17y^{20} - 198.876y^{22} + 91y^{24} \\
d_8 &= 5.79099e8 - 7.53828e8y^2 + 6.60078e8y^4 + 5.40367e7y^6 + 1.99846e8y^8 \\
&\quad -6.87656e6y^{10} - 6.89002e6y^{12} + 280428y^{14} + 161461y^{16} - 16493.7y^{18} \\
&\quad -567.164y^{20} + 364y^{22} \\
e_8 &= 2.11107e8 - 2.89676e8y^2 + 6.33496e7y^4 - 1.99846e8y^6 - 5.01017e7y^8 \\
&\quad -5.25722e6y^{10} + 1.9547e6y^{12} + 240373y^{14} - 55582y^{16} - 1012.79y^{18} \\
&\quad +1001y^{20}
\end{aligned}
$$

$$
\begin{aligned}
f_8 &= 6.11148e7 - 7.01358y^2 + 4.60396e7y^4 - 6.87656e6y^6 + 5.25722e6y^8 \\
&\quad +3.04316e6y^{10} + 123052y^{12} - 106663y^{14} - 1093.82y^{16} + 2002y^{18} \\
g_8 &= 1.44647e7 - 1.39465e7y^2 + 1.4841e7y^4 + 6.89002e6y^6 + 1.9547e6y^8 \\
&\quad -123052y^{10} - 131337y^{12} - 486.14y^{14} + 3003y^{16} \\
h_8 &= 2.85721e6 - 2.84954e6y^2 + 1.06352e6y^4 + 280428y^6 - 240373y^8 \\
&\quad -106663y^{10} + 486.14y^{12} + 3432y^{14} \\
o_8 &= 483737 - 498334y^2 - 217801y^4 - 161461y^6 - 55582y^8 \\
&\quad +1093.82y^{10} + 3003y^{12} \\
p_8 &= 70946.1 - 55600y^2 - 48153.3y^4 - 16493.7y^6 + 1012.79y^8 \\
&\quad +2002y^{10} \\
q_8 &= 9504.65 - 3058.26y^2 - 1500.17y^4 + 567.164y^6 + 1001y^8 \\
r_8 &= 955.194 + 533.254y^2 + 198.876y^4 + 364y^6 \\
s_8 &= 126.532 + 40.5117y^2 + 91y^4 \\
t_8 &= 3.68288 + 14y^2
\end{aligned}
$$

$$
\begin{aligned}
K(x,y) &= e^{y^2-x^2}\cos(2xy) - \\
&\quad \frac{a_7 + b_7x^2 + c_7x^4 + d_7x^6 + e_7x^8 + f_7x^{10} + g_7x^{12} + h_7x^{14} +}{a_8 + b_8x^2 + c_8x^4 + d_8x^6 + e_8x^8 + f_8x^{10} + g_8x^{12} + h_8x^{14} +} \cdots \\
&\quad \cdots \frac{+ o_7x^{16} + p_7x^{18} + q_7x^{20} + r_7x^{22} + s_7x^{24} + t_7x^{26}}{+ o_8x^{16} + p_8x^{18} + q_8x^{20} + r_8x^{22} + s_8x^{24} + t_8x^{26} + x^{28}}
\end{aligned}
$$

Bibliography

[1] W. J. Thompson, "Numerous neat algorithms for the Voigt profile function," *Computers in Physics* **7**(6), pp. 627–631, 1993.

[2] J. Humlicek, "Optimized computation of the Voigt and complex probability functions," *J. Quant. Spectrosc. Radiat. Transfer* **27**(4), pp. 437–444, 1982.

[3] F. Schreier, "The Voigt and complex error function: A comparison of computational methods," *J. Quant. Spectrosc. Radiat. Transfer* **48**(5/6), pp. 743–762, 1992.

[4] E. E. Whiting, "An empirical approximation to the Voigt profile," *J. Quant. Spectrosc. Radiat. Transfer* **8**, pp. 1379–1384, 1968.

[5] D. P. Edwards, "Atmospheric transmittance and radiance calculations using line–by–line computer models," in *Modelling of the Atmosphere*, L. S. Rothman, ed., *SPIE proceedings* **928**, pp. 94–116, 1988.

Transformation of irradiated to measured spectral distribution due to finite spectral resolution and field of view extent of a Fourier transform spectrometer

F. Hase

Abstract: It is described how instrumental effects on the irradiated spectrum (apodized instrumental line shape, noise, field of view, spectral shift) are simulated by KOPRA. The calculation of derivatives with respect to elevation, spectral shift, ordinate offset, ordinate scale, and instrumental line shape parameters is presented.

1 Introduction

Any real spectrometer has limited spectral resolution and thereby acts as a low pass filter on the irradiated spectral distribution. The measured spectrum can be described locally as the convolution of the incoming spectrum with an appropriate instrumental line shape (ILS). KOPRA performs the convolution with a truncated ILS in the spectral domain using an interpolatoric integration rule.
In case of a Fourier transform spectrometer the ILS is the Fourier transform of the modulation efficiency in the interferogram domain. Thereby the natural ILS assuming constant modulation efficiency up to maximal optical path difference L is the sinc function. In a real instrument the modulation efficiency decreases somewhat with increasing optical path difference due to finite acceptance angle of the interferometer (called self-apodisation). Interferometric misalignment, aberrations in the optical system and imperfect phase correction further distort the ILS and destroy its symmetry. The general ILS is the Fourier transform of a complex valued and wavenumber dependent modulation efficiency.
A multiplicative weighting function working in the interferogram domain can be used to modify the resulting AILS. This procedure is called numerical apodisation. To reduce the sidelobes of the raw ILS, a variety of apodisation functions are in use. KOPRA calculates the AILS from the resulting modulation efficiency in the interferogram domain (including self -apodisation) and handles all common apodisation functions. Instrumental imperfection can be described in terms of a set of 2 parameters: an additional linear modulation loss and a phase error. Derivatives of the spectrum with respect to these parameters can be calculated. In case of MIPAS ENVISAT the ILS model given by BOMEM is used by KOPRA. Alternatively, KOPRA can use an external ILS given in tabulated form.
Any real spectrometer accepts radiation out of a finite solid angle. It acts as a low pass filter on the irradiated spectral distribution with respect to directional variability. The instrumental responsitivity to a point source in infinity as a function of its orientation with respect to the instrumental line of sight characterises the *external* field of view (FOV). The irradiated spectral distribution has to be convolved with this sensitivity function. Since in a Fourier Transform spectrometer the optical path difference depends on the inclination of the wavefront in the interferometer, inhomogenous illumination of the FOV affects the spectral response also. This effect is small in typical situations of remote sensing and KOPRA allows to estimate it. For this purpose, obviously the FOV inside the interferometer (*internal* FOV) has to be used.
In the retrieval process, derivatives to each of the fit parameters are needed. In the context under consideration here KOPRA offers derivatives with respect to

- line of sight (LOS)
- spectral shift
- ordinate scale
- ILS parameters: linear modulation loss and phase error
- BOMEM ILS parameters: retroreflector linear shear variation along z-axis and IR misalignment along y direction

KOPRA offers a noise generator. If no numerical apodisation is performed, the generated Gaussian noise is uncorrelated on a spectral grid of width 1/2L. If numerical apodisation is performed and/or grid spacing differs from 1/2L, the correlations reproduce the statistical behaviour resulting from white Gaussian noise in interferogram domain.

2 Calculation of AILS

KOPRA determines a single sided ($x \geq 0$) and complex-valued modulation efficiency with symmetric real and antisymmetric imaginary parts. The AILS is the Fourier transform of the modulation efficiency. The modulation efficiency $M(x)$ includes the numeric apodisation function, the modulation loss due to self apodisation, the linear modulation loss and the phase error.

$$M(x) = M_{resolution}(x) \times M_{numeric}(x) \times M_{self}(x) \times M_{linear}(x) \times M_{phase}(x) \quad (1)$$

- resolution

  The interferogram is restricted to maximal optical path difference L

  $M_{resolution}(x) = 1 \quad \text{if} \quad |x| \leq L$
  $M_{resolution}(x) = 0 \quad \text{else}$

- apodisation function

  (a) sinc
  $M_{numeric} = 1$

  (b) triangle
  $M_{numeric} = 1 - |x|/L$

  (c) Hamming
  $M_{numeric} = 0.53856 + 0.46144 \times \cos(\pi \times x/L)$

  (d) Blackmann-Harris 3-term
  $M_{numeric} = 0.42323 + 0.49755 \times \cos(\pi \times x/L) + 0.07922 \times \cos(2 \times \pi \times x/L)$

  (e) Blackmann-Harris 4-term
  $$M_{numeric} = 0.35875 + 0.48829 \times \cos(\pi \times x/L) + 0.14128 \times \cos(2 \times \pi \times x/L) + 0.01168 \times \cos(3 \times \pi \times x/L)$$

  (f) Norton-Beer weak
  $M_{numeric} = 0.384093 - 0.087577 \times \left(1 - (x/L)^2\right) + 0.703484 \times \left(1 - (x/L)^2\right)^2$

  (g) Norton-Beer medium
  $M_{numeric} = 0.152442 - 0.136176 \times \left(1 - (x/L)^2\right) + 0.983734 \times \left(1 - (x/L)^2\right)^2$

  (h) Norton-Beer strong
  $M_{numeric} = 0.045335 - 0.554883 \times \left(1 - (x/L)^2\right)^2 + 0.399782 \times \left(1 - (x/L)^2\right)^4$

- self apodisation

  The interferometer has finite acceptance angle. It can be shown, that optical path difference depends on the inclination of the wavefront versus the optical axis. In case of homogeneously illuminated circular internal FOV of semidiameter one finds that this leads to an additional loss of modulation:
  $M_{self} = \frac{\sin(\pi \times \Delta\nu \times x)}{\pi \times \Delta\nu \times x}$ with $\Delta\nu = 0.5 \times \nu \times \alpha^2$
  Note that the self apodisation (and thereby the resulting AILS) depends on spectral position ν. The additional modulation loss describes the consequences of the finite acceptance angle not completely. In addition, the spectral abscissa is scaled by $0.5 \times (\cos(\alpha) + 1)$. Since the latter effect is usually absorbed in the spectral calibration, KOPRA *does not perform any spectral scaling or shift on the calculated spectrum due to self apodisation.*

- linear modulation loss

  This term is used to model the width of the imperfect AILS.

  $M_{linear} = 1 - (1 - a) \times x/L$

  the factor a gives the modulation efficiency at maximal path difference vs ideal instrument.

- phase error

  This term is used to model the asymmetry of the imperfect AILS. The phase error φ is given in radians.

  $M_{phase} = \frac{e^{-i\varphi}}{\cos(\varphi)}$

  The norm of the AILS is fixed by the real part of $M(0)$. The denominator ensures the norm to be unity.

Since the real valued AILS is the Fourier transform of the complex-valued modulation efficiency, the latter is symmetric in the real part and antisymmetric in the imaginary part. Due to this symmetry, the AILS is fully determined by a single sided modulation efficiency interferogram.
KOPRA uses a semianalytic discrete Fourier transform. This method avoids channeling effects and facilitates fast and accurate calculation of the AILS on any grid spacing. The values of the modulation efficiency are calculated at N equidistant optical path differences x_i up to maximal path difference L

$$x_i = \frac{N - i}{N - 1} \times L \qquad \text{with} \quad i = 1, 2, \cdots, N$$

It is assumed, that in between these positions the modulation efficiency can be approximated by linear interpolation to sufficient accuracy. KOPRA uses $N = 200$. This interferogram can be transformed analytically, by breaking both real and imaginary part down into a boxcar and N-1 triangles and adding up the Fourier transforms of these functions. The width of the boxcar is $x_1 = L$ and the widths of the triangles are $x_1, x_2, \cdots, x_{N-1}$. The contribution of the real-part boxcar $g_{box,re}$ is

$$g_{box,re} = Re(M(x_N))$$

and the contribution of the real-part triangle $g_{triang,re}$ to x_i is

$$\begin{aligned} g_{triag,re}(i) &= (N + 1 - i) \times Re(M(N + 1 - i) - M(N + 2 - i)) \\ &\quad - \sum_{k=2}^{i-1} g_{triang,re}(k)/(N + 1 - k) \end{aligned}$$

The contributions for the imaginary-part boxcar and triangles are found accordingly by exchanging real by imaginary parts in the formulas. To achieve a compact notation, we use $k = 2 \times \pi \times \nu$ in the following. Note that the analytic Fourier transforms given below are normalized to unity with respect to the wavenumber abscissa ν.
The Fourier transform of the real-part boxcar is:

$$FT_{box,re}(k, L) = \frac{2 \times \sin(k \times L)}{k}$$

The Fourier transform of the imaginary-part boxcar is:

$$FT_{box,im}(k, L) = \frac{4 \times \sin^2(0.5 \times k \times L)}{k}$$

The Fourier transform of the real-part triangle of width x_i is:

$$FT_{triag,re}(k, x_i) = \frac{4 \times \sin^2(0.5 \times k \times x_i)}{x_i \times k^2}$$

The Fourier transform of the imaginary-part triangle of width x_i is

$$FT_{triag,im}(k, x_i) = \frac{2 \times (x_i \times k - \sin(x_i \times k))}{x_i \times k^2}$$

The AILS then is given by superposition of all analytic contributions:

$$\begin{aligned} AILS(k) &= g_{box,re} \times FT_{box,re}(k, L) + \sum_{i=1}^{N-1} g_{triag,re}(i) \times FT_{triag,re}(k, x_i) \\ &\quad + g_{box,im} \times FT_{box,im}(k, L) + \sum_{i=1}^{N-1} g_{triag,im}(i) \times FT_{triag,im}(k, x_i) \end{aligned}$$

3 AILS Model for MIPAS-ENVISAT

KOPRA uses the semiempiric BOMEM ILS model to construct the AILS of the MIPAS instrument onboard ENVISAT. This model is discussed in detail in PO-RS-DOG-GS-0002. A complex modulation efficiency is calculated:

$$\begin{aligned} M(\nu, x) &= M_{IR}(\nu, x) \times M_L(\nu, x) \times M_V(\nu, x) \times M_{SP}(\nu, x) \times M_{Ld}(\nu, x) \\ &\quad \times M_{Lw}(\nu, x) \times M_{Lf}(\nu, x) \times M_{Res}(x) \times M_{numeric}(x) \end{aligned}$$

M_{IR}	complex modulation function related to IR-misalignment
M_L	complex modulation function related to laser misalignment
M_V	complex modulation function related to optical speed
M_{SP}	complex modulation function related to sampling distortions at turn around
M_{Ld}	complex modulation function related to laser drift
M_{Lw}	real modulation function related to white noise of laser
M_{Lf}	real modulation function related to 1/f noise of laser
M_{Res}	real modulation function related to limited optical path difference
$M_{numeric}$	real modulation function related to numerical apodisation (Norton-Beer strong)

These modulation functions depend on 24 parameters. Their values are specified by the user in the KOPRA input file. Two out of these are designated to describe AILS imperfections. They are marked with asterisks.

(a) maximal optical path difference [nominal/default value: 20 cm]

(b) infrared misalignment y-direction [nominal/default value: 0 rad] *

(c) infrared misalignment z-direction [nominal/default value: 2×10^{-4}rad]

(d) retroreflector linear shear along y [nominal/default value: 4×10^{-3}cm]

(e) retroreflector linear shear along z [nominal/default value: 4×10^{-3}cm]

(f) linear shear variation along y [nominal/default value: 0 , dimensionless]

(g) linear shear variation along z [nominal/default value: 0 , dimensionless] *

(h) full interferometer divergence along y [nominal/default value: 5.4×10^{-3}rad]

(i) full interferometer divergence along z [nominal/default value: 9.0×10^{-3}rad]

(j) blur angular width along y [nominal/default value: 5.2×10^{-4}rad]

(k) blur angular width along z [nominal/default value: 3.3×10^{-4}rad]

(l) laser misalignment along y [nominal/default value: 1.5×10^{-4}rad]

(m) laser misalignment along z [nominal/default value: 1.5×10^{-4}rad]

(n) optical speed of interferometer [nominal/default value: 10.0 cm/s]

(o) initial sampling pertubation [nominal/default value: 4×10^{-8}cm]

(p) time constant of exponential attentuation of initial sampling pertubation [nominal/default value: 0.16 s]

(q) time constant of exponential attentuation of initial speed fluctuation [nominal/default value: 0.016 s]

(r) initial relative speed fluctuation at beginning of scan [nominal/default value: 0.03, dimensionless]

(s) gain slope of IR electrical response [nominal/default value: -0.22, dimensionless]

(t) mismatched delay between electronic response and ADC trigger [nominal/default value: 1.4×10^{-6}s]

(u) laser wavenumber [nominal/default value: 7692.0 cm^{-1}]

(v) relative drift rate of laser wvnr [nominal/default value: 1.0×10^{-8}, dimensionless]

(w) bandwith laser white noise [nominal/default value: 2×10^{7}Hz]

(x) bandwith laser 1/f noise [nominal/default value: 0 Hz]

To avoid confusion, it is important to mention that the BOMEM ILS model and the KOPRA AILS model described above introduce complex modulation functions with different meaning. The KOPRA modulation function is just the inverse transform of a real AILS. The symmetry properties of the modulation function guarantee the imaginary part of the AILS to be zero. The BOMEM modulation function has a physical meaning: it characterises amplitude and phase orientation of the modulation related to a monochromatic input signal. The ILS is proportional to the real part of the Fourier transform of the modulation function.
KOPRA handles the BOMEM model in the following way: The double-sided modulation function $M(\nu,x)$ is calculated as given in document PO-RS-DOG-GS-0002. Then $M(\nu,x)$ is scaled and rotated in the complex plane to achieve a real $\tilde{M}(\nu,0)$ normalised to unity at zero path difference:

$$\tilde{M}(\nu,x) = M(\nu,x) \times e^{-i\varphi}/|M(\nu,0)| \qquad \text{with} \qquad M(\nu,0) = |M(\nu,0)| \times e^{i\varphi}$$

A single sided modulation function $\tilde{\tilde{M}}(\nu,x)$ is calculated from $\tilde{M}(\nu,x)$ with symmetric real and antisymmetric imaginary part.

$$\tilde{\tilde{M}}(\nu,x) = 0.5 \times (Re(\tilde{M}(\nu,x) + \tilde{M}(\nu,-x)) + i \times Im(\tilde{M}(\nu,x) - \tilde{M}(\nu,-x)))$$

The Fourier transform of $\tilde{\tilde{M}}(\nu,x)$ is the ILS. It is performed using the semianalytic discrete transform described above.

4 Convolution

The measured spectrum $\tilde{S}(\nu)$ is the convolution of the irradiated spectrum $S(\nu)$ with the instrumental lineshape $AILS(\nu - \nu_0)$.

$$\tilde{S}(\nu_0) = \int S(\nu) \times AILS(\nu_0 - \nu)d\nu \qquad \text{with} \qquad \int AILS(\nu)d\nu = 1$$

The AILS transmits the incoming flux at spectral position ν to measured flux at spectral position ν_0. Despite the fact, that the AILS is a function of ν it is assumed that all significant contributions to the measured flux at ν_0 are transmitted by the AILS strictly valid only at position ν_0. Moreover, the measured flux is usually calculated in a whole microwindow extending from ν_1 to ν_2. KOPRA then uses the AILS valid for the arithmetic mean wavenumber $0.5 \times (\nu_1 + \nu_2)$. In the practice of remote sensing, the first approximation is always justified. The second approximation demands $|(\nu_1 - \nu_2)|/(\nu_1 + \nu_2) \leq 0.02(0.1)$ as a rule of thumb for a negligible variation of modulation efficiency of 0.01 (0.05) at maximal path difference.
Since the calculation of $S(\nu)$ has to be restricted to a finite interval, the integral has to be truncated. The modification must not be done in the following way

$$\tilde{S}(\nu_0) = \int S(\nu) \times AILS(\nu_0 - \nu) \times W(\nu_1, \nu_2)d\nu$$

$$\begin{aligned} \text{with} \qquad W(\nu_1, \nu_2) &= 1 \qquad \text{if} \quad (\nu - \nu_1)(\nu - \nu_2) < 0 \\ W(\nu_1, \nu_2) &= 0 \qquad \text{if} \quad (\nu - \nu_1)(\nu - \nu_2) \geq 0 \end{aligned}$$

because this is equivalent to the convolution of the untruncated AILS with the windowed spectrum $S(\nu) \times W(\nu_1, \nu_2)$. The sharp boarders of the window give rise to the Gibbs phenomenon: unwanted oscillations spread from the boarders into the spectrum.
To avoid these oscillations, KOPRA calculates the spectrum in an enlarged interval extending from $\nu_{1e} = \nu_1 - \Delta\nu$ to $\nu_{2e} = \nu_2 + \Delta\nu$ and the truncation is assigned to the AILS

$$\begin{aligned} \tilde{S}(\nu_0) &= \int_{\nu_0 - \Delta\nu}^{\nu_0 + \Delta\nu} S(\nu) \times L(\nu_0 - \nu)d\nu \\ &= \int S(\nu) \times AILS(\nu_0 - \nu) \times W(\nu_0 - \Delta\nu, \nu_0 + \Delta\nu)d\nu \end{aligned}$$

KOPRA chooses the truncation window according to the users specification to ensure that the transmission $T(\nu) = AILS(\nu)/AILS(0)$ from outside the window is lower than a certain threshold. Due to the truncation, the norm of the AILS is slightly lower than unity (see Table 4).
KOPRA solves the radiative transfer equation at a finite number of discrete spectral positions. The calculated incoming spectrum is interpolated to an equidistant spectral grid before the convolution is performed. The grid spacing has to be dense enough to depict even the sharpest details to be expected in the spectrum. In the atmospheric spectrum, these are the cores of purely Doppler-broadened lines. The hwhm γ_D of a line at spectral position ν emerging from a species of mass m in an atmospheric layer of temperature T is

$$\begin{aligned} \gamma_D &= \sqrt{\frac{2kT}{m}} \times \frac{1}{c \ln 2} \times \nu \\ \gamma_D \left[cm^{-1}\right] &\approx 6.22 \times 10^{-7} \times \sqrt{\frac{T[K]}{m[amu]}} \times \nu \left[cm^{-1}\right] \end{aligned}$$

Apodisation	T=0.01	T=0.001	T=0.0001
boxcar	16/L	160/L	1600/L
	0.9957	0.9996	1.000
triangle	3/L	10/L	32/L
	0.9666	0.9899	0.9968
Hamming	1/L	23/L	510/L
	1.0029	0.9997	1.0000
3-term Blackmann-Harris	1.5/L	2/L	17/L
	0.9999	1.0000	0.9999
4-term Blackmann-Harris	2/L	2.5/L	14/L
	0.9998	0.9999	0.9999
Norton-Beer weak	9/L	85/L	860/L
	0.9971	0.9997	1.0000
Norton-Beer medium	4/L	40/L	510/L
	0.9977	0.9997	1.0000
Norton-Beer strong	1.5/L	14/L	140/L
	1.0005	0.9997	1.0000

Table 1: Truncation radius for AILS and norm of truncated AILS (no self-apodisation). L denotes the maximal optical path difference.

If higher atmospheric layers contribute noticeably in the spectrum, the user defined fine grid spacing should not exceed γ_D. In a strict sense, a sampling dense enough to determine the continuous function $S(\nu)$ exactly is impossible, because the incoming spectrum is *not bandwith limited*! In practice, the KOPRA user should determine a sufficient sampling density for the defined situation by a control run using halved spacing.
In case of a Fourier transform spectrometer, the highest detectable sinusoidal modulation in the irradiated spectrum is determined by the maximal optical path difference of the instrument L. The limiting spectral modulation along the wavenumber abscissa is then given by 1/L. Therefore, according to the sampling theorem, the gridpoint density in the measured spectrum must be at least 1/2L. *Because the measured spectrum is bandwith limited*, the continuous function $\tilde{S}(\nu)$ is then completely determined.
KOPRA combines the convolution with the reduction of gridpoint density. The convolution integral is evaluated on the user defined output grid. The simplest interpolatoric integration rule is applied:

$$\tilde{S}(\nu_i) = \int S(\nu) \times AILS(\nu_i - \nu) d\nu \approx \Delta\nu \times \sum_{n=-N}^{N} AILS(n) \times S(i-n)$$

The spectral stepwidth is denoted by $\Delta\nu$. Since the integrand is localised (tends to zero near the boarders of the integration interval), the accuracy of the result cannot be enhanced substantially by means of any higher polynomial integration rule.

5 Noise Generator

KOPRA offers a noise generator. In case of a Fourier transform spectrometer, it is usually assumed that the noise is Gaussian and independent from sampling position in the interferogram. We follow this assumption here, because noise contributions

of differing statistical behaviour are highly instrument and scene specific.
The irradiated spectrum calculated by KOPRA is interpolated to an equidistant fine grid spacing before the convolution with the AILS is performed (see section 'convolution'). KOPRA introduces the noise in the spectral domain: the statistically independent Gaussian noise is generated on this same fine grid. This noise is convolved with an appropriate AILS_{noise} and thereby adopts the correct statistical properties found in the measured spectrum. Note that the AILS to be applied to the irradiated spectrum and AILS_{noise} are *not identical.* The latter does incorporate neither self-apodisation nor modulation loss due to misalignment (see section 'Calculation of AILS'), because the variation of modulation efficiency with optical path difference does not affect the noise level, which depends on total photon number and detector characteristics.
The statistically independent Gaussian noise of standard deviation σ_f on the fine grid is calculated using

$$y = \sigma_f \times \cos(2 \times \pi \times x_1) \times \sqrt{-2 \times log \times x_2}$$

with evenly distributed random numbers $x_1, x_2 \in \{0, 1\}$. KOPRA uses the standard Fortran 90 calling sequence for random numbers. Since no standard algorithm is defined by the language, the quality of the random numbers generated may depend on the computer and compiler used.
The convolution with AILS_{noise} mixes the statistically independent sample points. If no numerical apodisation is performed, AILS_{noise} is the sinc-function to maximal optical path difference L. The distance between two independent sample points is enlarged from the fine grid spacing $\Delta\nu$ to 1/2L. Due to this smoothing, the standard deviation on the coarse grid σ_c is reduced:

$$\sigma_c = \sigma_f \times \sqrt{2 \times \Delta\nu \times L}$$

The KOPRA user specifies σ_c measured at each sample point in the resulting spectrum assuming a sinc-shaped AILS_{noise}. If numeric apodisation is performed, the standard deviation at each sample point in the resulting spectrum is smaller than σ_c. This convention is advisable, because it keeps the quality of the measurement constant when comparing the retrieval quality using different kinds of apodisation functions.

6 Field of View (FOV)

If the radiation field shows a significant directional variability inside the external field of view (FOV), an averaged spectrum representative for the FOV has to be constructed by weighting the radiance with the responsivity distribution over the FOV.
In general, in the case of an upward looking remote sensing spectrometer, the FOV convolution can be neglected and the spectrum irradiated along the instrumental line of sight is representative. In case of a limb sounding instrument strong vertical gradients in radiance arise especially at low tangent heights. The instrumental responsivity $R(\chi_1, \chi_2)$ to a point source in infinity as a function of its orientation with respect to the instrumental line of sight characterises the external field of view. The FOV is assumed to be small enough to project R onto a plane without significant distortion. χ_1, χ_2 are Cartesian coordinates measured in radians in this plane. The χ_1 axis is tangential to the vertical great circle containing the instrumental line of sight and the χ_2 axis is perpendicular to it and parallel to the horizon. The origin of the system coincides with the instrumental line of sight. The irradiated spectrum S

can be assumed to be independent from χ_2 and the averaged spectrum S_{av} is given by

$$\begin{aligned} S_{av} &= \int\int_{FOV} S(\chi_1) \times R(\chi_1,\chi_2) d\chi_1 d\chi_2 \\ &= \int_{\chi_1} S(\chi_1) \times \int_{\chi_2} R(\chi_1,\chi_2) d\chi_1 d\chi_2 \\ &= \int_{\chi_1} S(\chi_1) \times R_{vert}(\chi_1) d\chi_1 \end{aligned}$$

with

$$\int\int_{FOV} R(\chi_1,\chi_2) d\chi_1 d\chi_2 = 1 \quad \text{and} \quad R_{vert}(\chi_1) = \int_{\chi_2} R(\chi_1,\chi_2) d\chi_2$$

KOPRA subdivides the FOV in 20 horizontal bands of equal thickness. The user specifies the weight of each band $R_{vert,i}$ and the total vertical extent of the FOV H according to the characteristics of his instrument.

$$R_{vert,i} = \int_{\chi_1=\chi_i-\Delta\chi/2}^{\chi_i+\Delta\chi/2} R_{vert}(\chi_1) d\chi_1$$

with

$$\Delta\chi = 0.05 \times H \quad \text{and} \quad \chi_i = \Delta\chi \times (i - 10.5) \qquad i = 1, 2, 3, \cdots, 20$$

The irradiated spectrum has to be calculated for at least 3 relevant directions. Linear interpolation between adjacent calculated spectra yields $S_{interpol}(i)$ in the centre of band i. The averaged spectrum is approximated by KOPRA using

$$S_{av} \approx \frac{\sum_{i=1}^{20} R_{vert,i} \times S_{interpol}(i)}{\sum_{i=1}^{20} R_{vert,i}}$$

As can be seen, KOPRA does not demand $R_{vert} = 1$ for the weights of the bands in the input file. Any set of weights proportional to the normalised R_{vert} is valid also.

Since in a Fourier Transform spectrometer the optical path difference depends on the inclination of the wavefront in the interferometer, inhomogenous illumination of the FOV affects the spectral response also. This effect is of secondary importance. KOPRA allows to estimate the consequences on the measured spectrum assuming a circular internal FOV.

The relation between the optical path difference x and the inclination of the wavefront α versus optical axis *measured inside the interferometer* is

$$x(\alpha) = x(0) \times \cos(\alpha)$$

As a consequence, the contribution to the measured spectrum out of a cone surface centered on the optical axis has a common spectral shift versus the irradiated spectrum. A monochromatic line located at ν_0 is shifted to lower wavenumber $\nu(\alpha)$ in the measured spectrum. In small angle approximation one finds

$$\nu(\alpha) = \nu_0 \times (1 - 0.5 \times \alpha^2) \quad \text{and thus} \quad d\nu(\alpha) = -\nu_0 \times \alpha \times d\alpha$$

We consider an isotropic radiation field. The spectrum S contains a single monochromatic line with flux F_0

$$S(\nu) = F_0 \times \delta(\nu - \nu_0)$$

This line is mapped into a boxcar in the measured spectrum extending from $\nu_0 \times (1 - 0.5 \times \alpha_{max}^2)$ to ν_0. α_{max} is the maximal wavefront inclination accepted by the interferometer. The radiance (height of the boxcar) is $2F_0/\nu_0\alpha_{max}^2$. This is understandable in the following way: each distinct cone surface leads to a contribution proportional to its area $2\pi\alpha d\alpha$, that is smeared along the spectral abscissa over the interval $\nu_0\alpha d\alpha$. Both area and spectral width are proportional to α and so the radiances $\tilde{S}$ of all contributions are equal and a boxcar results.

$$\tilde{S}(\nu(\alpha)) = \frac{F_0}{\pi\alpha_{max}^2} \times \frac{2 \times \pi \times \alpha \times d\alpha}{\nu_0 \times \alpha \times d\alpha} = \frac{2 \times F_0}{\nu_0 \times \alpha_{max}^2}$$

The Fourier transform of the boxcar leads to the sinc-shaped self apodisation term (see section 'Calculation of AILS, self apodisation') in case of isotropic illumination. Next we consider an anisotropic radiation field. The spectrum S to consist of a single monochromatic line as before but the flux F depends on χ_1. We expand a Maclaurin series

$$\begin{aligned} S(\nu, \chi_1) &= F(\chi_1) \times \delta(\nu - \nu_0) \\ &= (F(0) + F' \times \chi_1 + 0.5 \times F'' \times \chi_1^2 + \cdots) \times \delta(\nu - \nu_0) \end{aligned}$$

To find the contribution of each cone surface, we have to integrate the flux in each concentric circle in the χ_1, χ_2 plane. We introduce the azimuthal coordinate ρ around the optical axis. The direction $\rho = 0$ is the positive χ_2 axis. The contribution of each cone surface is given by

$$\begin{aligned} \tilde{S}(\nu(\alpha)) &= \frac{1}{\pi \times \alpha_{max}^2} \times \frac{2}{\nu_0 \times \alpha} \\ &\quad \times \int_{\rho=-\pi/2}^{\pi/2} (F(0) + F' \times \chi_1 + 0.5 \times F'' \times \chi_1^2 + \cdots) \times \alpha \times d\rho \end{aligned}$$

The integration is performed by substituting the integration variable using $\alpha \times \sin(\rho) = \chi_1$. This leads to

$$\begin{aligned} \tilde{S}(\nu(\alpha)) &= \frac{1}{\pi \times \alpha_{max}^2} \times \frac{2}{\nu_0 \times \alpha} \\ &\quad \times \int_{\chi_1=-\alpha}^{\alpha} (F(0) + F' \times \chi_1 + 0.5 \times F'' \times \chi_1^2 + \cdots) \times \frac{1}{\sqrt{1 - \chi_1^2/\alpha^2}} d\chi_1 \end{aligned}$$

Due to the symmetry properties of the integrand all terms containing odd derivatives of F cancel. KOPRA omits derivatives higher than 2, therefore the error is of order O(4).

$$\tilde{S}(\nu(\alpha)) \approx \frac{1}{\pi \times \alpha_{max}^2} \times \frac{2}{\nu_0 \times \alpha} \times (\pi \times F(0) \times \alpha + 0.25 \times \pi \times F'' \times \alpha^3)$$

Substitution of α^2 using $\alpha^2 = 2 \times (1 - \nu/\nu_0)$ leads to

$$\tilde{S}(\nu(\alpha)) \approx \frac{2 \times F(0)}{\alpha_{max}^2 \times \nu_0} + \frac{F'' \times (1 - \nu/\nu_0)}{\alpha_{max}^2 \times \nu_0}$$

The additional second term describes a triangle of zero height at ν_0 and height $F''/2\nu_0$ at $\nu(\alpha_{max})$. The monochromatic line is mapped into a boxcar (connected to F(0)) with a slant increasing towards lower wavenumber (connected to F'') upon. An arbitrary spectrum can be interpreted as a dense superposition of monochromatic peaks. The relations given above for the monochromatic example therefore hold for the general case also.

To find $\tilde{S}$ for an arbitrary irradiated spectrum, KOPRA convolves S with a normalised boxcar extending along the spectral abscissa from $-0.25 \times \nu_0 \times \alpha_{max}^2$ to $+0.25 \times \nu_0 \times \alpha_{max}^2$. KOPRA calculates the second derivative of S with respect to the coordinate $\tilde{\chi_1} = \chi_1/\alpha_{max}$ fixing the maximal inclination in the interferometer as unit angle. $\delta^2 S/\delta\tilde{\chi_1}^2$ is convolved with a triangle. At $+0.25 \times \nu_0 \times \alpha_{max}^2$ the height of the triangle is zero. At $-0.25 \times \nu_0 \times \alpha_{max}^2$ the height of the triangle is a fourth of the boxcar height. Note that the bases of boxcar and triangle are centered on $\nu = 0$. KOPRA *does not perform any spectral scaling or shift on the calculated spectrum due to the finite FOV.* It is assumed that the spectral scaling due to finite FOV is absorbed in the spectral calibration of the instrument.
We expect the approximation to give more accurate results if the value for $\delta^2 S/\delta\tilde{\chi_1}^2$ is chosen appropriate for the extended FOV as a whole instead of using the value of $\delta^2 S/\delta\tilde{\chi_1}^2$ at $\tilde{\chi_1} = 0$. KOPRA constructs the representative value of the second derivative from three FOV-averaged spectra S_{av} to nominal and slightly up- and downward shifted line of sights.

7 Derivatives

In the retrieval process, derivatives with respect to all fitted parameters are needed. In the context under consideration here KOPRA offers derivatives with respect to elevation, spectral shift, ordinate offset, ordinate scale and parameters for AILS description.
We call the spectral distributions at the location of the instrument for specified LOS and FOV 'irradiated spectra' and the affiliated results of the measurements (without scale and offset consideration) 'instrumental spectra'. KOPRA models irradiated spectra and instrumental spectra yielding 'synthetic irradiated spectra' and 'synthetic instrumental spectra'. Scale and offset are considered via a multiplicative and additive constant, respectively, applied to the synthetic instrumental spectra: $S_{calcor} = aS_{instr} + b$. Spectra of the latter kind are used in the calculation of the derivatives.

- elevation

  The derivation with respect to elevation is calculated from the difference of two synthetic instrumental spectra, one referring to the specified line of sight, the other one to a slightly modified line of sight.
  If the FOV extent is neglected, KOPRA has only a single synthetic irradiated spectrum at its disposal. The complete forward calculation has to be repeated performing a renewed KOPRA run. If the FOV extent is taken into account, KOPRA has several synthetic irradiated spectra referring to different elevations at its disposal, and the two synthetic measured spectra needed are found by interpolation using the given set of synthetic irradiated spectra (see section 'field of view'). The increment in elevation is set to a small fraction (0.025) of the full vertical extent of the FOV.

- spectral shift

  The derivation with respect to spectral shift is calculated from the synthetic instrumental spectrum $S(i), i = 1, 2, \cdots, n_{max}$ by

  $$\frac{\delta S}{\delta \nu}(1) = \frac{S(2)-S(1)}{\Delta\nu}$$
  $$\frac{\delta S}{\delta \nu}(i) = \frac{S(i+1)-S(i-1)}{2\Delta\nu} \qquad 2 \leq i \leq n_{max} - 1$$
  $$\frac{\delta S}{\delta \nu}(n_{max}) = \frac{S(n_{max})-S(n_{max}-1)}{\Delta\nu}$$

- ordinate offset

The derivative of S_{calcor} with respect to an additional constant is unity.

- ordinate scale

  The derivative of S_{calcor} with respect to the multiplicative constant a is the synthetic instrumental spectrum S_{instr}.

- AILS parameters: modulation efficiency and phase error

  To calculate the derivative with respect to the modulation efficiency parameter m (see section 'Calculation of AILS') the AILS is calculated twice from the beginning. The derivative of the synthetic instrumental spectrum $S_{instr}(m_0)$ with respect to m is the convolution of the synthetic irradiated spectrum S_{irr} with the derivative of the AILS with respect to m.

$$\frac{\delta S_{instr}(m_0)}{\delta m} = \frac{AILS(m_0 + \Delta m) - AILS(m_0)}{\Delta m} \otimes S_{irr}$$

  The derivative with respect to phase error φ is calculated in the same way, but the amplitude of the modulation efficiency remains unchanged and is not recalculated. The increments used by KOPRA are $\Delta m = 10^{-3}$ and $\Delta\varphi = 10^{-4}$, well below the practical limit of retrieval accuracy.

- BOMEM AILS parameters: retroreflector linear shear variation along z-axis and IR misalignment along y direction

  The AILS is calculated twice from the beginning. The increments used by KOPRA are 10^{-4} in case of the linear shear variation along z-axis and 10^{-5} rad in case of IR misalignment, well below the practical limit of retrieval accuracy.

Derivatives and interface to the retrieval

M. Höpfner

Abstract: The simultaneous determination of the atmospheric spectrum and it's derivatives with respect to various atmospheric and instrumental parameters is the basis for the use of KOPRA in retrieval processors. In this part the basic principles implemented for the calculation of derivatives with respect to atmospheric quantities are described. The derivatives with respect to instrumental parameters are summarized.

1 Introduction

Besides the calculation of the atmospheric spectrum convolved with the instrumental line shape and the field-of-view of the interferometer KOPRA is dedicated to the determination of the derivatives of the spectrum with respect to various retrieval parameters (which are just called 'parameters' in the following). The parameters can roughly be subdivided in two parts: the atmospheric ones and the ones describing the instrumental performance. Atmospheric parameters are:

- temperature
- pressure
- volume mixing ratio
- non-LTE/LTE population ratio
- temperature gradient
- volume mixing ratio gradient
- atmospheric continuum

Instrumental parameters are:

- line-of-sight
- offset
- scale
- wavenumber shift
- ils-parameters

This subdivision reflects the internal procedure for the calculation of derivatives: the atmospheric derivatives need to be considered during the ray-tracing and mass calculation, during the absorption-coefficient determination and during the radiative transfer. Instrumental derivatives are determined after the monochromatic spectrum is known: during the ils-convolution and fov-calculation and the offset/scale addition/multiplication. Derivatives with respect to pressure are not supported by KOPRA in a fast way but calculated numerically. In the following we will first describe the principles for the calculation of atmospheric derivatives and then for instrumental ones. At the end the interface to retrieval approaches via various possible parameterizations of the retrieval quantities is presented.

2 Calculation of derivatives wrt atmospheric parameters

2.1 The formulas

In order to explain the implemented method for the calculation of derivatives wrt atmospheric parameters we start with the formula for the radiative transfer:

$$S = B(T_{back}) \prod_{i=1}^{N} \tau_i + \sum_{i=1}^{N} \tilde{J}_i (1 - \tau_i) \prod_{k=i+1}^{N} \tau_k \qquad (1)$$

i is the index for the layers with layer 1 the farthest and layer N the nearest with respect to the observer. $\tilde{J}_i$ is the source function for layer i and τ_i the single layer transmission:

$$\tau_i = exp[-\delta_i] \tag{2}$$

δ_i is the optical depth for layer i:

$$\delta_i = \delta_{e,Aerosol,i} + \sum_{g=1}^{G} \left[\sigma_{a,ig} + \sum_{b=1}^{B_g} \alpha_{igb} \sigma_{a,igb} \right] u_{ig} \tag{3}$$

$\delta_{e,Aerosol,i}$ is the optical depth of aerosol extinction, $\sigma_{a,ig}$ the absorption coefficient for all LTE bands of gas g, $\sigma_{a,igb}$ the LTE absorption coefficient for NLTE band b of gas g, α_{igb} the ratio of NLTE to LTE absorption coefficient for NLTE band b of gas g, u_{ig} the column density for gas g in layer i, $B_g$the number of NLTE bands for gas g, and G the number of gases.
In the most simple case:

$$\tilde{J}_i = B(T_{Air,i}), \tag{4}$$

i.e. the Planck function for the Curtis-Godson temperature of air for layer i.
For NLTE and more exact calculations:

$$\tilde{J}_i = \frac{\hat{J}_i}{\delta_i} = \frac{B(T_{Air,i})\delta_{a,Aerosol,i} + \sum_{g=1}^{G} \left[B(T_{ig})\sigma_{a,ig} + \sum_{b=1}^{B_g} J_{igb}\alpha_{igb}\sigma_{a,igb} \right] u_{ig}}{\delta_i} \tag{5}$$

$\delta_{a,Aerosol,i}$ is the optical depth of aerosol absorption, $B(T_{ig})$ the Planck function for the Curtis-Godson temperature of gas g for layer i, and J_{igb} the NLTE source function for NLTE band b of gas g and layer i.
The aim is the calculation of the derivatives of the spectrum S with respect to the atmospheric retrieval parameters q_m of some quantity. This can be written as:

$$\frac{dS}{dq_m} = \sum_{i=1}^{N} \left(\frac{dS}{d\tilde{J}_i} \frac{d\tilde{J}_i}{dq_m} + \frac{dS}{d\tau_i} \frac{d\tau_i}{dq_m} \right). \tag{6}$$

Parallel to the radiative transfer calculation the derivative of the radiance with respect to the layer transmission is determined by

$$\begin{aligned} \frac{dS}{d\tau_i}\tau_i &= B(T_{back}) \prod_{k=1}^{N} \tau_k + \sum_{j=1}^{i-1} \tilde{J}_j (1-\tau_j) \prod_{k=j+1}^{N} \tau_k - \tilde{J}_i \prod_{k=i}^{N} \tau_k \\ &= \prod_{k=i}^{N} \tau_k \left[B(T_{back}) \prod_{k=1}^{i-1} \tau_k + \sum_{j=1}^{i-1} \tilde{J}_j (1-\tau_j) \prod_{k=j+1}^{i-1} \tau_k - \tilde{J}_i \right] \\ &= \prod_{k=i}^{N} \tau_k \left[S_{i-1} - \tilde{J}_i \right] \end{aligned} \tag{7}$$

The last term (1st row) is the derivative of the emission of layer i attenuated by all layers between i and the observer and the first two terms (1st row) are the derivative of the attenuation for the radiation of each layer up to layer i. Note that the sum (1st row) is the sum over the contribution functions for layer 1 to layer i-1. The formula in the 3rd row is essential for the implementation in the radiative transfer

code: the derivative of the spectrum wrt transmission of layer i can be calculated successively when the radiance leaving layer i-1 (S_{i-1}) and the source function of the actual layer is known.
The derivative of the spectrum wrt the source function of layer i is just:

$$\frac{dS}{d\tilde{J}_i} = (1 - \tau_i) \prod_{k=i+1}^{N} \tau_k \tag{8}$$

At this point we have to distinguish between the various kinds of atmospheric parameters wrt which we want to calculate the derivatives.

2.1.1 Derivatives wrt volume mixing ratio and volume mixing ratio gradient parameters

We neglect the dependence of the source function and the absorption coefficients on volume mixing ratio (through Curtis-Godson temperature and pressure) and need only to determine the derivative of the layer transmission:

$$\begin{aligned} \frac{d\tau_i}{dq_{vmr,mg}} &= -\tau_i \frac{d\delta_i}{dq_{vmr,mg}} = -\tau_i \frac{d\delta_i}{du_{ig}} \frac{du_{ig}}{dq_{vmr,mg}} \\ &= -\tau_i \left[\sigma_{a,ig} + \sum_{b=1}^{B_g} \alpha_{igb} \sigma_{a,igb} \right] \frac{du_{ig}}{dq_{vmr,mg}} \end{aligned} \tag{9}$$

$q_{vmr,mg}$ is the m^{th} (vmr or vmr-gradient) parameter for gas g.
Hence, during the calculation of the path variables the derivatives of the partial columns u_{ig} of gas g for each layer i with respect to the vmr (or vmr gradient) parameters must be determined.

2.1.2 Derivatives wrt aerosol absorption coefficient parameters

a) for cases in which $\tilde{J}_i = B(T_{Air,i})$, the source term is not dependent on aerosol absorption and therefore:

$$\begin{aligned} \frac{d\tau_i}{dq_{Aerosol,m}} &= -\tau_i \frac{d\delta_i}{dq_{Aerosol,m}} \\ &= -\tau_i \frac{d\delta_i}{d\delta_{e,Aerosol,i}} \frac{d\delta_{e,Aerosol,i}}{dq_{Aerosol,m}} = -\tau_i \frac{d\delta_{a,Aerosol,i}}{dq_{Aerosol,m}} \end{aligned} \tag{10}$$

b) for cases in which the source term $\tilde{J}_i$ is dependent on aerosol absorption additionally to a) we have to determine:

$$\frac{d\tilde{J}_i}{dq_{Aerosol,m}} = \frac{\delta_i B(T_{Air,i}) - \hat{J}_i}{\delta_i^2} \frac{d\delta_{a,Aerosol,i}}{dq_{Aerosol,m}} \tag{11}$$

$q_{Aerosol,m}$ is the m^{th} (aerosol absorption) parameter (the microwindow dependence is skipped in this context).
Hence, during the calculation of the path variables the derivatives of the aerosol absorption optical depths $\delta_{a,Aerosol,i}$ for each layer i with respect to the aerosol absorption parameters must be determined.

2.1.3 Derivatives wrt temperature (and temperature gradient) parameters

The derivative of the layer transmission wrt temperature parameters is:

$$\frac{d\tau_i}{dq_{T,m}} = -\tau_i \frac{d\delta_i}{dq_{T,m}} = -\tau_i \frac{d\delta_i}{dT_{Air,i}} \frac{dT_{Air,i}}{dq_{T,m}} \tag{12}$$

with:

$$\begin{aligned} \frac{d\delta_i}{dT_{Air,i}} &= \frac{d}{dT_{Air,i}} \sum_{g=1}^{G} \left[\sigma_{a,ig} + \sum_{b=1}^{B_g} \alpha_{igb} \sigma_{a,igb} \right] u_{ig} \\ &= \sum_{g=1}^{G} \left[\frac{d\sigma_{a,ig}}{dT_{ig}} + \sum_{b=1}^{B_g} \left(\frac{d\alpha_{igb}}{dT_{ig}} \sigma_{a,igb} + \frac{d\sigma_{a,igb}}{dT_{ig}} \alpha_{igb} \right) \right] u_{ig} \\ &+ \sum_{g=1}^{G} \left[\sigma_{a,ig} + \sum_{b=1}^{B_g} \alpha_{igb} \sigma_{a,igb} \right] \frac{du_{ig}}{dT_{ig}} \end{aligned} \tag{13}$$

$q_{T,m}$ is the m$^{\text{th}}$ temperature (or temperature gradient) parameter.
During the calculation of the path variables the derivatives of the Curtis-Godson temperatures for air wrt the temperature parameters must be determined.
The derivatives of the absorption coefficients wrt temperature is calculated in parallel to the absorption coefficients themselves. The absorption coefficients are the product of line-intensity and line-profile:

$$\sigma_{a,ig} = \sum_{n=1}^{N_g} A_{ign} \Phi_{ign} \tag{14}$$

A_{ign} is the line intensity for line n, gas g, N_g the number of lines for gas g, and layer i and Φ_{ign} the line profile. Since the most important temperature dependence in this formula stems from the line intensity we calculate only that derivative (also because the derivative calculation of the line profile function wrt T would be too time consuming). Therefore,

$$\frac{d\sigma_{a,ig}}{dT_{ig}} = \sum_{n=1}^{N_g} \frac{dA_{ign}}{dT_{ig}} \Phi_{ign} \tag{15}$$

Two cases have to be distinguished for determination of the source function derivatives wrt temperature:
a) for cases in which $\tilde{J}_i = B\left(T_{Air,i}\right)$:

$$\frac{d\tilde{J}_i}{dq_{T,m}} = \frac{dB\left(T_{Air,i}\right)}{dT_{Air,i}} \frac{dT_{Air,i}}{dq_{T,m}} \tag{16}$$

b) for cases in which the source term $\tilde{J}_i$ is calculated for NLTE:

$$\frac{d\tilde{J}_i}{dq_{T,m}} = \frac{\delta_i \frac{d\hat{J}_i}{dT_{Air,i}} + \hat{J}_i \frac{d\delta_i}{dT_{Air,i}}}{\delta_i^2} \frac{dT_{Air,i}}{dq_{T,m}} \tag{17}$$

with:

$$\begin{aligned}\frac{d\hat{J}_i}{dT_{Air,i}} &= \frac{d}{dT_{Air,i}} B\left(T_{Air,i}\right)\delta_{a,Aerosol,i} + \sum_{g=1}^{G}\left[B\left(T_{ig}\right)\sigma_{a,ig} + \sum_{b=1}^{B_g} J_{igb}\alpha_{igb}\sigma_{a,igb}\right]u_{ig} \\ &= \frac{dB\left(T_{Air,i}\right)}{dT_{Air,i}}\delta_{a,Aerosol,i} + \sum_{g=1}^{G}\left[B\left(T_{ig}\right)\sigma_{a,ig} + \sum_{b=1}^{B_g} J_{igb}\alpha_{igb}\sigma_{a,igb}\right]\frac{du_{ig}}{dT_{ig}} \quad (18) \\ &+ \sum_{g=1}^{G}\left[\begin{array}{l}\frac{dB(T_{ig})}{dT_{ig}}\sigma_{a,ig} + B\left(T_{ig}\right)\frac{d\sigma_{a,ig}}{dT_{ig}} + \\ \sum_{b=1}^{B_g}\frac{dJ_{igb}}{dT_{ig}}\alpha_{igb}\sigma_{a,igb} + J_{igb}\left(\frac{d\alpha_{igb}}{dT_{ig}}\sigma_{a,igb} + \frac{\sigma_{a,igb}}{dT_{ig}}\alpha_{igb}\right)\end{array}\right]u_{ig}\end{aligned}$$

2.1.4 Derivatives wrt pressure parameters

The derivative of the spectrum with respect to pressure parameters is:

$$\frac{dS}{dq_{p,m}} = \sum_i\left[\frac{dS}{dp_{Air,i}}\frac{dp_{Air,i}}{dq_{p,m}} + \sum_g \frac{dS}{du_{ig}}\frac{du_{ig}}{dq_{p,m}}\right]. \quad (19)$$

The first term of this equation reads:

$$\frac{dS}{dp_{Air,i}} = -\tau_i\frac{dS}{d\tau_i}\frac{d\delta_i}{dp_{Air,i}} \quad (20)$$

with:

$$\frac{d\delta_i}{dp_{Air,i}} \approx \sum_{g=1}^{G}\left[\frac{d\sigma_{a,ig}}{dp_{ig}} + \sum_b \alpha igb\frac{d\sigma_{a,ig}}{dp_{ig}}\right]u_{ig} \quad (21)$$

The derivatives of the cross-sections $\sigma_{a,ig}$ with respect to pressure are determined during the calculation of the cross-sections. This is done numerically by recalculating the cross-sections for each path with slightly changed pressures.
The second term in 19, i.e. the dependence of partial columns on pressure parameters is determined during the calculation of integrated path values during ray-tracing. Mind that the partial columns do not depend linearly on the pressure due to refraction, especially at low tangent altitudes.

2.1.5 Derivatives wrt non-LTE/LTE population ratio parameters

For the determination of derivatives wrt non-LTE/LTE population ratios r we must calculate $\frac{d\tilde{J}_i}{dq_{r,m}}$ and $\frac{d\tau_i}{dq_{r,m}}$:

$$\frac{d\tau_i}{dq_{r,m}} = -\tau_i\frac{d\delta_i}{dr_{ign}}\frac{dr_{ign}}{dq_{r,m}} \quad (22)$$

with the derivative of the optical depth wrt non-LTE/LTE population ratio of a distinct state n of gas g:

$$\frac{d\delta_i}{dr_{ign}} = u_{ig}\sum_{b=1}^{B_g}\frac{d\alpha_{igb}}{dr_{ign}}\sigma_{a,igb} \quad (23)$$

The source function derivative wrt vibrational temperature is:

$$\frac{d\tilde{J}_i}{dq_{r,m}} = \frac{\delta_i \frac{d\hat{J}_i}{dT_{Vib,i}} + \hat{J}_i \frac{d\delta_i}{dr_{ign}}}{\delta_i^2} \frac{dr_{ign}}{dq_{r,m}} \tag{24}$$

with:

$$\frac{d\hat{J}_i}{dr_{ign}} = u_{ig} \sum_{b=1}^{B_g} \left(\frac{dJ_{igb}}{dr_{ign}} \alpha_{igb} + J_{igb} \frac{d\alpha_{igb}}{dr_{ign}} \right) \sigma_{a,igb} \tag{25}$$

2.2 The implementation in the radiative transfer module

In this section it is described how the formulas above are implemented in the module radtra_m. We use the variable names of the code and whenever possible a reference to the variable names in section 2.1 is made. The basic steps are:
a) go from the layer far from the observer to the layer near the observer and calculate the radiative transfer and some derivative auxiliary variables.
b) go back layer for layer from the observer to the end of the atmosphere and multiply the auxiliary variables by the transmission between the actual layer and the observer in order to get the radiance derivatives wrt each layer
c) add together the derivatives wrt each layer weighted by the influence of the retrieval parameter in each layer (post-derivation)

And now in detail:
a)
Begin with layer far from observer
For actual layer i calculate: tau = τ_i, src = $\tilde{J}_i$, opt = δ_i , srcn = $\hat{J}_i$, dopt_dcol = dopt_dcol = $\frac{d\delta_i}{du_{ig}}$, dsrc_daerabsopt = $\frac{d\tilde{J}_i}{d\delta_{a,Aerosol,i}}$, dsrc_dTvib = $\frac{d\tilde{J}_i}{dT_{Vib,i}}$, dopt_dTvib = $\frac{d\delta_i}{dT_{Vib,i}}$.
The derivatives wrt kinetic temperature are stored in the 'imaginary' parts of the variable src and opt: derivT(src) = $\frac{d\tilde{J}_i}{dT_{Air,i}}$, derivT(opt) = $\frac{d\delta_i}{dT_{Air,i}}$.
Store tau in variable tausave$_i$ for each layer
Calculate: derirad = tau*(rad-src) = $\tau_i \left[S_{i-1} - \tilde{J}_i \right]$ (see 7)
Calculate actual radiance at end of layer i: rad = derirad + src = $\tau_i \left[S_{i-1} - \tilde{J}_i \right] + \tilde{J}_i$
Calculate derivative auxiliary variables:
derivmr$_i$ = derirad * dopt_dcol
deriaer$_i$ = (1-tau) * dsrc_daerabsopt-derirad
deripla$_i$ = (1-tau) * derivT(src) - derirad * derivT(opt)
derisrc$_i$ = (1-tau) * dsrc_dTvib - derirad * dopt_dTvib

b)
go back layer for layer from the observer to the end of the atmosphere and multiply the auxiliary variables by the transmission between the actual layer and the observer in order to get the radiance derivatives wrt each layer.

Go back from i=N to i=1:
tautot = tausavei * tautot
derivmr$_{i+1}$ = derivmr$_{i+1}$ * tautot
deriaer$_{i+1}$ = deriaer$_{i+1}$ * tautot
deripla$_{i+1}$ = deripla$_{i+1}$ * tautot

$\text{derisrc}_{i+1} = \text{derisrc}_{i+1} * \text{tautot}$

c)
add together the derivatives wrt each layer weighted by the influence of the retrieval parameter in each layer (post-derivation).

This is performed in subroutines:

derivmr: derivmr_calc@radtra_m, derivmrgrad_calc@radtra_m
deriaer: deriaer_calc@radtra_m
deripla: deriT_calc@radtra_m, deriTgrad_calc@radtra_m
derisrc: deriTvib_calc@radtra_m

2.3 Derivatives of path values wrt parameters during ray-tracing

To be able to perform the 'post-derivation', i.e. the step from the derivatives of the spectrum wrt the integrated path variables (partial columns, Curtis-Godson means) to the derivatives wrt the retrieval parameters the following quantities have to be known:

$\frac{du_{ig}}{dq_{vmr,mg}}$ derivative of partial column amount of gas g and layer i with respect to the m^{th} vmr parameter of gas g
Variable: geo()%par()%lay()%speci()%dcol()

$\frac{du_{ig}}{dq_{vmrgradient,mg}}$ derivative of partial column amount of gas g and layer i with respect to the m^{th} vmr-gradient parameter of gas g
Variable: geo()%par()%lay()%speci()%dcolgrad()

$\frac{d\delta_{a,Aerosol,i}}{dq_{Aerosol,m}}$ derivative of aerosol absorption optical depth of layer i with respect to the m^{th} aerosol absorption coefficient parameter
Variable: geo()%par()%lay()%dabsopt()

$\frac{dT_{Air,i}}{dq_{T,m}}$ derivative of Curtis-Godson-temperature of air for layer i with respect to the m^{th} T-parameter
Variable: geo()%par()%lay()%dT()

$\frac{dT_{Air,i}}{dq_{Tgradient,m}}$ derivative of Curtis-Godson-temperature of air for layer i with respect to the m^{th} T-gradient-parameter
Variable: geo()%par()%lay()%dTgrad()

$\frac{dr_{ign}}{dq_{r,m}}$ derivative of non-LTE/LTE population ratio for layer i with respect to the m^{th} non-LTE/LTE population ratio parameter
Variable: geo()%par()%lay()%speci()%iso()%state()
%dTvib_cg_dT()

$\frac{dp_{Air,i}}{dq_{p,m}}$ derivative of Curtis-Godson-pressure of air for layer i with respect to the m^{th} pressure-parameter
Variable: geo()%par()%lay()%dp()

$\frac{du_{i,g}}{dq_{p,m}}$ derivative of partial column amount of gas g and layer i with respect to the m^{th} pressure-parameter
Variable: geo()%par()%lay()%dcol()

Nearly all of these quantities are computed in integrate@ray_m where also the partial column amounts and Curtis-Godson values themselves are determined for each layer.

The procedure is simply numerical:

- change the parameter q using some increment
- recalculate the partial column, Curtis-Godson value, ...
- calculate the numeric derivative using the previously determined partial column, Curtis-Godson value for the undisturbed case

Especially in limb-observations for the derivatives of the layer pressure and column amounts with respect to the pressure parameter it is not possible to do the same procedure as just described layer by layer. Because of the dependence of refraction on pressure it is rather necessary to recalculate the whole limb path with a slightly changed pressure parameter (this has to be done for each parameter). The derivatives are then determined numerically.

3 Calculation of derivatives wrt instrumental parameters

(This is also described in more detail in Part XII: 'FOV and AILS')

- line-of-sight
  Derivatives wrt the nadir angle are determined numerically in the subroutines fovils1@ilsfov_m and envfovils@ilsfov_m: additionally to the field-of-view weighted spectrum around the nominal nadir angle another one is calculated around the nominal nadir angle plus an increment. From these two the fov derivative spectrum is determined.

- offset
  The offset derivative (which is obviously =1) is set in off_deri@offsca_m.

- scale
  The scale derivative (which is obviously equal to the spectrum itself) is set in sca_deri@offsca_m.

- wavenumber shift
  The derivative of the spectrum wrt wavenumber shift is calculated numerically in derive@ilsfov_m: the derivative in the fine-grid spectrum S is:

$$\frac{dS_k}{d\nu} = \frac{S_{k+1} - S_{k-1}}{2\Delta\nu_f}$$

  $\Delta\nu_f$ is the wavenumber fine grid distance. After that the wavenumber shift derivative on the fine grid is convolved with the ails-function.

- ils-parameters
  In the case of circular aperture the ils parameters wrt which the derivatives are determined are linear apodization and phase. These derivatives are calculated numerically in the routines ilsapo@ilsfov_m and fovils1@ilsfov_m. For this purpose the ils is recomputed with slightly modified linear apodization or phase parameters to get the derivative of the ils wrt these parameters. This ils-derivative is then convolved with the fine-grid spectrum. The same procedure is implemented for the ils parameters defined by ESA.

4 Principles of derivative and retrieval parameter handling

One main feature of KOPRA is the flexible handling of abstract retrieval parameters in order to support different retrieval strategies. Therefore, in the core of KOPRA input profile data are strictly separated from parameters used for the description of atmospheric profiles.
Derivatives are only computed with respect to parameters. I.e. if derivatives of the spectrum wrt some atmospheric quantity (e.g. the temperature) should be calculated, the profile of this quantity must be parameterized (e.g. determination of temperature parameters). On the other hand, a profile can be parameterized without derivatives being calculated (i.e. the parameters are only used for the forward calculation of the spectrum). This procedure implies that at some point in the program a transformation from the atmospheric profile to the parameter space and somewhere else a back-transformation from the parameters to the value of the atmospheric quantity is made.
The transformation into the parameter space (i.e. the initialization of the parameter vectors) is done outside the core of the forward model (in the module `inipar_m`) following the reading of input files. Here, the input profiles are transformed into parameters using some formula which has to be explicitly implemented in `inipar_m`. The parameter values are stored in the variable `para%` while the input profiles have already been read into `inprof%`.
The back-transformation is calculated during the ray-tracing and path integration (module `ray_m`) where the explicit value of the quantity at some point in the atmosphere is needed. This task is done by the `give`-functions of module `give_m`. These routines return the value of any atmospheric quantity at any position in the atmosphere. Hence, they not only perform the back-transformation from the parameters but also the interpolation of the input profiles and the distinction between parameters (using variable `para%`) and input profiles (using variable `inprof%`). Obviously, changing the transformation formulas in `inipar_m` requires also a modification of the back-transformation in `give_m` and vice versa.

5 Retrieval interface

5.1 Input from the retrieval-code

- *Which derivative should be calculated?* In $ 10 of the main KOPRA input file it has to be defined which derivatives of the spectrum should be calculated by KOPRA.

- *Atmospheric retrieval parameters:* as explained above, the atmospheric profile parameters with respect to which derivatives are determined are handled differently from the rest of the profiles. The retrieval profile parameters are stored in the variable `para%`[1] and are initialized in module `inipar_m`. In the basic version of KOPRA the profile parameters are set identical to the values at the levels given in the main KOPRA input file under $11.5 and the interpolation rule is linear in altitude and only for pressure linear in ln(altitude). For different parameterizations the user has to adjust the module `inipar_m` for initialization and the module `give_m` for the transformation from the parameter space into profile-altitude, -latitude, -longitude space (the interpolation rules).

[1] For details about the variables see Part XIX: 'Module, subroutine and variable listing and description'

The interface to the retrieval is performed through the variable `para%` by use of the module `inpdat_m`.

- *Instrumental retrieval parameters:* for initialization of the instrumental retrieval parameters the values given in the main KOPRA input file are used and for these parameters the derivatives are determined. I.e. no re-parameterization like for the atmospheric retrieval parameters is made. The interface to the retrieval is performed through variable `inst%` by use of the module `inpdat_m`.
- *Information on necessary allocations and recalculations:* for some variables during each iteration the retrieval code has to tell the forward model which ones have to be allocated and which ones should be recalculated. The recalculation concerns the leveling of the atmosphere, the determination of additional geometries for the simulation of the field-of-view, and the calculation of absorption coefficients. The relating control parameters are `sw%new_modelgeo`, `sw%new_modelgrid`, `sw%new_absco`.

5.2 Output to the retrieval-code

- *Spectra and derivatives:* all spectra and their derivatives with respect to the retrieval parameters are available through variable `outdat%` by use of module `outdat_m`.

5.3 Example for use of KOPRA in a retrieval environment

The call of KOPRA inside a retrieval program and the setting of the variables for allocation and recalculation is given in the example below.

```
! read main KOPRA input-file
  call input('input/kopra.inp')

! read retrieval input-file (belongs to the retrieval code)
  call input_invers('input/koprainv.inp')

! initialize the atmospheric retrieval parameters (module inipar_m)
  call ini_para

! for KOPRA run in the first iteration
  sw%firstrun=.true.

! variables geo%, sim%, modprof%, Sails%, deri%, mw% are not allocated
  sw%alloc_geo = 0
  sw%alloc_sim = 0
  sw%alloc_modprof = 0
  sw%alloc_Sails = 0
  sw%alloc_outdat = 0
  sw%alloc_deri = 0
  sw%alloc_mw = 0

! additional geometries for fov will be added
  sw%new_modelgeo = .true.

! the atmospheric layering will be done
```

```
  sw%new_modelgrid = .true.

! the absorption coefficients will be calculated
  sw%new_absco = .true.

! call KOPRA - first iteration
  call kopra_forwrd

! call the inversion algorithm - first iteration
  call kopra_invers

! for KOPRA run in the next iterations
  sw%firstrun=.false.

! loop on maxiter iterations
  do i = 2, maxiter

   ! additional geometries for fov will not be determined new
     sw%new_modelgeo = .false.

   ! the atmospheric layering will not be done new
     sw%new_modelgrid = .false.

   ! the absorption coefficients will be calculated new for each iteration
     sw%new_absco = .true.

   ! deallocation of various variables if necessary
     if ( sw%alloc_geo/=0 .and.&
          (sw%firstrun.or.sw%new_absco)  ) then
        call deallocate_geo(0)
        sw%alloc_geo = 0
     end if
     if ( sw%alloc_Sails /= 0 ) then
        call deallocate_Sails(0)
        sw%alloc_Sails = 0
     end if
     if (sw%alloc_deri/=0 .and. sw%firstrun ) then
        call deallocate_deri(0)
        sw%alloc_deri = 0
     end if
     if (sw%alloc_mw/=0 .and. sw%firstrun ) then
        call deallocate_mw(0)
        sw%alloc_mw = 0
     end if

   ! call KOPRA - next iterations
     call kopra_forwrd

   ! call the inversion algorithm - next iterations
     call kopra_invers
  end do
```

Optimization of model accuracy parameters

M. Höpfner and S. Kellmann

Abstract: For the discretization of the atmospheric radiative transfer problem a variety of parameters are necessary which determine the performance of the program. An adjustment of these parameters taking into account the trade-off between accuracy and run-time is required for routine applications of the processor. We present the model errors for all spectral bands of MIPAS-Envisat for different parameter settings compared to the expected noise of the measurements.

1 Introduction

A variety of parameters are necessary for the discretization of the atmospheric radiative transfer problem. For routine applications of the processor these parameters have to be adjusted by choosing a trade-off between accuracy and run-time. These parameters, which can be modified in \$6 and \$7 of the main Kopra input file, are:

- Ray-tracing step length (\$7.11)
- Finest spectral grid (\$6.1)
- Accuracy for cross-section calculation (\$7.2)
- Width of the AILS function (\$7.8)
- Number of cross-section recalculations for limb-scans (\$7.7)
- Additional ray-paths for field-of-view (\$7.9-7.10)
- Atmospheric layering (\$7.3-7.6)
- Temperature for calculation of the Planck-function (\$7.12)

We performed calculations over the entire spectral range of MIPAS-Envisat for different settings of each parameter and compared these to 'reference' cases where the parameters were set to very high accuracy. To give an impression on how these settings influence the total run-time of KOPRA a test was performed with a microwindow selection for ozone retrieval. It must be kept in mind, that these run-times are strictly valid only for the probed microwindows. However, the example should be sufficient to estimate the relative importance of the parameter settings (table 6). In the following we discuss the optimizations one by one. In tables 1 to 5 the mean, the standard deviation and the maximum relative error with respect to the MIPAS-Envisat noise (NESR, Noise Equivalent Signal Radiance) values are given for different tangent altitudes.

2 Ray-tracing step length

For modeling the atmospheric ray path and calculating partial column amounts and Curtis-Godson (C.G.) values of pressure and temperature the line of sight is subdivided into straight segments of equal length. The effect of the step length is twofold:

(a) Determining the accuracy of the refracted ray-path

(b) Determining the accuracy of the partial column and C.G. integrals and their derivatives with respect to retrieval parameters

Reference spectra were calculated with a step length of 0.1km. The main effect of enlarging this value is recognizable at the lowest tangent altitude. The errors are even for 10km steps below 10% of the NESR. For 1km the errors are below 1%. Therefore, step lengths between 1km and 5km are recommended for MIPAS-Envisat retrievals.

3 Finest spectral grid

Absorption cross-sections are calculated line-by-line on an irregular spectral grid which is an integer multiple of the finest spectral grid. Hence, the finest spectral grid is the minimum possible distance between two wavenumber grid points. If it is chosen too large, lines will not be sampled with a sufficient number of points. Due to this effect the error should increase with tangent altitude due to the narrower lines in the Doppler region. This can slightly be seen in band A. However, in the other bands this effect is not visible. This can be ascribed to the fact that at lower altitudes the radiance spectra are stronger and that in the tables the absolute errors with respect to the MIPAS noise are shown and no relative errors.
Reference spectra for the tables are made with a grid interval of 0.0001cm^{-1}. The coarsest grid tested was 0.001cm^{-1}. Also in this case the standard deviation of the error distribution is well below 10% of the NESR. However, for some tangent altitudes and wavenumbers the maximum error over the whole band can reach the NESR. Hence, we suggest to use settings between 0.0005cm^{-1} and 0.0008cm^{-1}.

4 Accuracy for cross-section calculation

The line-by-line calculation of the absorption cross-sections is controlled by an accuracy parameter being the maximum error in optical depth allowed for each line when interpolating the irregular frequency grid to the finest spectral grid. From the tables it is evident that the largest errors occur at low tangent altitudes due to the strong overlap of many lines, a fact which is not considered by the accuracy coefficient since it is applied line by line.
The reference value for this accuracy coefficient has been 10^{-12}. A value of about 10^{-7} is sufficient to keep the errors for all cases well below 10% of the NESR.

5 Width of the AILS function

The apodized instrumental line shape (AILS) function is convolved with the monochromatic spectrum after radiative transfer. By this procedure information of transitions from outside the spectral region of interest is convolved into the range. In order to minimize the effect a Norton-Beer strong apodization was chosen. It is necessary to test how far the wings of this function have to be calculated. The width of the AILS in- fluences the run-time twofold:

(a) spectral microwindows have to be extended by the width of the AILS - hence, more time for the monochromatic calculation is needed

(b) the convolution time of the monochromatic spectrum with the AILS is increased

For simulations with many spectral microwindows (a) is the most important effect while for broadband calculations (b) is more relevant.
The reference was calculated with an width of the Norton-Beer strong apodization function of 7cm^{-1} ($7.8 = 3) where the apodization function decreases to 0.01% of it's center value. With an width of 0.15cm^{-1} ($7.8 = 1) the standard deviation of the error distribution is about 10% of the NESR, however, maximum errors of about the noise value can appear. With 1.4cm^{-1} width ($7.8 = 2) one is on the save side. However, for a distinct microwindow selection one should check if it is not possible to use 0.15cm^{-1} ($7.8 = 1) since the time saving can be substantial.

6 Number of cross-section recalculations for limb-scans

An entire limb-scan (16 tangent altitudes in the standard MIPAS mode) is calculated during one model-run. For minimization of cross-section determination cross-sections for all layers of the lowest line-of-sight are computed. For higher tangent altitudes cross-sections from the related layers of the lowest tangent path can be used since the Curtis-Godson p and T values vary only slightly[2] due to the bended layers of the atmosphere. Therefore, the differences in C.G. values of each path increase when approaching the tangent layer. With respect to a reference calculation where the cross-sections for all paths are calculated, it was tested how many 'extra' paths have to be recalculated for the geometries above the lowest one.
The result was that at least the tangent layers must be recalculated ($7.7 = 1). With this option the error standard deviation is around 4% of the NESR. A recalculation of the lowest two layers is recommended for more exact calculation ($7.7 = 2). A further enhancement of the recalculated layers seems to be unnecessary.

7 Additional ray-paths for field-of-view

For simulation of the finite field-of-view (FOV) of the instrument it is necessary to calculate additional ray-paths through the atmosphere. The FOV simulation is especially crucial at low tangent heights where the radiance profiles have strong gradients with respect to altitude. Therefore, KOPRA has the possibility to separate two altitude regions where the FOV simulation is performed with more or less additionally simulated ray-paths.
Reference calculations were performed with 13 geometries (6 on each side of the central ray) over the total FOV. In order to achieve standard deviations of the error of less than 10% above 14km tangent altitude one additional line of sight ($7.10 = -1) between the standard altitudes (with 3km spacing) is sufficient. With this setting maximum band errors of about 50% of the NESR can occur. Below 14km tangent altitude 2 additional ($7.10 = -2) geometries are necessary. At very low altitudes (8km) this even may not be sufficient and 3 ($7.10 = -3) could be necessary. Since the number of additional ray paths is an important factor determining the run-time it is surely worth to adjust it for used microwindow sets.

8 Atmospheric layering

The subdivision of the atmosphere into layers inside KOPRA determines the accuracy of the radiative transfer calculation. This is an important issue since the run-time of the program is nearly proportional to the number of layers.
Following options have been tested:

(a) US76 standard atmosphere layers
46 levels, with criteria:
$7.31 = 2
$7.32 = 46 levels: 1km up to 25km altitude, 2.5km up to 50km altitude, 5km up to 100km altitude

(b) automatic layering
75 levels with criteria:
$7.31 = 4

[2]The Curtis-Godson values would be exactly equal for plane-parallel layers.

$7.3 = 2, 5, 40 (ΔT 2K below 40km, 5K above)
$7.4 = 1.4 (maximum variation of the Lorentz line width between two model levels)
$7.5 = 100 2 (not relevant)
$7.6 = 0.4 (minimum layer thickness)

(c) automatic layering
66 levels with criteria:
$7.31 = 4
$7.3 = 2, 5, 25 (ΔT 2K below 25km, 5K above)
$7.4 = 1.4 (maximum variation of the Lorentz line width between two model levels)
$7.5 = 100 2 (not relevant)
$7.6 = 0.4 (minimum layer thickness)

(d) coarse layers 29 levels with criteria:
$7.31 = 2
$7.32 = 29 levels: 3km from 8 to 51km altitude, 5km above

(e) coarse layers and automatic layering
77 levels with criteria:
$7.31 = 3
$7.32 = 29 levels: 3km from 8 to 51km altitude, 5km above $7.3 = 2, 5, 40 (ΔT 2K below 40km, 5K above)
$7.4 = 1.4 (maximum variation of the Lorentz line width between two model levels)
$7.5 = 100 2 (not relevant)
$7.6 = 0.4 (minimum layer thickness)

(f) reference calculation
156 levels with criteria:
$7.31 = 2
$7.32 = 156 levels: 0.5km distance up to 75km altitude, 1km distance above

Below about 30km altitude the coarse layering with 3km width is not sufficient to reach error standard deviations of 10% of the noise. The 46 US76 standard levels reach this value down to 11km except for band AB. Below 11km (i.e. in the troposphere) a finer layering is needed, for example like in the case of 66 total levels.

9 Temperature for calculation of the Planck-function

In $7.12 of the main KOPRA input file one can decide which temperature is used for the Planck function of each atmospheric path[3]. For $7.12 = 0 the Planck functions with Curtis-Godson temperatures for each molecule are mixed and weighted by their optical depth at each wavenumber grid point. This option is used for reference calculation. For microwindow selections under $7.12 the number of the main absorbing molecule of these spectral regions can be inserted. $7.12 = -1 means that the Curtis-Godson temperatures of the different molecules will not be mixed but that just the Curtis-Godson temperature of total air is used. In this case the model errors are acceptable above the tropopause. For tangent altitudes in the troposphere it is recommended to perform tests for the specific microwindows in combination with a finer atmospheric layering.

[3]Paths are the segments cut by the internal layering out of the line-of-sight.

Band A: 685 – 970 cm^{-1}					
Mean, **standard deviation**, *maximum* relativ error in % with respect to MIPAS NESR for tangent altitudes					
		8 km	23 km	38 km	53 km
Ray–tracing	1 km	–0.210, **0.175**, *0.917*	0.006, **0.010**, *0.065*	0.002, **0.005**, *0.046*	–0.0002, **0.0009**,*0.026*
step length	10 km	–3.71, **2.77**, *11.77*	0.42, **0.78**, *5.19*	0.17, **0.47**, *4.01*	–0.02, **0.08**, *2.36*
Finest	0.0005 cm^{-1}	–0.20, **0.73**, *19.59*	–0.02, **0.89**, *16.87*	0.03, **0.14**, *1.22*	–0.02, **0.31**, *5.96*
spectral	0.0008 cm^{-1}	–0.32, **0.90**, *10.97*	–0.08, **1.13**, *22.09*	–0.04, **1.43**, *15.93*	–0.09, **3.16**, *53.80*
grid	0.001 cm^{-1}	–0.42, **2.19**, *37.76*	–0.10, **2.22**, *36.94*	0.06, **4.05**, *40.69*	0.10, **7.35**, *133.11*
Accuracy for	10^{-8}	–0.90, **0.79**, *2.97*	–0.26, **0.15**, *0.66*	–0.05, **0.03**, *0.21*	–0.01, **0.01**, *0.04*
cross–section	10^{-6}	–11.95, **13.02**, *53.85*	–3.26, **2.40**, *16.87*	–0.81, **0.64**, *3.92*	–0.13, **0.14**, *0.82*
calculation	10^{-4}	–195, **266**, *2353*	–51.51, **40.97**, *447*	–13.02, **11.58**, *76.23*	–2.17, **2059**, *21.18*
Width of	1.40 cm^{-1}	–2.71, **2.97**, *31.35*	–1.12, **4.90**, *25.42*	–0.42, **4.75**, *29.88*	–0.08, **1.98**, *13.22*
AILS function	0.15 cm^{-1}	7.91, **7.61**, *45.86*	3.29, **12.81**, *92.78*	1.20, **11.56**, *75.08*	0.23, **4.89**, *62.41*
Number of	3	–0.57, **0.65**, *4.99*	–0.83, **0.89**, *4.41*	–0.26, **0.41**, *3.21*	0.01, **0.06**, *1.49*
cross–section	2	–0.83, **0.98**, *7.57*	–2.28, **2.86**, *14.13*	–0.45, **0.84**, *6.76*	–0.01, **0.13**, *3.71*
recalculations	1	–1.81, **2.09**, *16.42*	–3.50, **4.09**, *19.90*	–1.13, **2.60**, *19.68*	–0.62, **1.85**, *26.73*
for limb–scans	0	–31.08, **48.28**, *300.98*	–9.49, **12.58**, *64.86*	–3.29, **6.83**, *44.05*	6.42, **26.97**, *408.11*
Additional	2	53.24, **52.41**, *324.54*	0.15, **3.81**, *22.79*	0.29, **1.73**, *21.30*	0.40, **0.96**, *12.30*
ray–paths	1	116.09, **104.05**, *464.04*	–0.36, **8.23**, *53.07*	0.61, **4.12**, *52.42*	0.92, **2.22**, *28.77*
for	0	687.4, **636.0**, *3238.5*	–2.53, **39.08**, *230.35*	3.11, **21.91**, *236.73*	4.49, **11.60**, *146.35*
field–of–view					
Atmospheric	a (46 levels)	–0.23, **18.44**, *174.84*	1.82, **9.49**, *80.95*	0.57, **5.98**, *94.85*	0.06, **1.96**, *47.60*
layering	b (75 levels)	5.37, **7.18**, *44.67*	2.65, **4.09**, *24.57*	0.76, **1.66**, *22.55*	0.19, **0.39**, *8.11*
	c (66 levels)	4.90, **9.28**, *46.56*	2.22, **9.98**, *61.67*	1.02, **2.54**, *40.27*	0.19, **0.39**, *8.11*
	d (29 levels)	–67.65, **197.82**, *1602.4*	1.36, **14.71**, *116.88*	0.33, **6.60**, *100.27*	0.08, **1.80**, *36.95*
	e (77 levels)	4.77, **7.48**, *43.72*	3.01, **4.26**, *25.28*	0.82, **1.83**, *28.07*	0.22, **0.43**, *8.73*
Gas/isotope number of the main gas	–1	–5.00, **11.43**, *119.13*	1.73, **1.95**, *10.45*	–0.07, **0.56**, *3.33*	–0.25, **0.67**, *8.87*

Table 1: Mean-value, **standard deviation**, and *maximum* of the relative errors (%) with respect to MIPAS NESR in channel A for various model parameter settings and tangent altitudes.

Band AB: 1020 – 1170 cm^{-1}					
Mean, **standard deviation**, *maximum* relativ error in % with respect to MIPAS NESR for tangent altitudes					
		8 km	23 km	38 km	53 km
Ray–tracing	1 km	–0.010, **0.089**, *0.481*	0.016, **0.030**, *0.147*	0.008, **0.015**, *0.091*	0.0004, **0.0012**, *0.0106*
step length	10 km	–2.11, **1.82**, *6.73*	1.25, **2.09**, *11.85*	0.64, **1.28**, *7.70*	0.04, **0.12**, *1.02*
Finest	0.0005 cm^{-1}	–0.30, **1.12**, *7.95*	–0.25, **0.34**, *2.05*	–0.36, **0.38**, *1.88*	–0.004, **0.081**, *0.687*
spectral	0.0008 cm^{-1}	–0.45, **0.54**, *10.46*	–0.35, **0.37**, *3.37*	–0.29, **0.56**, *3.83*	–0.01, **2.26**, *24.32*
grid	0.001 cm^{-1}	–0.61, **1.39**, *9.19*	–0.47, **0.92**, *10.47*	–0.53, **1.78**, *13.50*	0.01, **6.47**, *62.57*
Accuracy for	10^{-8}	–0.35, **0.31**, *1.70*	–0.20, **0.12**, *0.66*	–0.22, **0.15**, *0.74*	–0.03, **0.03**, *0.10*
cross–section	10^{-6}	–6.54, **6.37**, *34.79*	–2.96, **1.70**, *12.61*	–2.98, **2.46**, *10.96*	–0.36, **0.35**, *1.37*
calculation	10^{-4}	–95.30, **98.97**, *869.46*	–41.47, **26.70**, *219.95*	–38.53, **30.24**, *195.59*	–5.81, **4.59**, *47.61*
Width of	1.40 cm^{-1}	–2.84, **5.19**, *30.13*	–2.08, **6.67**, *29.95*	–1.28, **9.46**, *47.46*	–0.23, **4.00**, *23.96*
AILS function	0.15 cm^{-1}	8.12, **12.77**, *72.28*	6.02, **16.79**, *74.29*	3.83, **22.94**, *120.62*	0.69, **9.88**, *55.50*
Number of	3	–0.45, **0.40**, *2.45*	–1.83, **1.23**, *5.07*	–0.62, **0.68**, *2.92*	–0.01, **0.02**, *0.16*
cross–section	2	–0.62, **0.59**, *3.34*	–3.71, **1.96**, *11.72*	–1.35, **1.72**, *7.72*	–0.06, **0.08**, *0.58*
recalculations	1	–1.12, **1.12**, *5.98*	–4.93, **2.65**, *16.35*	–4.29, **6.13**, *25.84*	–0.83, **1.19**, *6.82*
for limb–scans	0	–11.50, **16.34**, *121.04*	–10.62, **7.45**, *43.88*	–11.66, **14.05**, *66.14*	7.62, **14.22**, *97.34*
Additional	2	21.99, **22.84**, *170.32*	–0.12, **2.76**, *14.58*	–1.68, **4.38**, *30.17*	1.27, **1.54**, *10.72*
ray–paths	1	47.96, **44.87**, *236.85*	–0.29, **5.59**, *32.99*	–3.94, **10.34**, *72.08*	2.89, **3.50**, *24.59*
for	0	270.7, **266.8**, *1637.6*	–1.01, **26.84**, *147.22*	–18.76, **49.99**, *325.67*	14.26, **17.22**, *119.03*
field–of–view					
Atmospheric	a (46 levels)	–6.91, **15.71**, *101.77*	–10.81, **24.14**, *124.22*	–2.13, **11.65**, *110.93*	7.08, **17.94**, *139.82*
layering	b (75 levels)	1.22, **6.88**, *24.22*	–0.83, **7.21**, *27.36*	1.25, **3.72**, *27.70*	1.44, **2.38**, *16.58*
	c (66 levels)	–3.54, **16.07**, *63.06*	–9.70, **24.57**, *78.40*	1.16, **5.93**, *52.72*	1.44, **2.38**, *16.58*
	d (29 levels)	–44.28, **78.43**, *799.22*	–21.52, **44.41**, *202.84*	–6.62, **17.81**, *162.71*	6.09, **15.90**, *124.82*
	e (77 levels)	1.14, **6.76**, *24.27*	–0.67, **7.34**, *27.30*	1.10, **4.33**, *37.09*	1.41, **2.28**, *15.87*
Gas/isotope	–1	–2.35, **10.77**, *77.16*	–6.65, **7.00**, *43.67*	–0.19, **3.42**, *31.95*	–6.58, **11.75**, *79.35*
number of the					
main gas					

Table 2: Mean-value, **standard deviation**, and *maximum* of the relative errors (%) with respect to MIPAS NESR in channel AB for various model parameter settings and tangent altitudes.

Band B: 1215 – 1500 cm^{-1}					
Mean, **standard deviation**, *maximum* relativ error in % with respect to MIPAS NESR for tangent altitudes					
		8 km	23 km	38 km	53 km
Ray–tracing step length	1 km	–0.007, **0.094**, *0.527*	0.002, **0.004**, *0.045*	0.0002, **0.0015**,*0.0267*	–0.00003, **0.00018**,*0.00538*
	10 km	–1.00, **1.72**, *7.13*	0.09, **0.30**, *3.64*	0.01, **0.13**, *2.34*	–0.003, **0.017**, *0.466*
Finest spectral grid	0.0005 cm^{-1}	–0.18, **0.35**, *5.56*	0.003, **1.075**, *31.122*	–0.008, **0.055**, *0.603*	–0.003, **0.018**, *0.327*
	0.0008 cm^{-1}	–0.27, **0.16**, *1.39*	–0.07, **0.94**, *24.27*	–0.02, **0.08**, *0.92*	–0.004, **0.031**, *0.509*
	0.001 cm^{-1}	–0.36, **0.38**, *5.75*	–0.04, **1.07**, *30.90*	–0.02, **0.10**, *1.19*	–0.01, **0.04**, *0.95*
Accuracy for cross–section calculation	10^{-8}	–0.01, **1.27**, *0.96*	–0.08, **0.07**, *0.48*	–0.02, **0.02**, *0.07*	–0.002, **0.001**, *0.007*
	10^{-6}	0.35, **3.35**, *26.63*	–0.96, **6.33**, *36.25*	–0.36, **4.68**, *48.11*	–0.04, **1.59**, *17.51*
	10^{-4}	–4.16, **45.61**, *342.78*	–16.82, **15.19**, *77.12*	–5.79, **4.22**, *37.47*	–0.77, **0.63**, *5.02*
Width of AILS function	1.40 cm^{-1}	–1.26, **1.50**, *8.42*	–0.30, **2.31**, *26.68*	–0.07, **2.62**, *11.92*	–0.02, **1.06**, *5.36*
	0.15 cm^{-1}	3.83, **3.59**, *25.77*	0.89, **4.99**, *32.20*	0.21, **6.04**, *43.13*	0.05, **2.47**, *17.11*
Number of cross–section recalculations for limb–scans	3	0.06, **0.20**, *1.30*	–0.33, **0.33**, *2.11*	–0.07, **0.09**, *1.37*	0.0003, **0.0173**, *0.5458*
	2	0.18, **0.39**, *2.32*	–1.00, **1.08**, *5.65*	–0.12, **0.19**, *3.19*	–0.002, **0.044**, *1.363*
	1	0.21, **0.95**, *4.55*	–1.52, **1.59**, *8.33*	–0.25, **0.62**, *8.58*	–0.06, **0.20**, *3.44*
	0	–10.47, **19.14**, *133.76*	–5.51, **6.08**, *33.99*	–0.84, **1.97**, *21.66*	0.43, **2.40**, *50.39*
Additional ray–paths for field–of–view	2	7.64, **24.23**, *182.76*	0.50, **2.47**, *15.03*	0.21, **0.51**, *4.23*	0.11, **0.34**, *4.21*
	1	2.07, **49.76**, *274.79*	0.94, **5.22**, *33.26*	0.47, **1.16**, *10.00*	0.26, **0.79**, *9.85*
	0	35.7, **269.2**, *1887.8*	4.43, **24.41**, *131.90*	2.13, **7.38**, *72.45*	1.26, **4.28**, *44.83*
Atmospheric layering	a (46 levels)	28.63, **27.42**, *161.11*	–0.14, **6.82**, *80.34*	–0.13, **4.58**, *60.25*	0.27, **2.65**, *69.56*
	b (75 levels)	3.50, **5.12**, *26.33*	0.62, **2.38**, *20.17*	0.33, **1.09**, *14.29*	0.11, **0.32**, *7.99*
	c (66 levels)	3.08, **6.25**, *44.69*	0.12, **5.97**, *65.36*	0.34, **1.56**, *21.09*	0.11, **0.32**, *7.99*
	d (29 levels)	304.9, **254.4**, *1328.1*	–0.86, **11.07**, *122.62*	–0.44, **5.53**, *78.02*	0.22, **2.41**, *61.80*
	e (77 levels)	4.66, **4.27**, *24.10*	0.69, **2.42**, *19.76*	0.32, **1.20**, *16.08*	0.11, **0.30**, *7.58*
Gas/isotope number of the main gas	–1	–14.62, **16.69**, *97.48*	0.30, **2.60**, *27.48*	0.43, **1.74**, *21.00*	–0.24, **0.97**, *12.94*

Table 3: Mean-value, **standard deviation**, and *maximum* of the relative errors (%) with respect to MIPAS NESR in channel B for various model parameter settings and tangent altitudes.

Band C: 1570 – 1750 cm^{-1}					
Mean, **standard deviation**, *maximum* relativ error in % with respect to MIPAS NESR for tangent altitudes					
		8 km	23 km	38 km	53 km
Ray–tracing step length	1 km	–0.008, **0.013**, *0.071*	0.003, **0.010**, *0.154*	0.001, **0.006**, *0.099*	–0.00002, **0.00028**, *0.00497*
	10 km	0.34, **0.47**, *5.22*	0.25, **0.84**, *12.42*	0.09, **0.50**, *8.60*	–0.002, **0.025**, *0.443*
Finest spectral grid	0.0005 cm^{-1}	–0.14, **0.32**, *1.61*	–0.12, **0.80**, *19.32*	–0.12, **0.09**, *0.74*	–0.02, **0.03**, *0.41*
	0.0008 cm^{-1}	–0.23, **0.15**, *0.38*	–0.08, **0.72**, *17.18*	–0.03, **0.09**, *1.03*	–0.01, **0.04**, *0.74*
	0.001 cm^{-1}	–0.29, **0.33**, *1.87*	–0.17, **0.80**, *19.21*	–0.14, **0.13**, *1.38*	–0.02, **0.05**, *0.95*
Accuracy for cross–section calculation	10^{-8}	–0.001, **0.017**, *0.091*	–0.06, **0.03**, *0.14*	–0.03, **0.01**, *0.07*	–0.003, **0.001**, *0.008*
	10^{-6}	0.04, **0.37**, *2.43*	–0.97, **0.48**, *2.51*	–0.58, **0.18**, *1.44*	–0.06, **0.02**, *0.19*
	10^{-4}	0.24 **5.98**, *43.09*	–14.42, **7.05**, *42.43*	–9.70, **4.54**, *33.23*	–1.18, **0.81**, *7.07*
Width of AILS function	1.40 cm^{-1}	–0.92, **1.39**, *6.44*	–0.30, **1.97**, *18.19*	–0.11, **2.20**, *7.90*	–0.02, **1.30**, *5.32*
	0.15 cm^{-1}	2.48, **3.30**, *20.52*	0.77, **4.55**, *21.72*	0.31, **5.63**, *28.92*	0.05, **3.19**, *17.86*
Number of cross–section recalculations for limb–scans	3	0.15, **0.16**, *0.72*	–0.57, **0.50**, *2.75*	–0.16, **0.31**, *5.28*	–0.002, **0.023**, *0.438*
	2	0.29, **0.28**, *1.12*	–1.50, **1.05**, *7.02*	–0.31, **0.72**, *11.78*	–0.0002, **0.053**, *1.042*
	1	0.42, **0.45**, *1.68*	–2.08, **1.42**, *9.99*	–0.77, **2.12**, *29.46*	–0.11, **0.43**, *10.44*
	0	–0.02, **0.26**, *1.50*	–6.91, **4.47**, *32.69*	–1.97, **4.17**, *52.06*	1.23, **7.99**, *210.82*
Additional ray–paths for field–of–view	2	6.51, **6.40**, *18.70*	0.82, **2.10**, *13.03*	–0.07, **0.85**, *12.91*	0.19, **0.59**, *12.55*
	1	–2.73, **2.94**, *10.05*	1.78, **4.68**, *30.00*	–0.18, **2.05**, *31.56*	0.44, **1.35**, *28.83*
	0	–14.29, **15.82**, *81.16*	8.46, **21.79**, *122.86*	–0.82, **10.33**, *151.81*	2.18, **6.78**, *140.97*
Atmospheric layering	a (46 levels)	36.74, **31.35**, *99.48*	–0.03, **4.47**, *70.27*	1.11, **4.88**, *56.25*	1.09, **7.23**, *87.28*
	b (75 levels)	12.77, **8.27**, *33.84*	0.39, **1.80**, *18.16*	0.71, **1.05**, *21.61*	0.22, **0.74**, *8.76*
	c (66 levels)	12.47, **9.37**, *55.07*	–0.50, **6.35**, *79.51*	0.87, **1.64**, *42.89*	0.22, **0.74**, *8.76*
	d (29 levels)	504.0, **398.9**, *1140.7*	–1.22, **8.56**, *123.88*	0.72, **4.59**, *66.78*	0.99, **6.70**, *81.98*
	e (77 levels)	7.48, **5.94**, *20.67*	0.44, **1.84**, *16.46*	0.72, **1.22**, *29.76*	0.22, **0.67**, *9.12*
Gas/isotope number of the main gas	–1	–11.97, **8.93**, *32.94*	–1.36, **2.08**, *27.08*	0.21, **1.87**, *25.14*	–0.32, **1.00**, *15.45*

Table 4: Mean-value, **standard deviation**, and *maximum* of the relative errors (%) with respect to MIPAS NESR in channel C for various model parameter settings and tangent altitudes.

Band D: 1820 – 2410 cm^{-1}					
Mean, **standard deviation**, *maximum* relativ error in % with respect to MIPAS NESR for tangent altitudes					
		8 km	23 km	38 km	53 km
Ray–tracing step length	1 km	–0.030, **0.035**, *0.129*	0.0004, **0.0019**, *0.0365*	0.0005, **0.0016**, *0.0211*	–0.0001, **0.0004**, *0.0052*
	10 km	–0.26, **0.37**, *1.88*	0.03, **0.15**, *3.06*	0.04, **0.14**, *1.86*	–0.01, **0.03**, *0.46*
Finest spectral grid	0.0005 cm^{-1}	–0.028, **0.211**, *2.585*	–0.002, **0.014**, *0.234*	–0.001, **0.015**, *0.263*	–0.001, **0.005**, *0.179*
	0.0008 cm^{-1}	–0.024, **0.074**, *0.397*	–0.004, **0.019**, *0.398*	–0.002, **0.018**, *0.409*	–0.0003, **0.008**, *0.315*
	0.001 cm^{-1}	–0.047, **0.215**, *2.602*	–0.005, **0.022**, *0.516*	–0.003, **0.024**, *0.552*	–0.002, **0.013**, *0.406*
Accuracy for cross–section calculation	10^{-8}	–0.004, **0.011**, *0.083*	–0.010, **0.008**, *0.051*	–0.010, **0.008**, *0.046*	–0.001, **0.001**, *0.004*
	10^{-6}	–0.052, **0.187**, *1.324*	–0.142, **0.105**, *0.829*	–0.130, **0.098**, *0.566*	–0.017, **0.010**, *0.070*
	10^{-4}	–1.01, **2.84**, *22.00*	–2.38, **1.90**, *26.53*	–2.43, **1.89**, *35.00*	–0.42, **0.34**, *7.77*
Width of AILS function	1.40 cm^{-1}	–0.16, **0.47**, *3.61*	–0.04, **0.47**, *2.83*	–0.03, **0.61**, *3.49*	–0.009, **0.308**, *2.340*
	0.15 cm^{-1}	0.24, **0.94**, *8.38*	0.03, **1.07**, *9.03*	0.02, **1.42**, *11.14*	0.004, **0.701**, *7.807*
Number of cross–section recalculations for limb–scans	3	0.005, **0.027**, *0.122*	–0.07, **0.08**, *1.30*	–0.05, **0.11**, *0.93*	0.005, **0.023**, *0.340*
	2	0.02, **0.08**, *0.50*	–0.13, **0.16**, *1.95*	–0.09, **0.18**, *2.06*	0.015, **0.083**, *1.136*
	1	0.02, **0.20**, *1.30*	–0.17, **0.21**, *2.74*	–0.18, **0.36**, *6.20*	–0.05, **0.24**, *2.82*
	0	–3.05, **5.07**, *29.02*	–0.38, **0.56**, *7.97*	–0.27, **0.47**, *9.74*	1.59, **4.32**, *36.65*
Additional ray–paths for field–of–view	2	4.77, **10.08**, *42.13*	0.01, **0.16**, *2.90*	–0.05, **0.16**, *2.42*	0.15, **0.26**, *2.39*
	1	6.09, **16.94**, *58.08*	0.02, **0.33**, *6.52*	–0.11, **0.39**, *5.87*	0.35, **0.59**, *5.55*
	0	28.46, **92.63**, *405.01*	0.14, **1.91**, *29.98*	–0.76, **2.26**, *37.22*	1.73, **2.93**, *24.51*
Atmospheric layering	a (46 levels)	4.98, **13.26**, *70.81*	–1.25, **3.03**, *21.93*	–1.39, **3.92**, *32.38*	–0.14, **2.07**, *49.84*
	b (75 levels)	0.32, **1.70**, *13.24*	–0.13, **0.70**, *6.16*	–0.21, **1.07**, *8.64*	0.03, **0.33**, *5.83*
	c (66 levels)	0.08, **1.97**, *13.29*	–0.55, **1.37**, *16.07*	–0.36, **1.39**, *9.22*	0.03, **0.33**, *5.83*
	d (29 levels)	61.97, **118.58**, *591.82*	–1.92, **4.00**, *29.88*	–1.74, **4.52**, *31.59*	–0.13, **1.76**, *46.48*
	e (77 levels)	0.88, **2.17**, *11.59*	–0.14, **0.69**, *5.60*	–0.23, **1.07**, *7.84*	0.03, **0.32**, *5.67*
Gas/isotope number of the main gas	–1	–7.24, **11.70**, *44.30*	–0.09, **0.44**, *8.64*	0.05, **0.45**, *6.99*	–0.08, **0.61**, *10.13*

Table 5: Mean-value, **standard deviation**, and *maximum* of the relative errors (%) with respect to MIPAS NESR in channel D for various model parameter settings and tangent altitudes.

Relative run-times		
		Relative time
Ray–tracing step length	1 km	1
	10 km	0.96
Finest spectral grid	0.0005 cm^{-1}	1
	0.0008 cm^{-1}	0.92
	0.001 cm^{-1}	0.86
Accuracy for cross–section calculation	10^{-8}	1
	10^{-6}	0.64
	10^{-4}	0.44
Width of AILS function	1.40 cm^{-1}	1
	0.15 cm^{-1}	0.74
Number of cross–section recalculations for limb–scans	3	1
	2	0.86
	1	0.72
	0	0.58
Additional ray–paths for field–of–view	2	1
	1	0.73
	0	0.46
Atmospheric layering	a (46 levels)	1
	b (75 levels)	1.43
	c (66 levels)	1.27
	d (29 levels)	0.93
	e (77 levels)	1.43
Temperature for Planck function calculation	0	1
	−1	0.85

Table 6: Relative run-times for a microwindow selection for ozone retrieval.

Appendix A

Parameter optimization for the line–by–line radiative transfer model KOPRA to be used in MIPAS–ENVISAT retrievals

Band A: 685 – 970 cm^{-1}

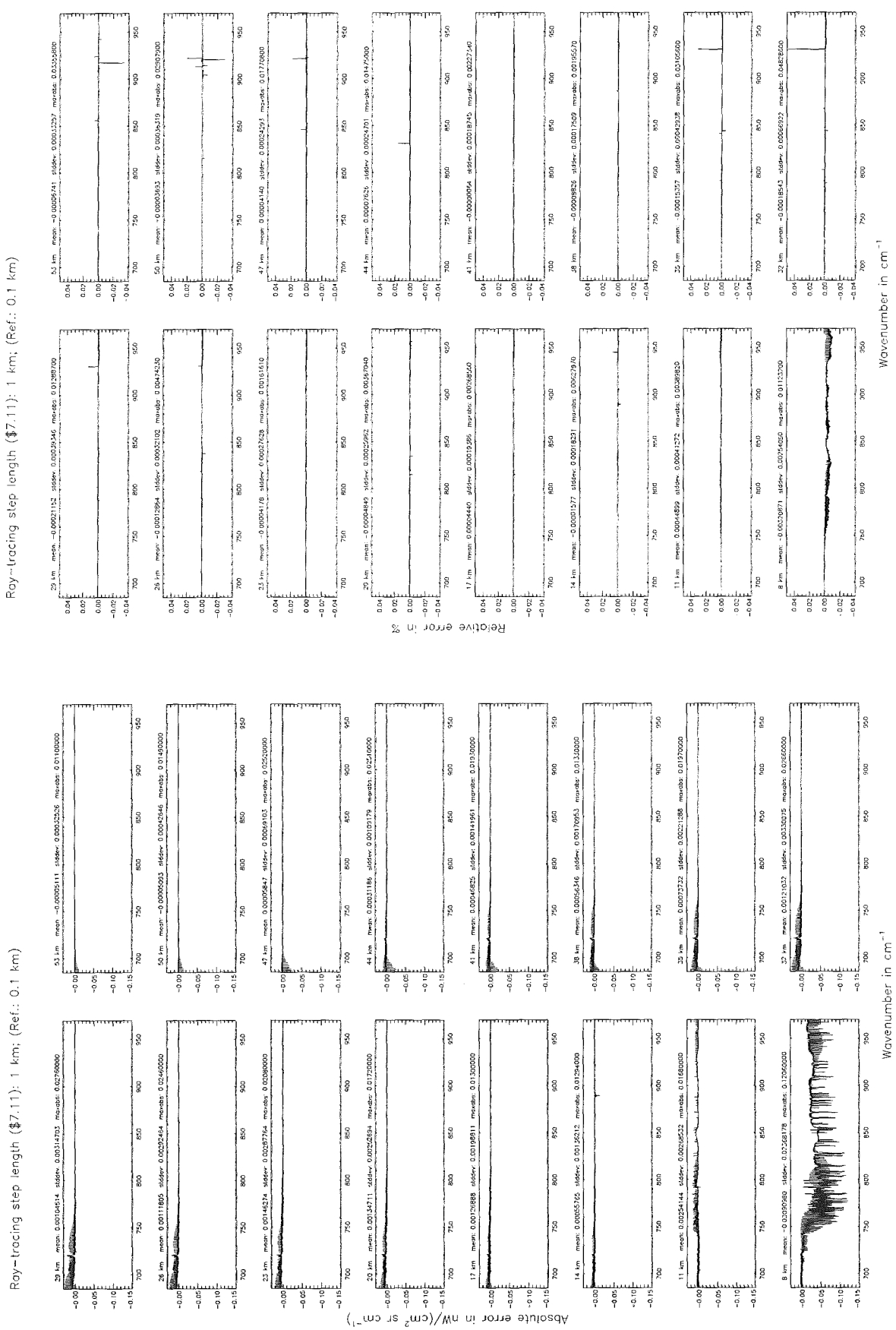


Figure 1: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

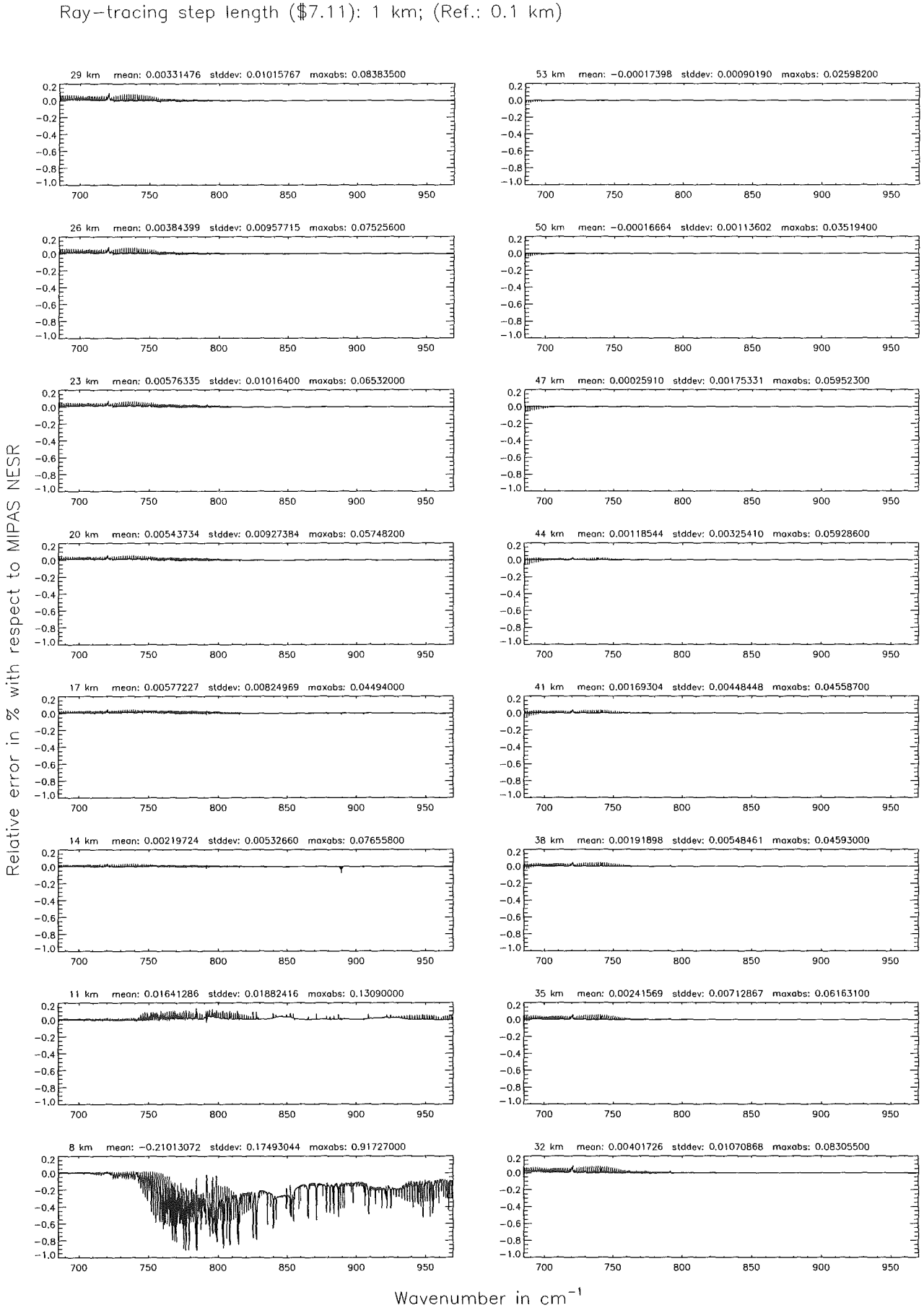


Figure 2: Relative error [%] with respect to MIPAS NESR

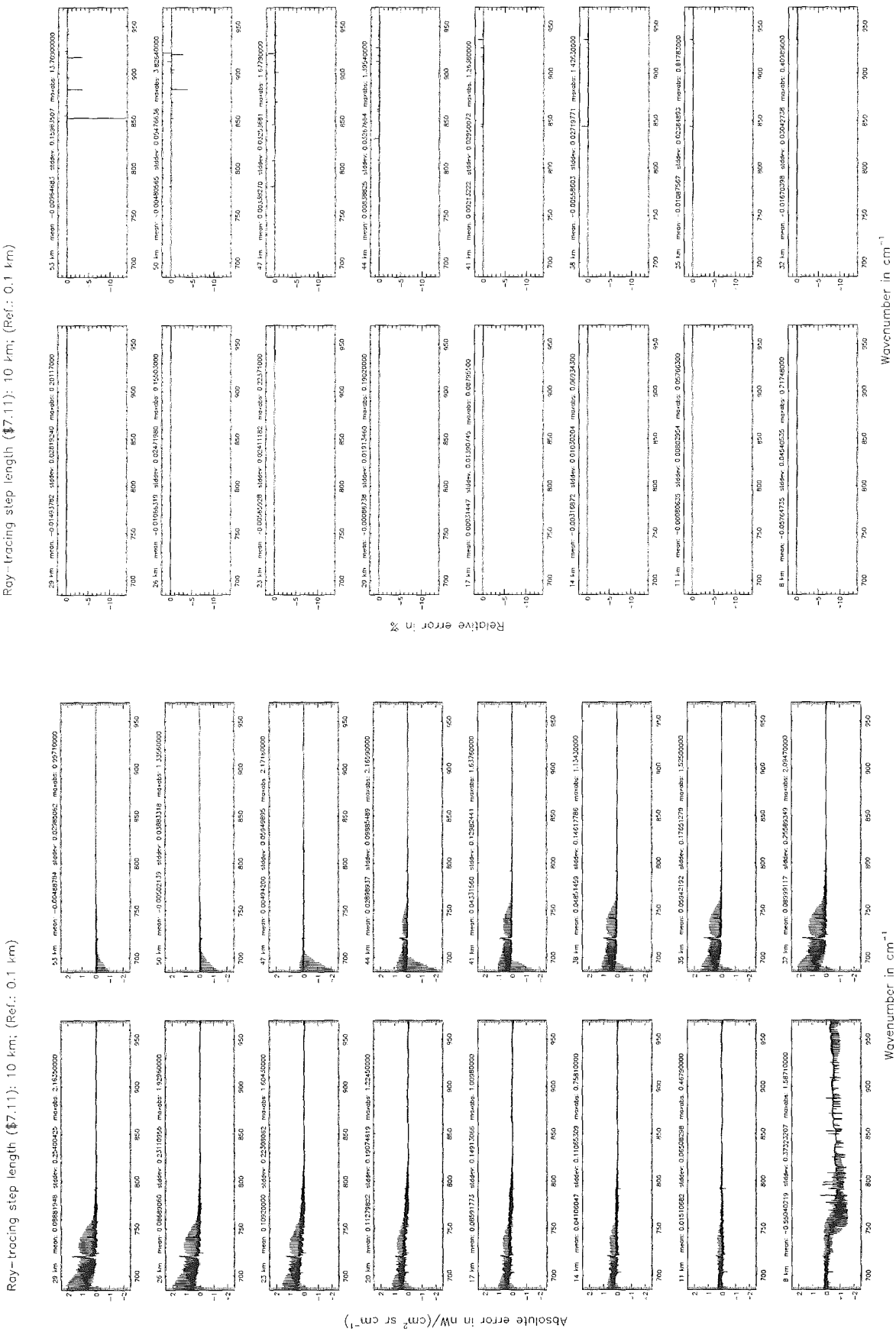


Figure 3: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

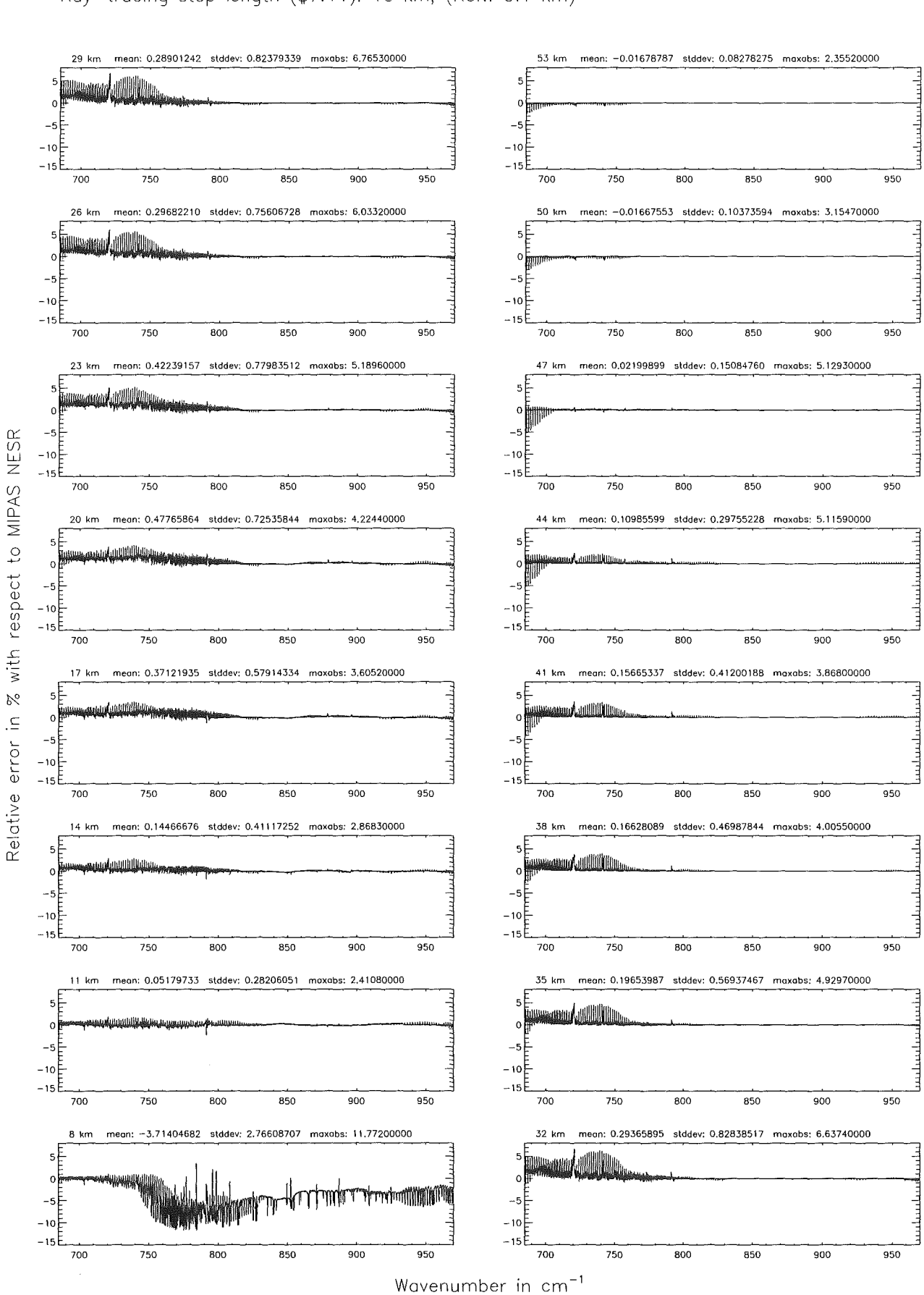


Figure 4: Relative error [%] with respect to MIPAS NESR

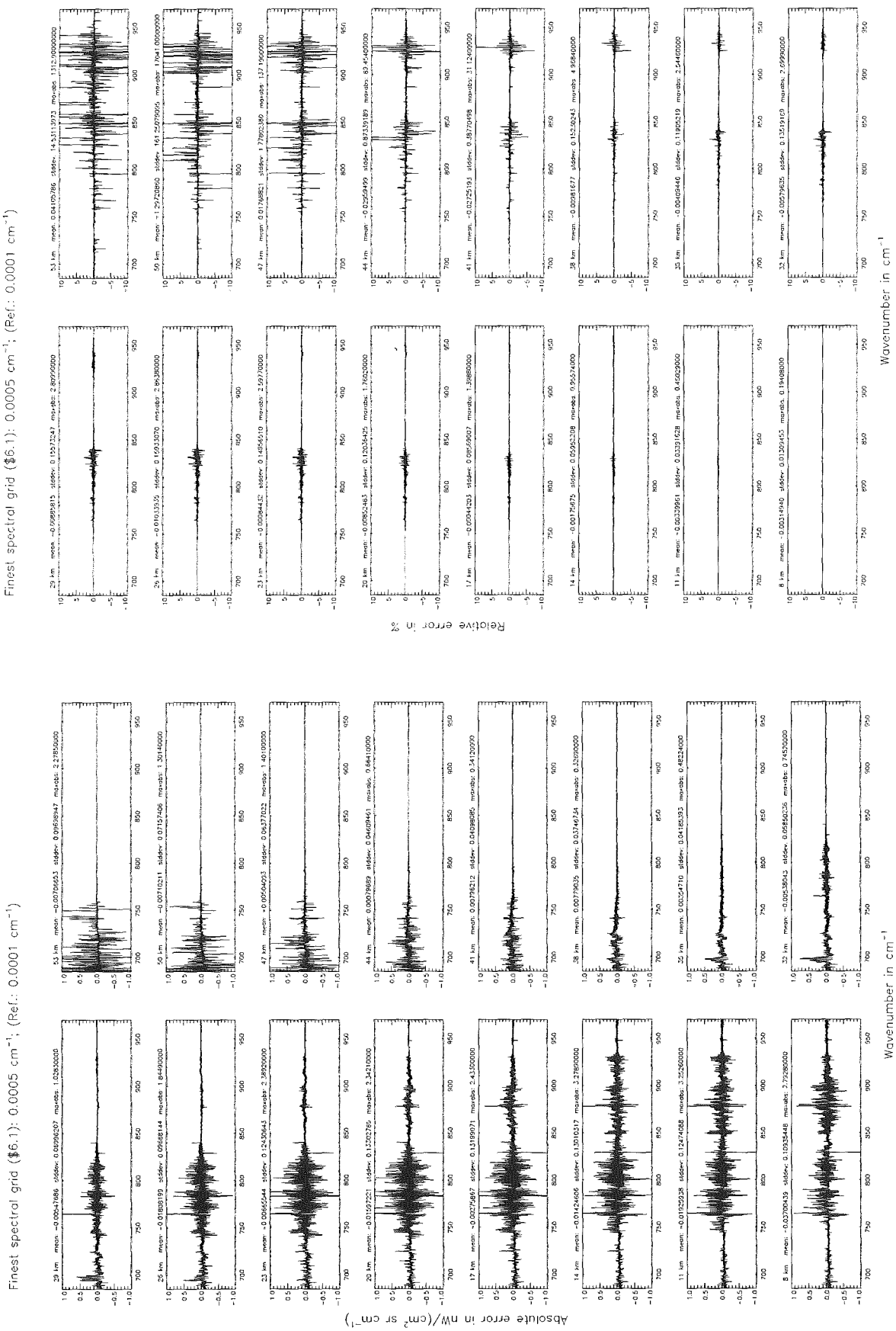


Figure 5: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

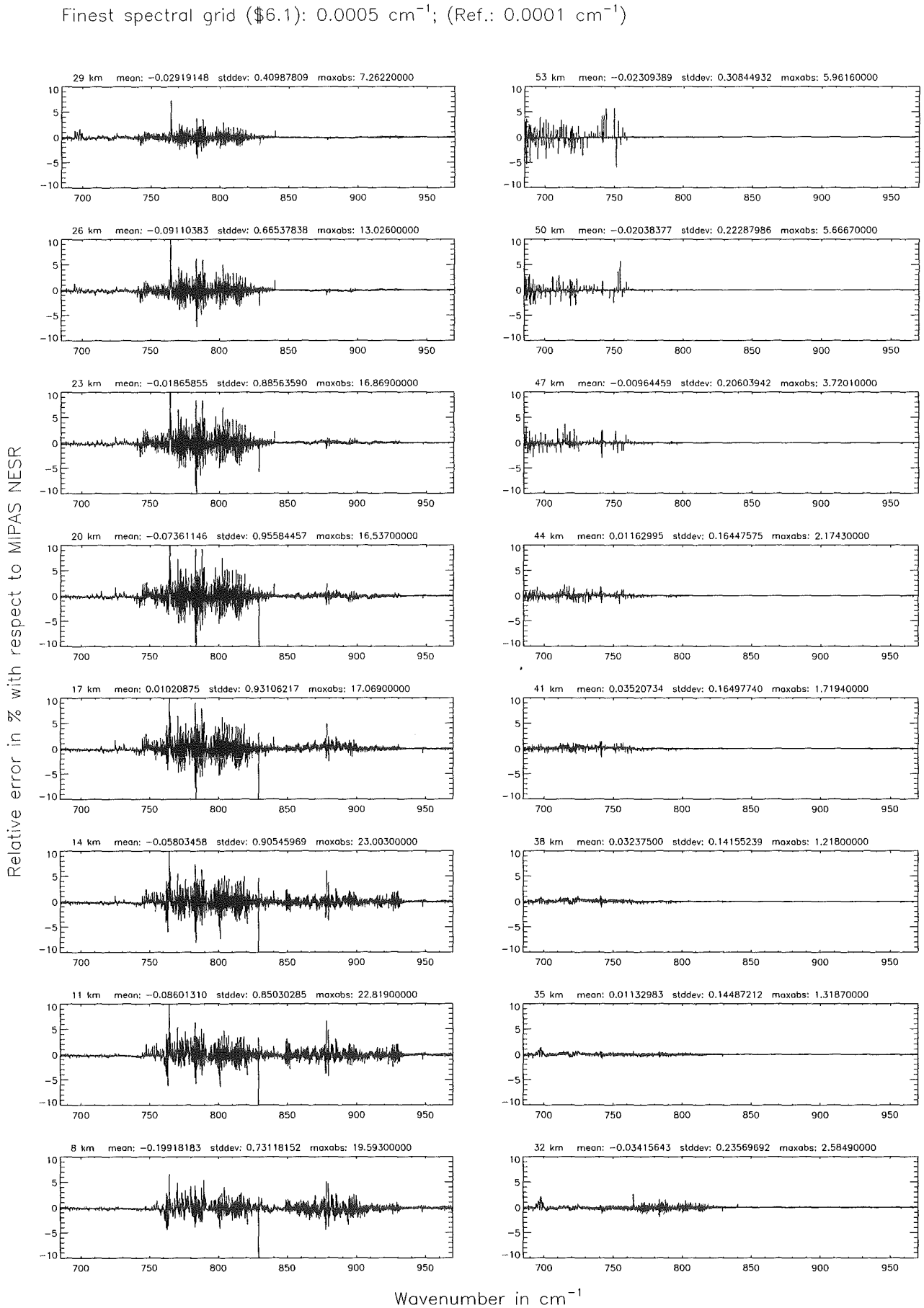


Figure 6: Relative error [%] with respect to MIPAS NESR

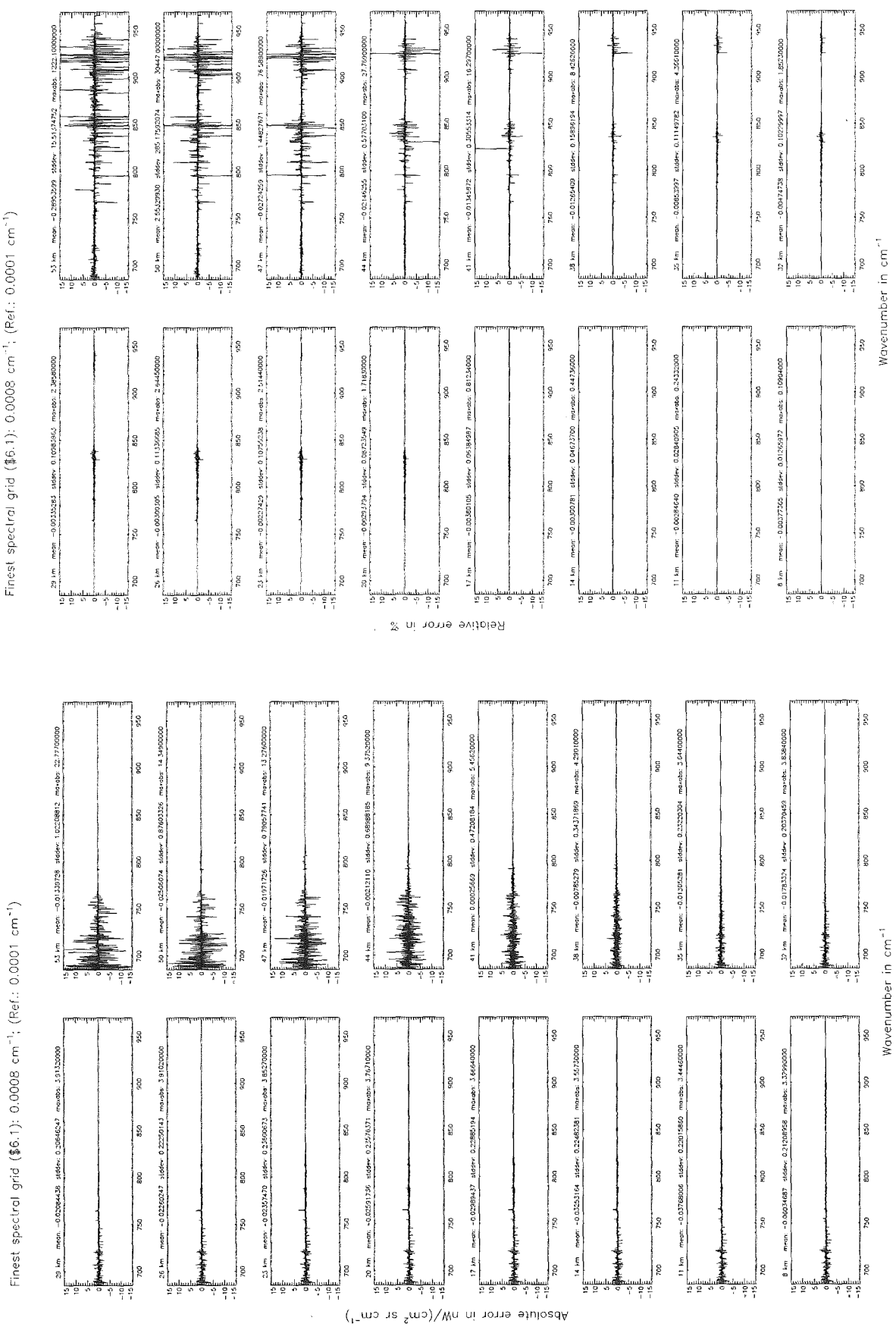


Figure 7: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

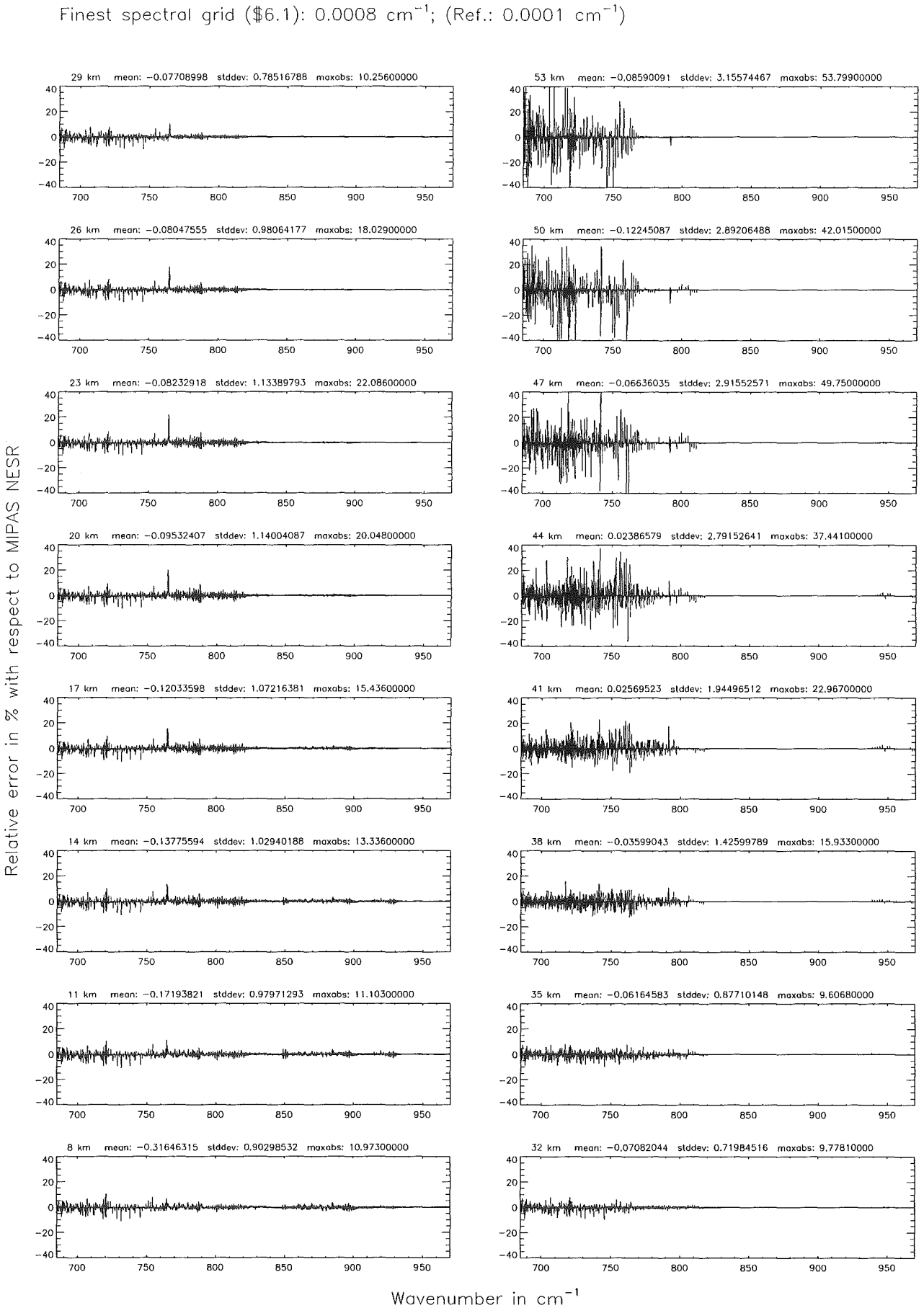


Figure 8: Relative error [%] with respect to MIPAS NESR

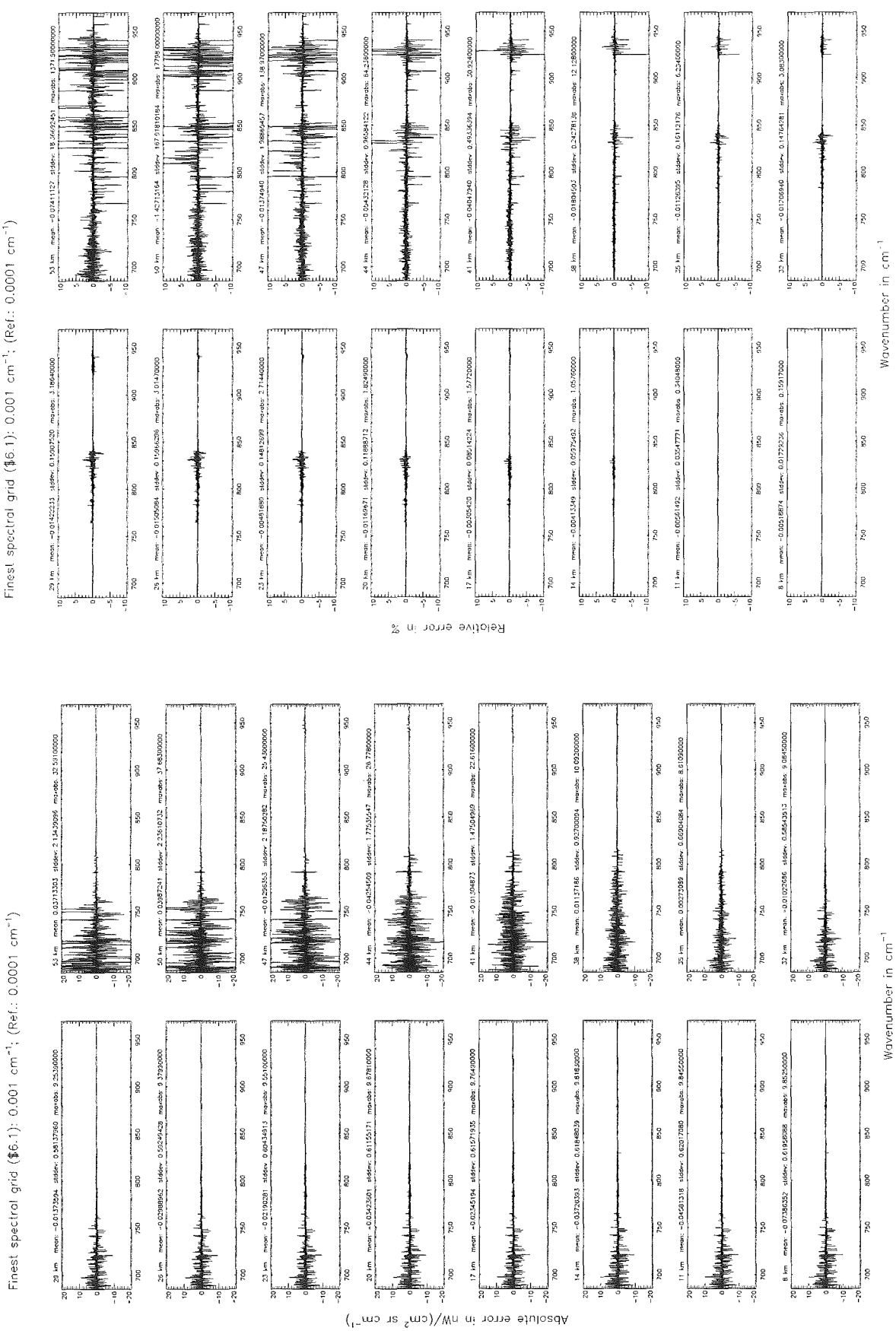


Figure 9: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

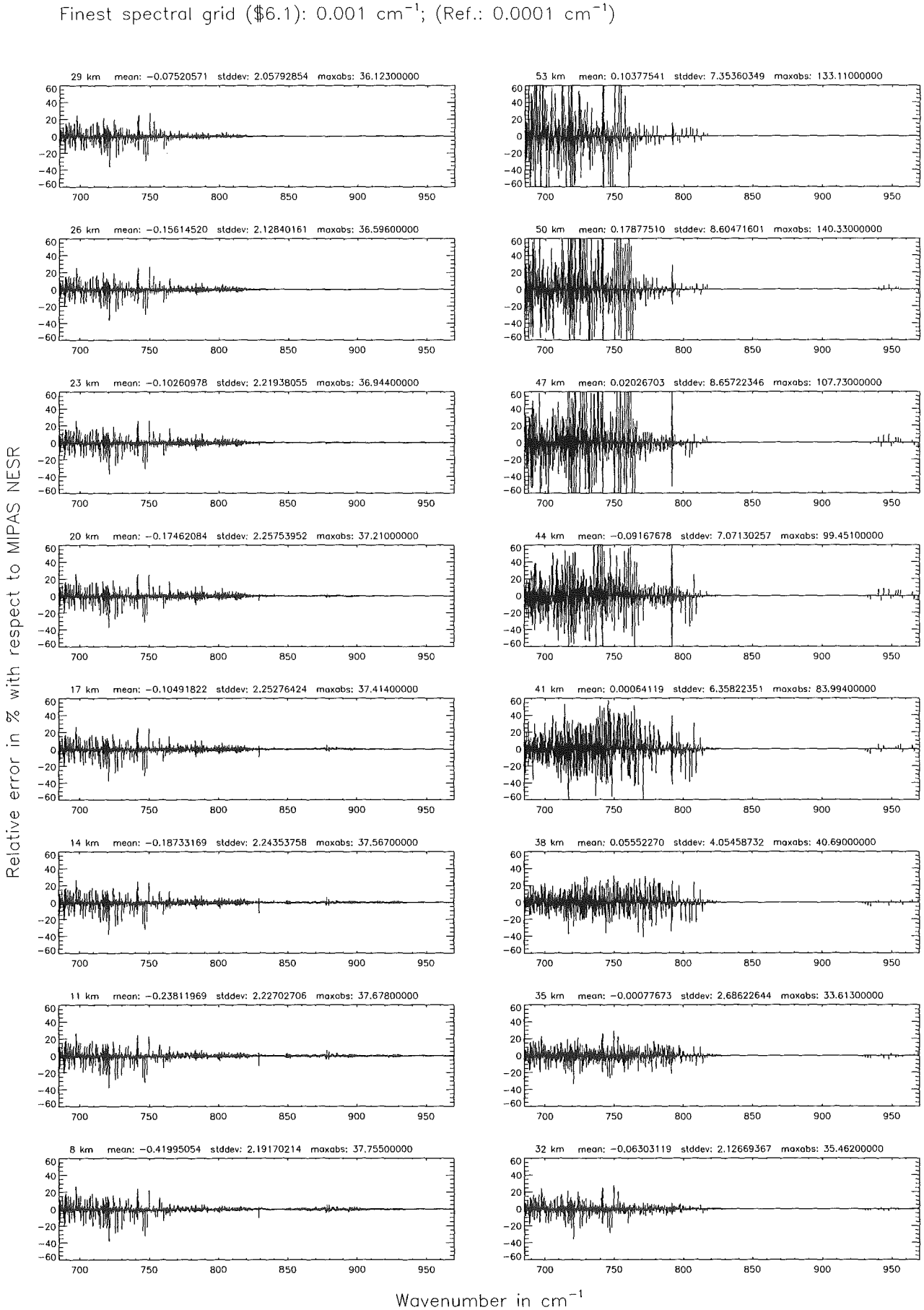


Figure 10: Relative error [%] with respect to MIPAS NESR

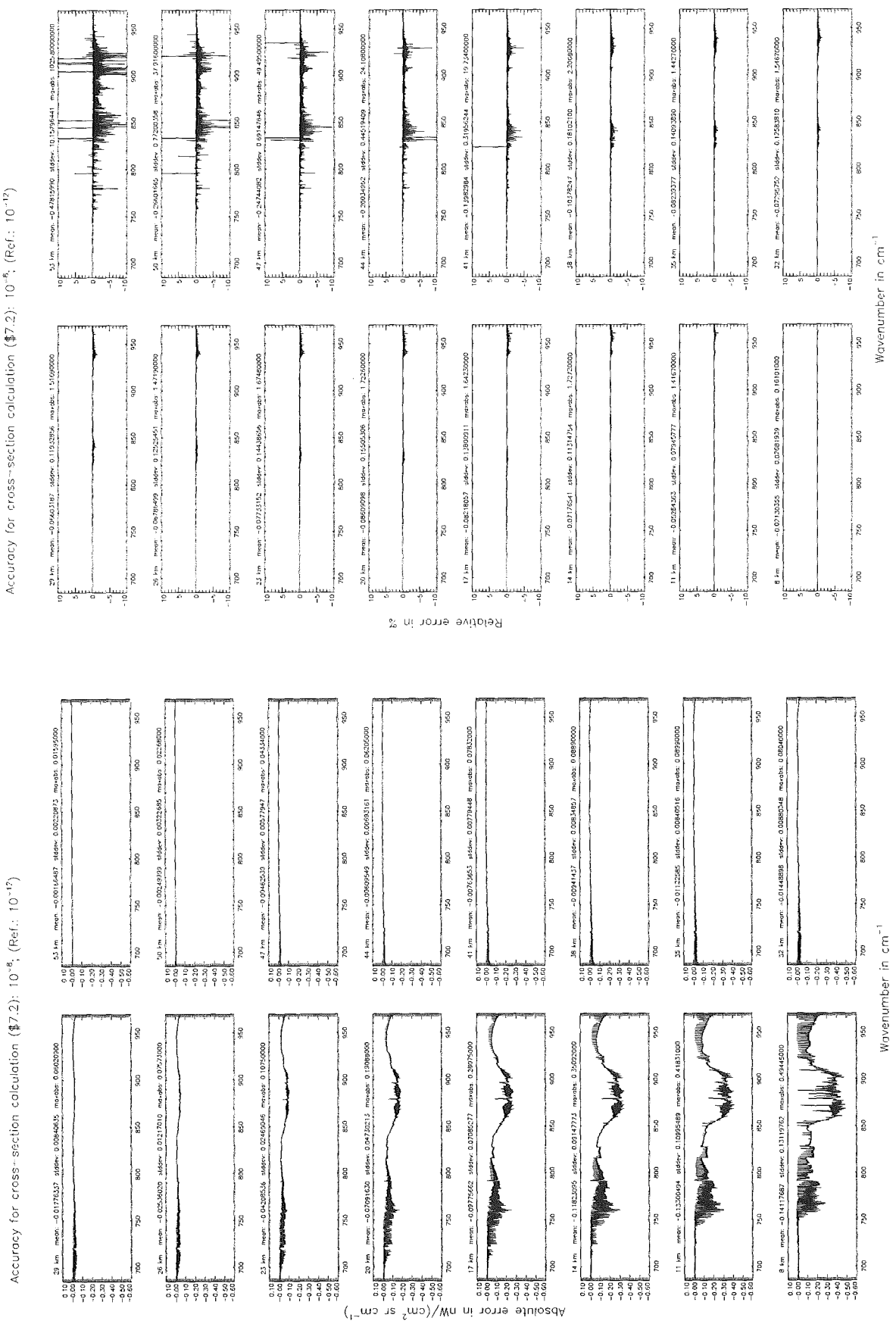


Figure 11: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

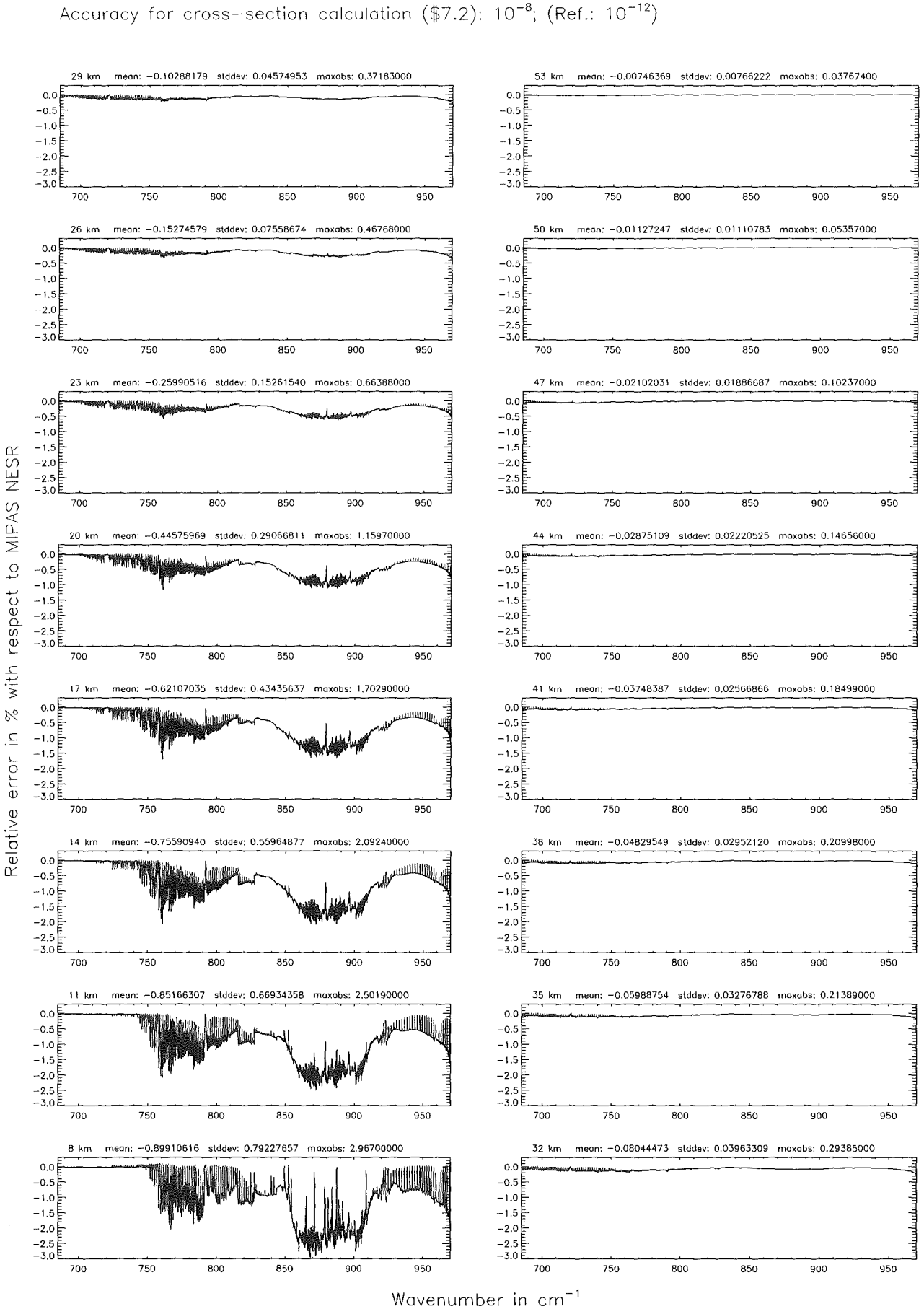


Figure 12: Relative error [%] with respect to MIPAS NESR

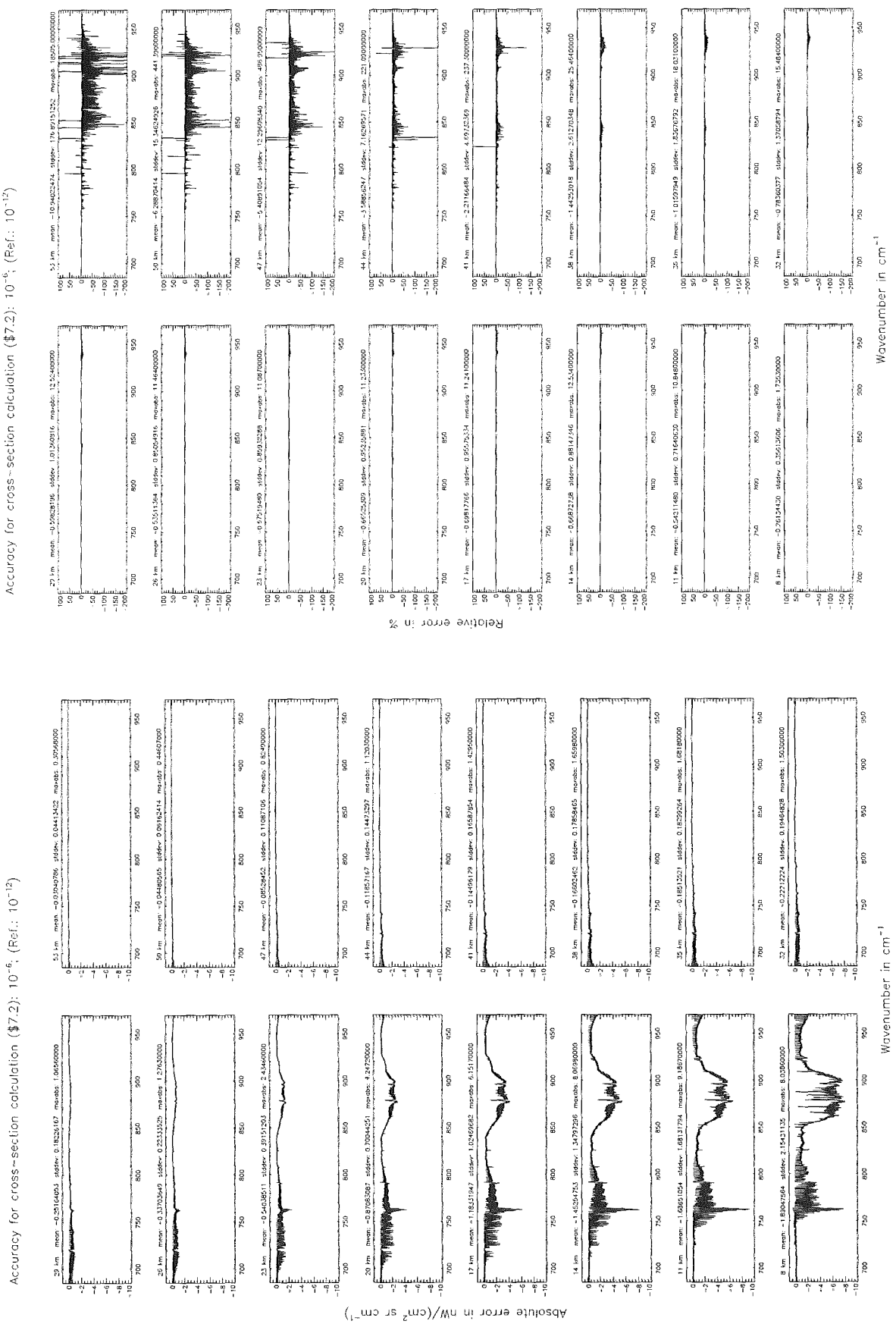


Figure 13: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

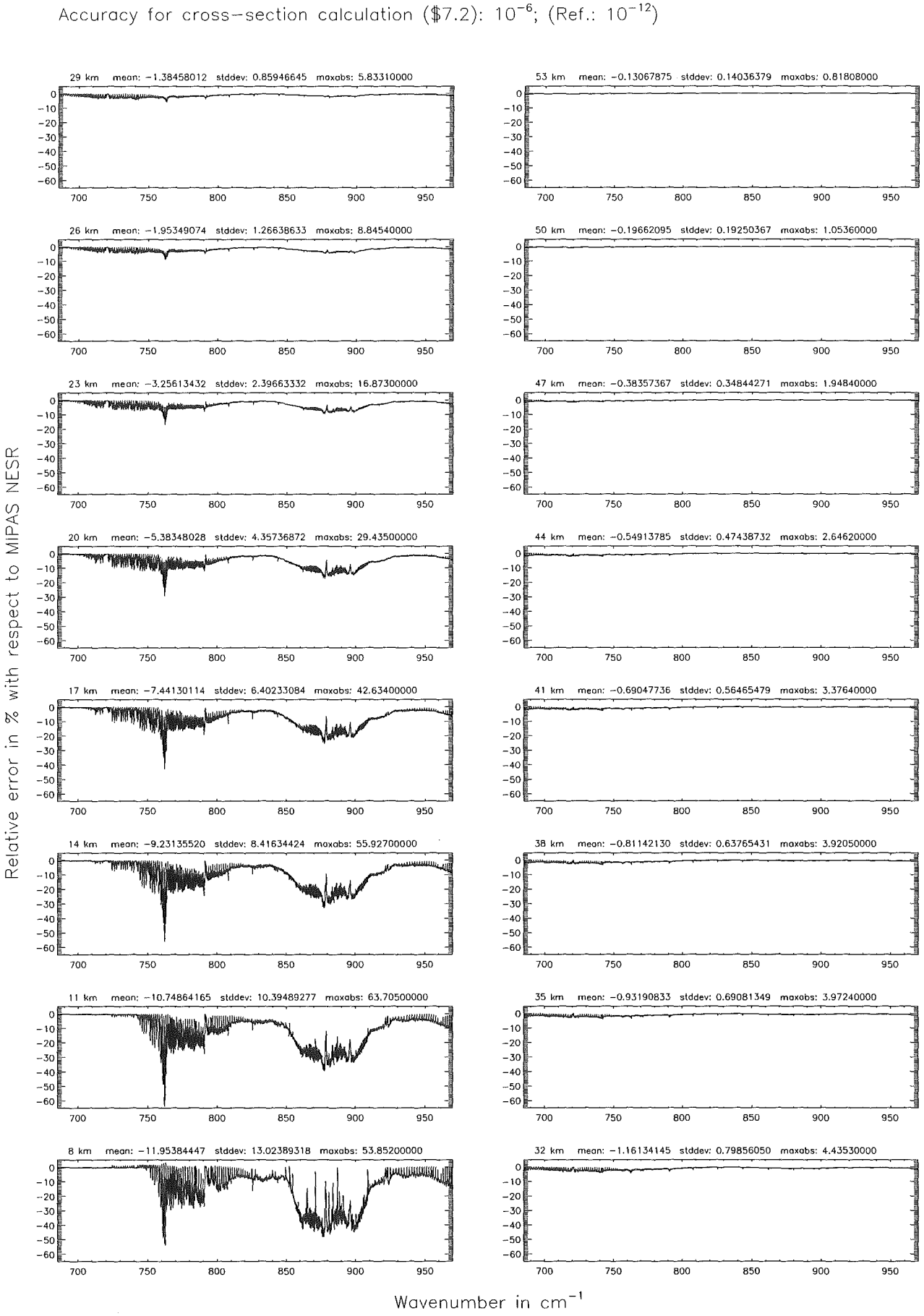


Figure 14: Relative error [%] with respect to MIPAS NESR

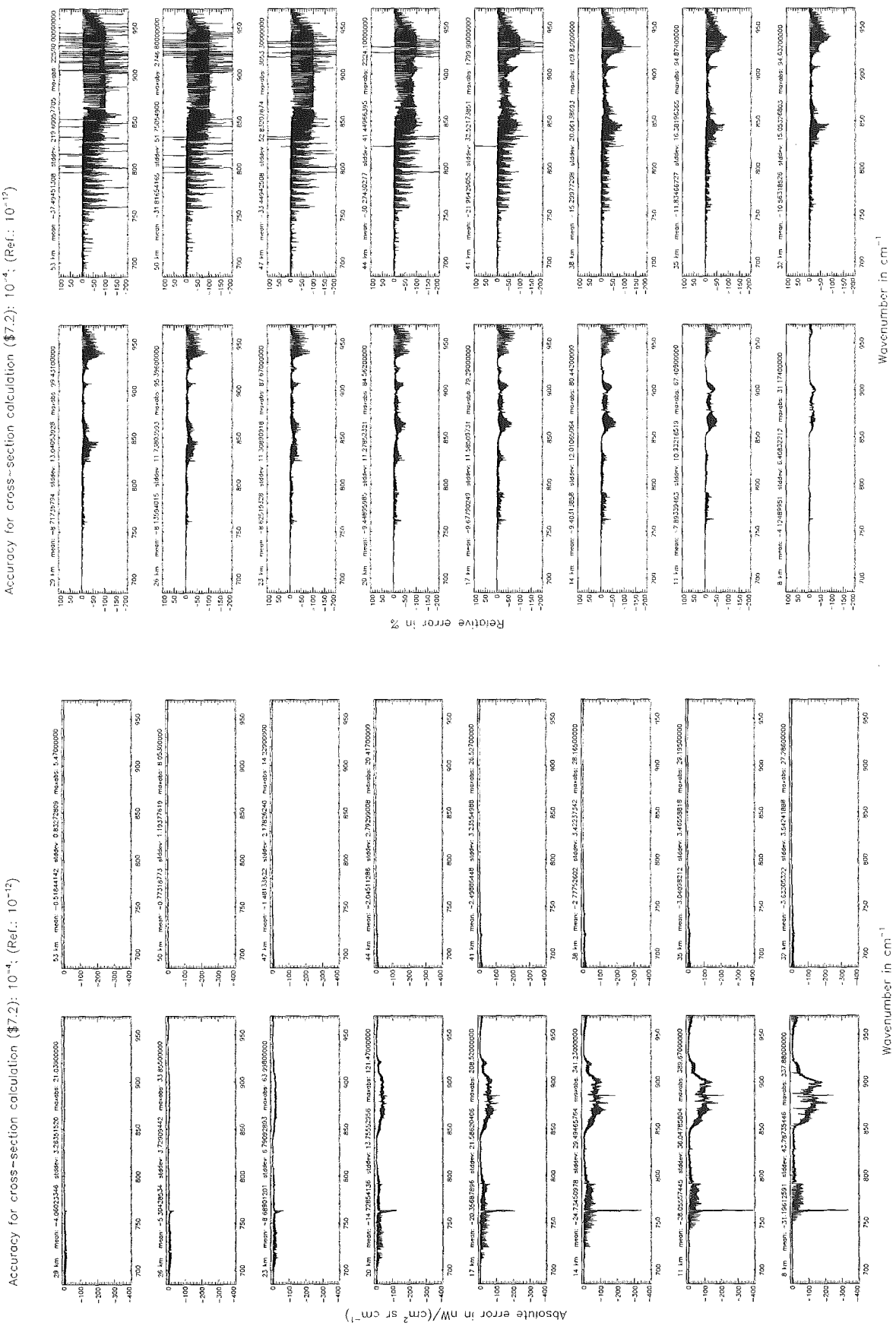


Figure 15: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

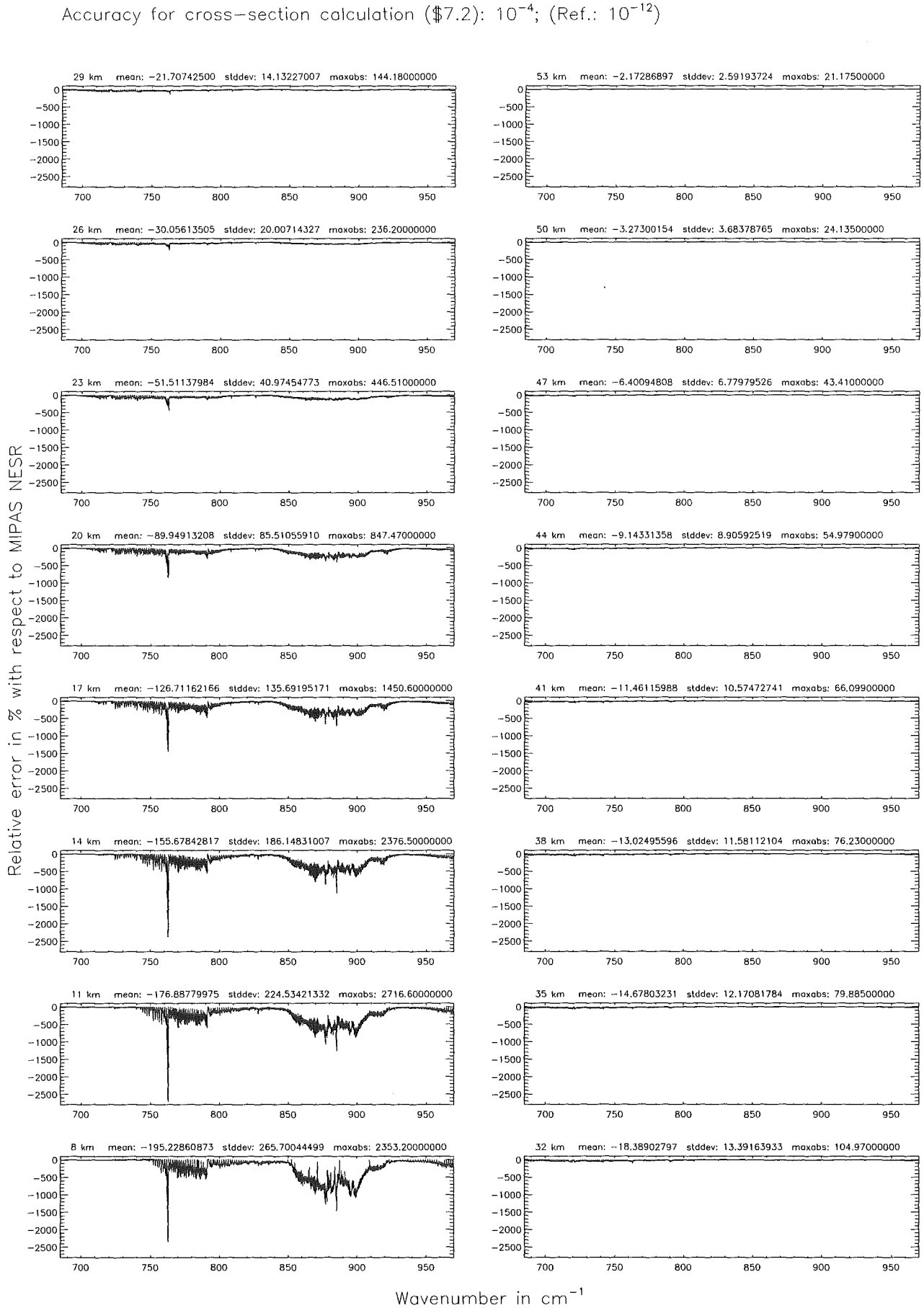


Figure 16: Relative error [%] with respect to MIPAS NESR

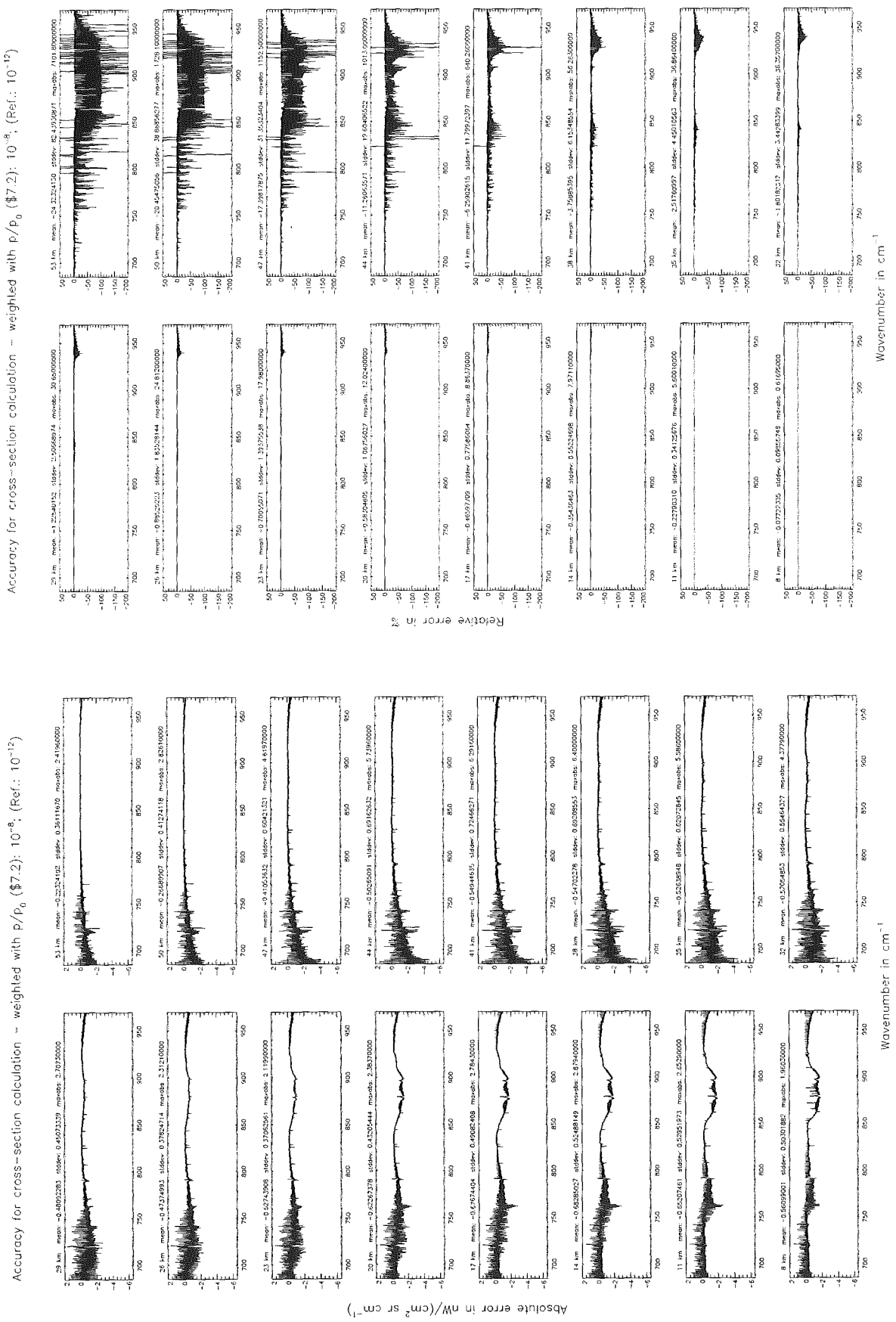


Figure 17: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

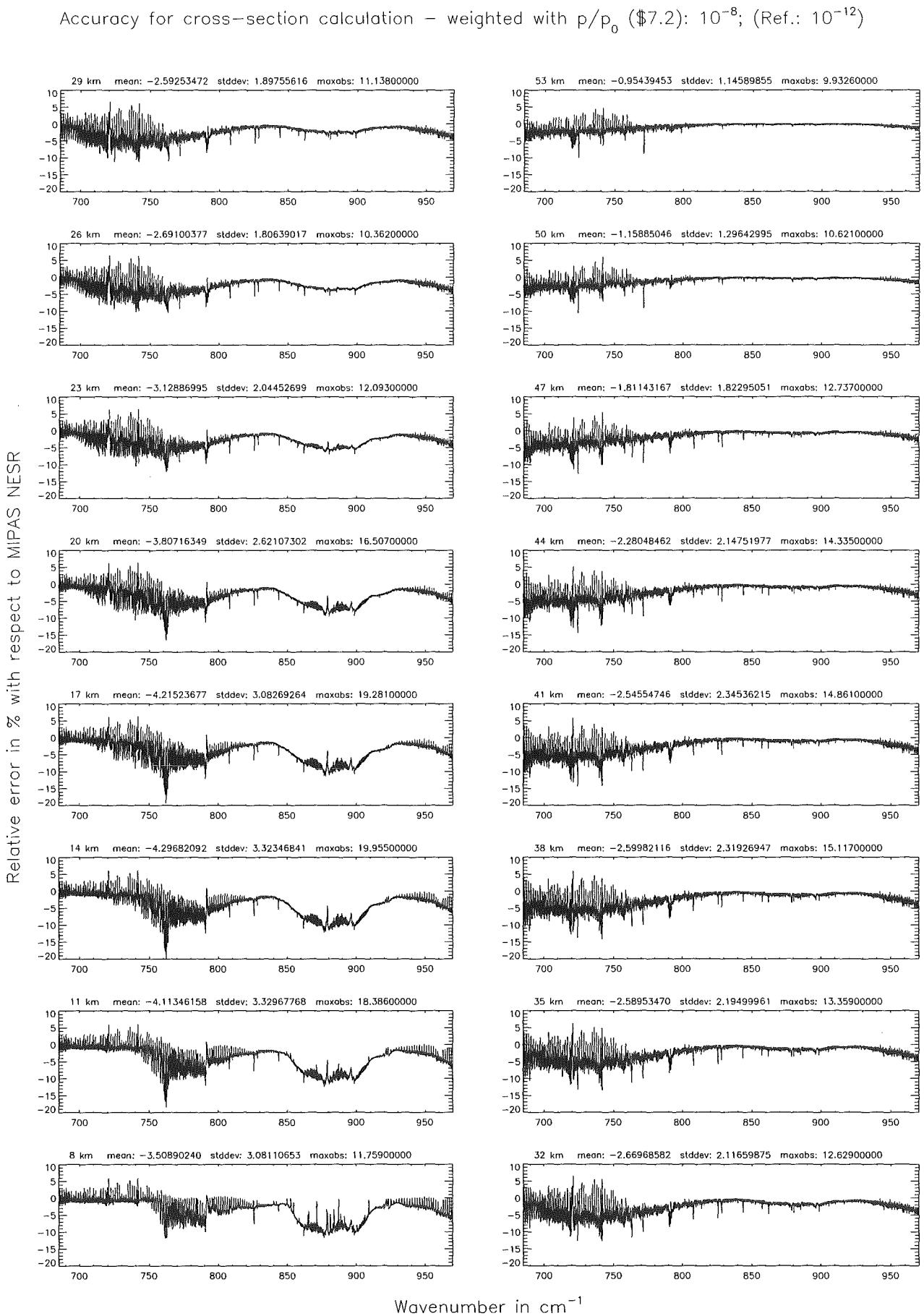


Figure 18: Relative error [%] with respect to MIPAS NESR

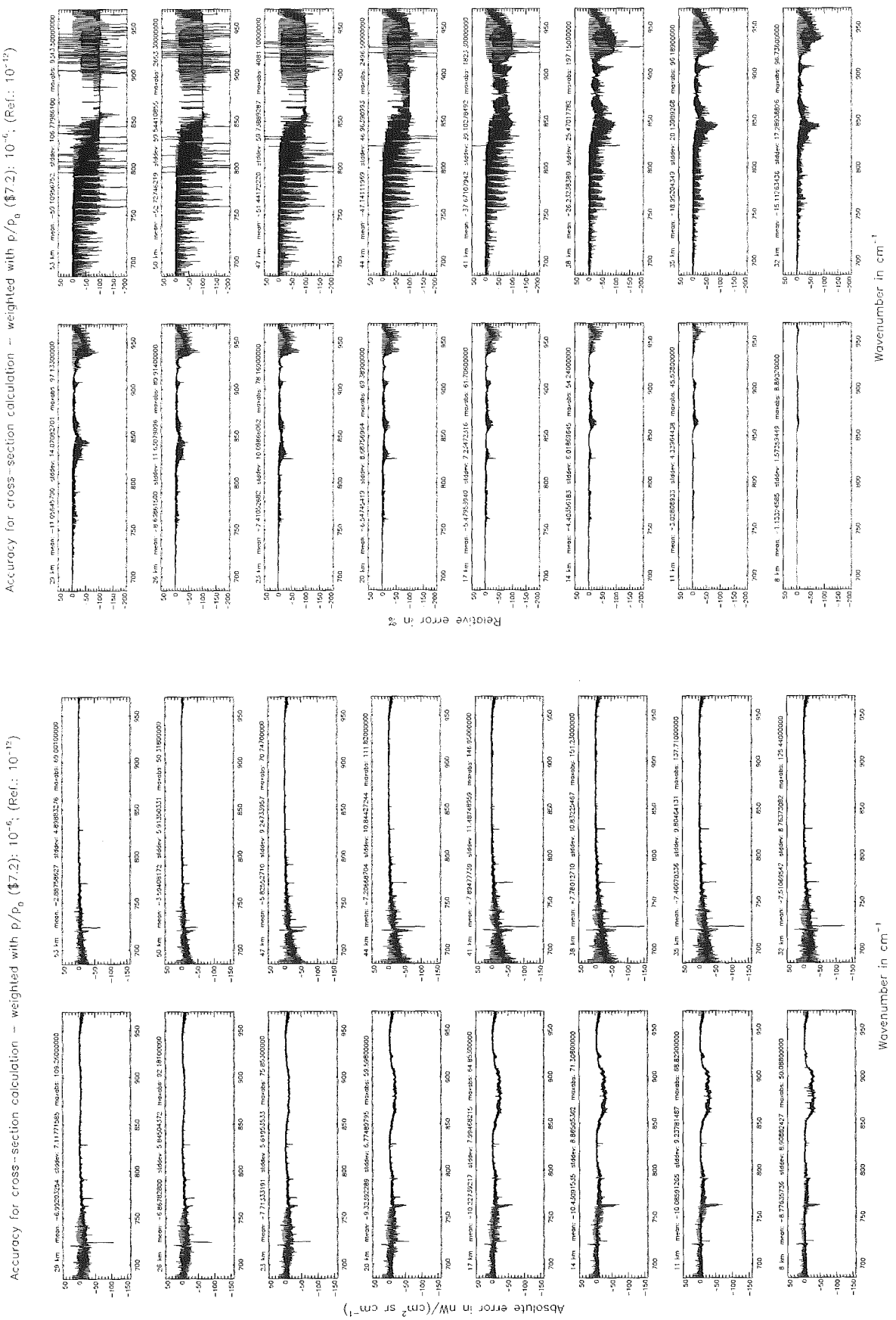


Figure 19: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

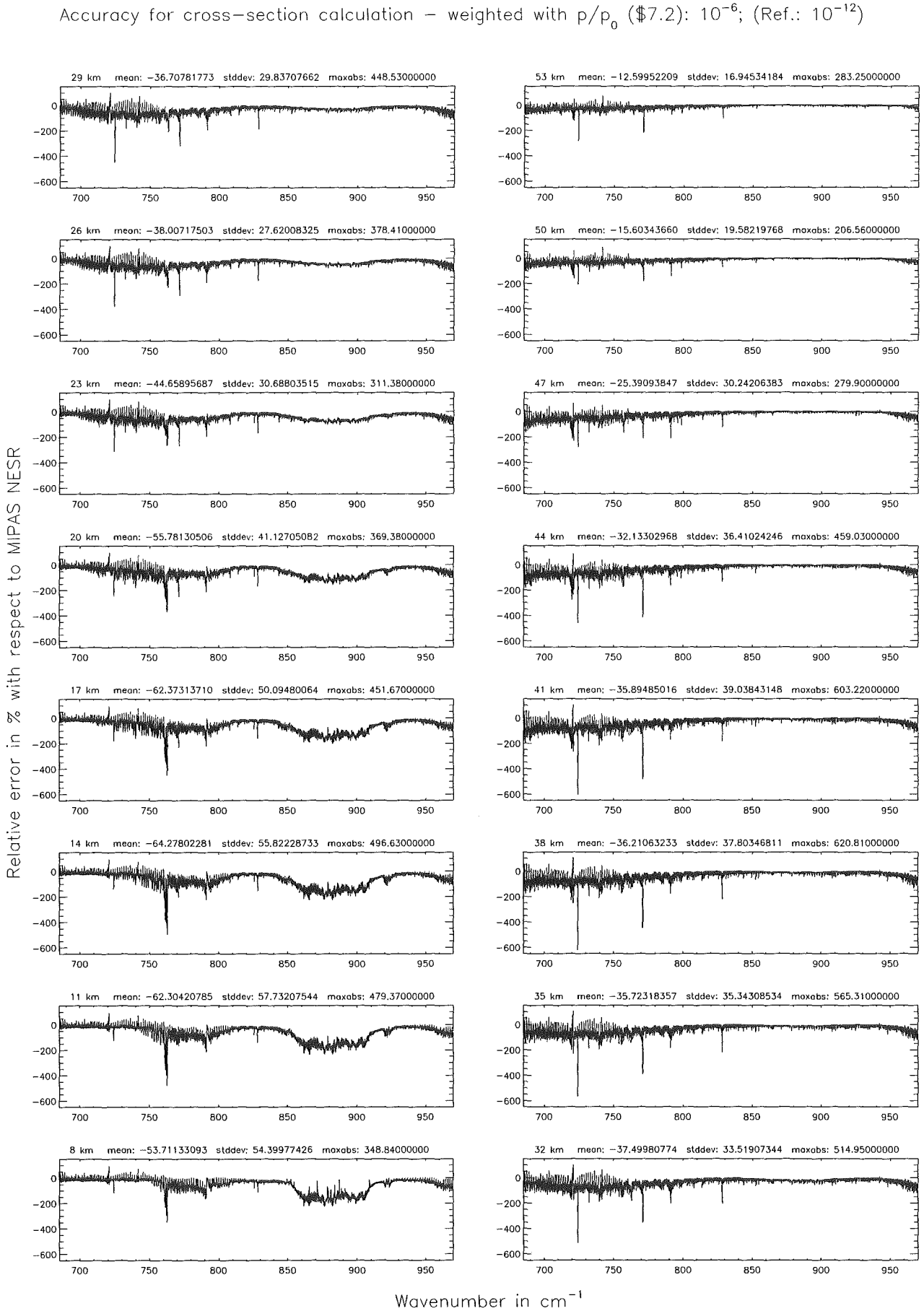


Figure 20: Relative error [%] with respect to MIPAS NESR

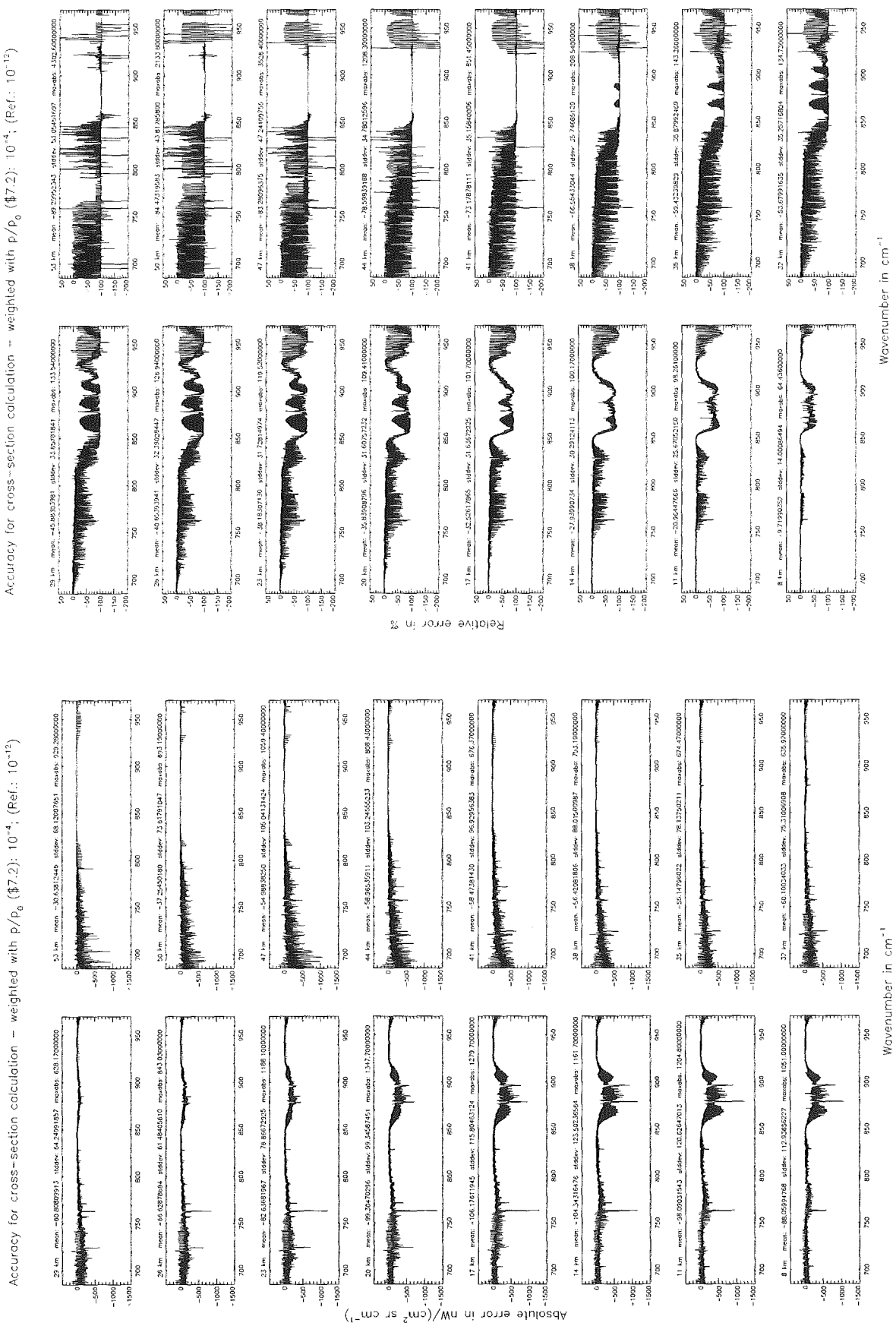


Figure 21: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

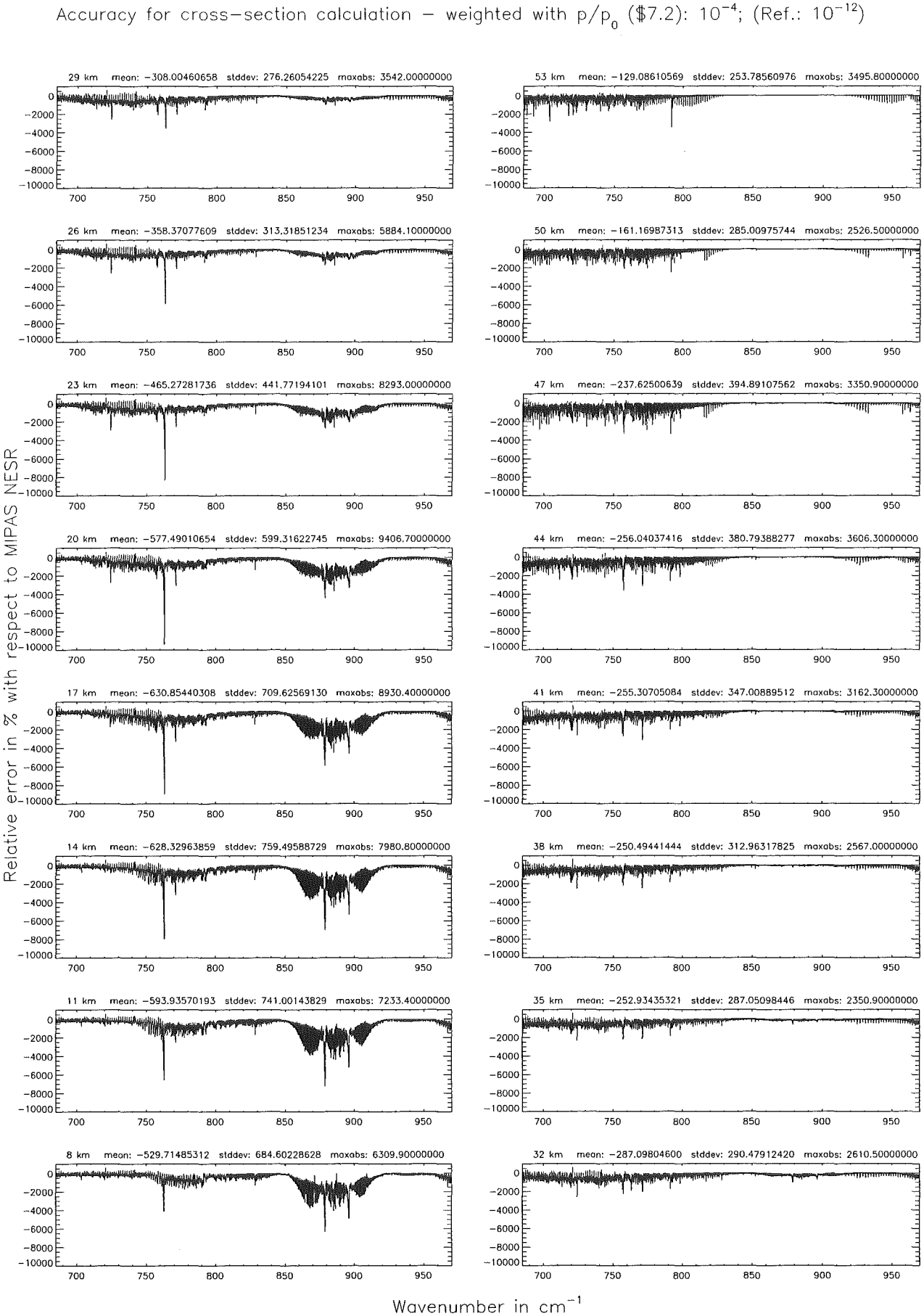


Figure 22: Relative error [%] with respect to MIPAS NESR

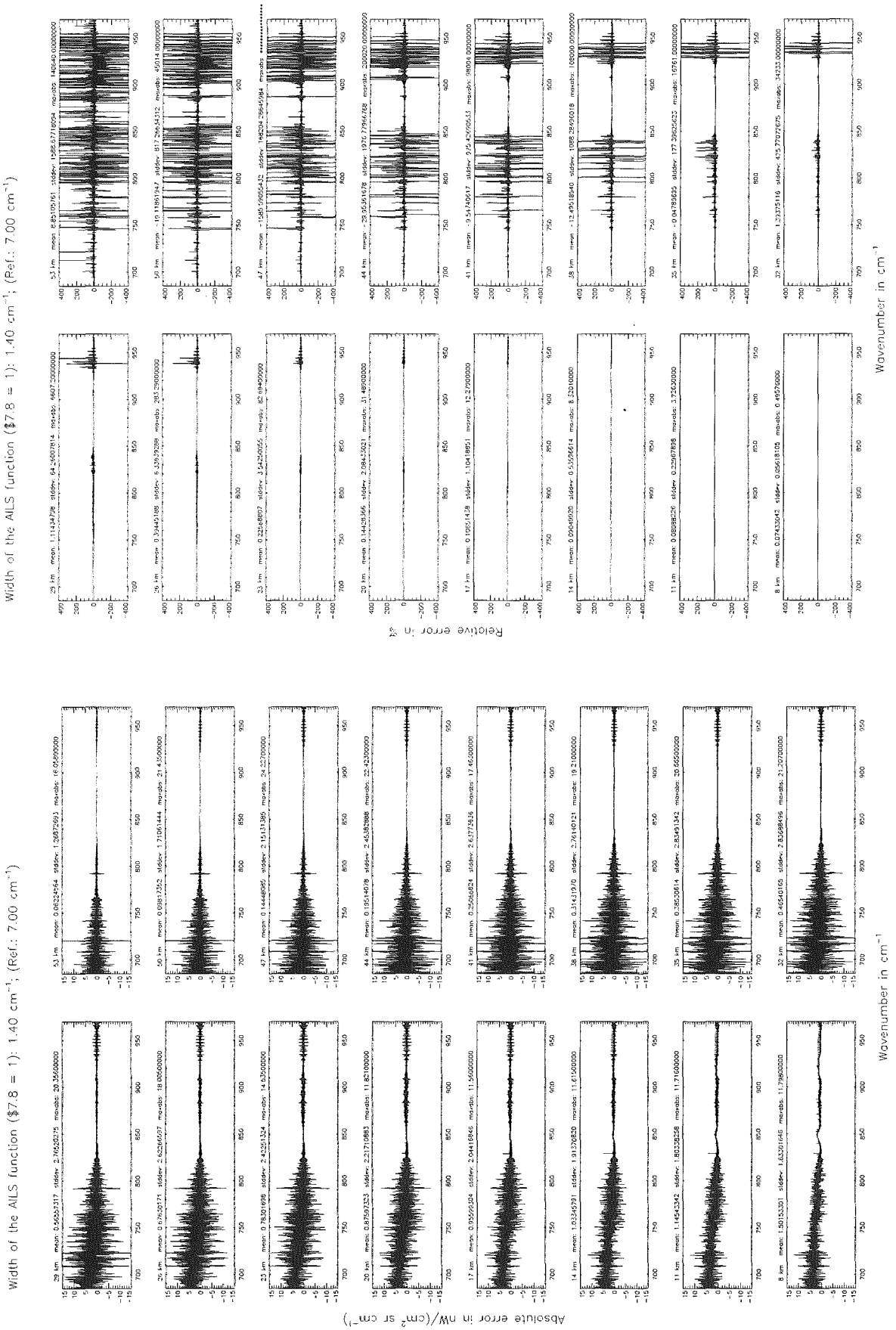


Figure 23: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

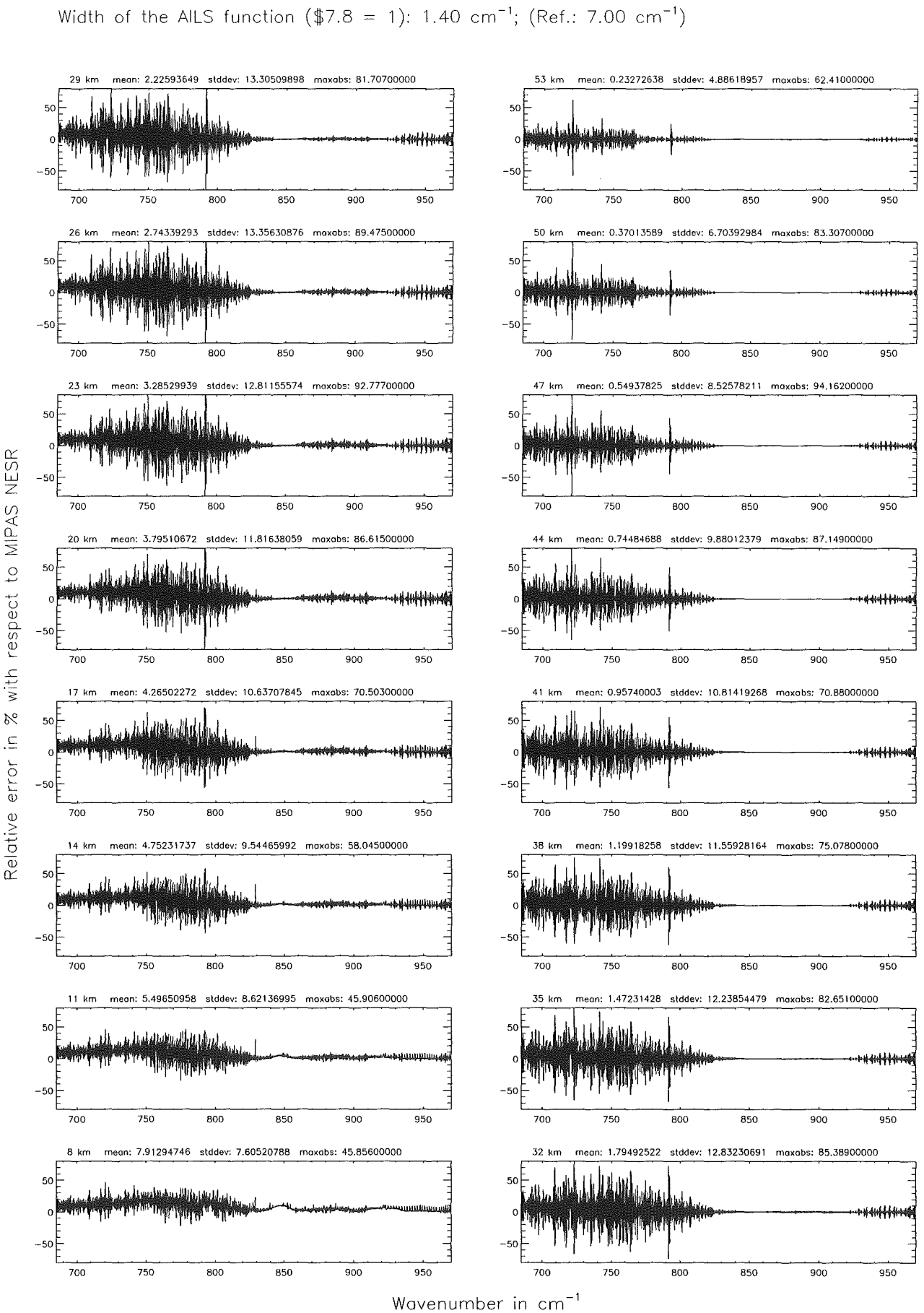


Figure 24: Relative error [%] with respect to MIPAS NESR

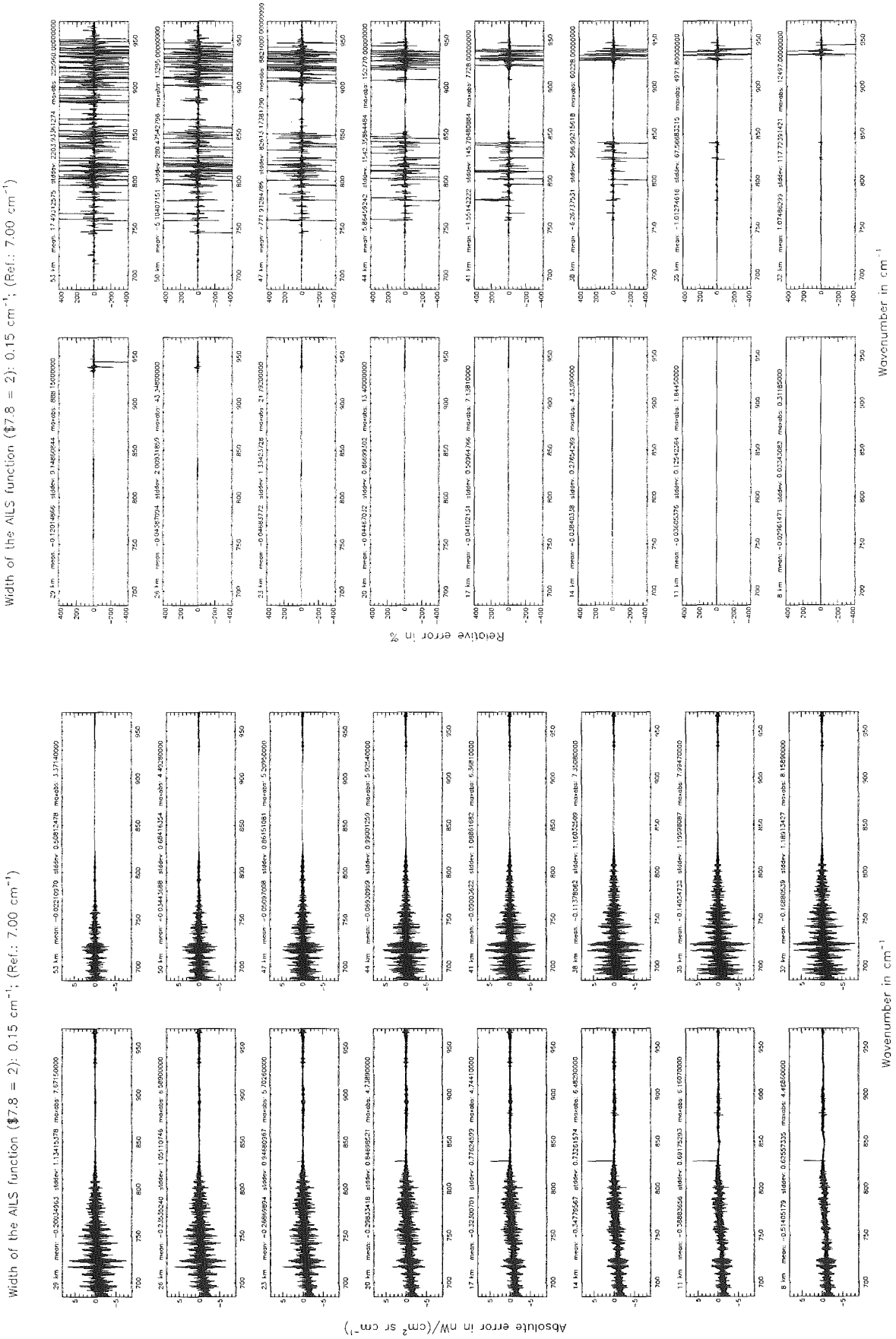


Figure 25: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

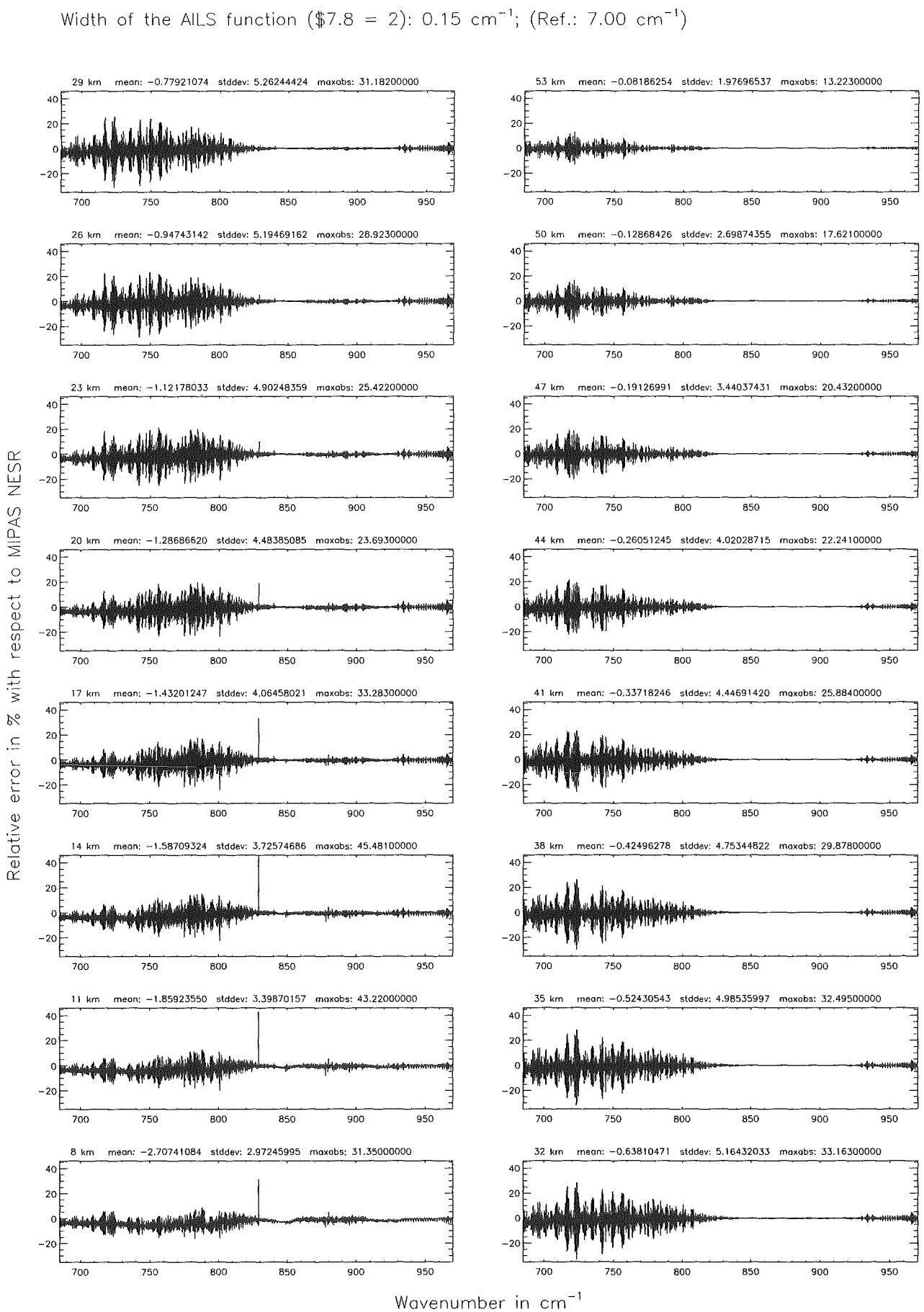


Figure 26: Relative error [%] with respect to MIPAS NESR

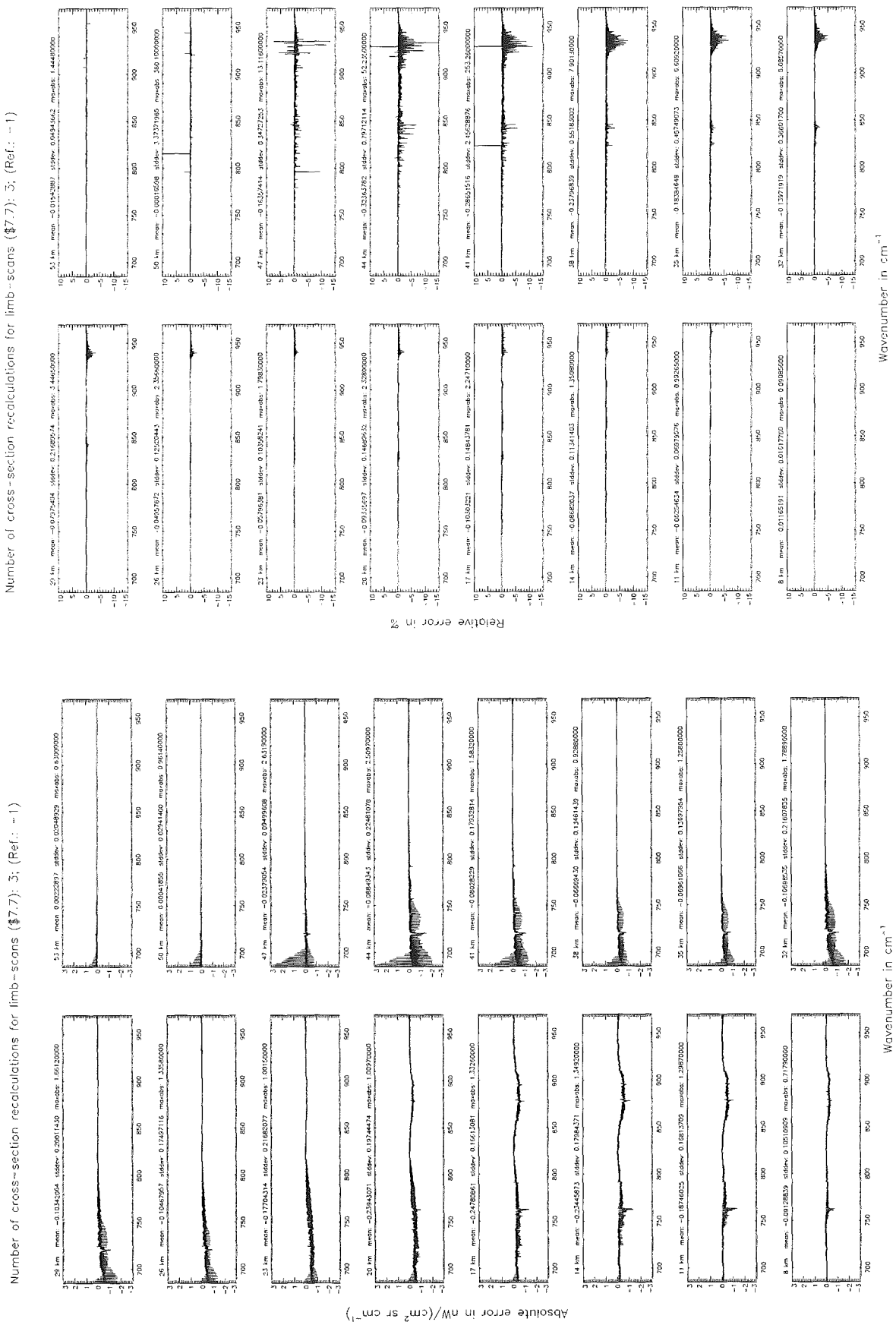


Figure 27: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

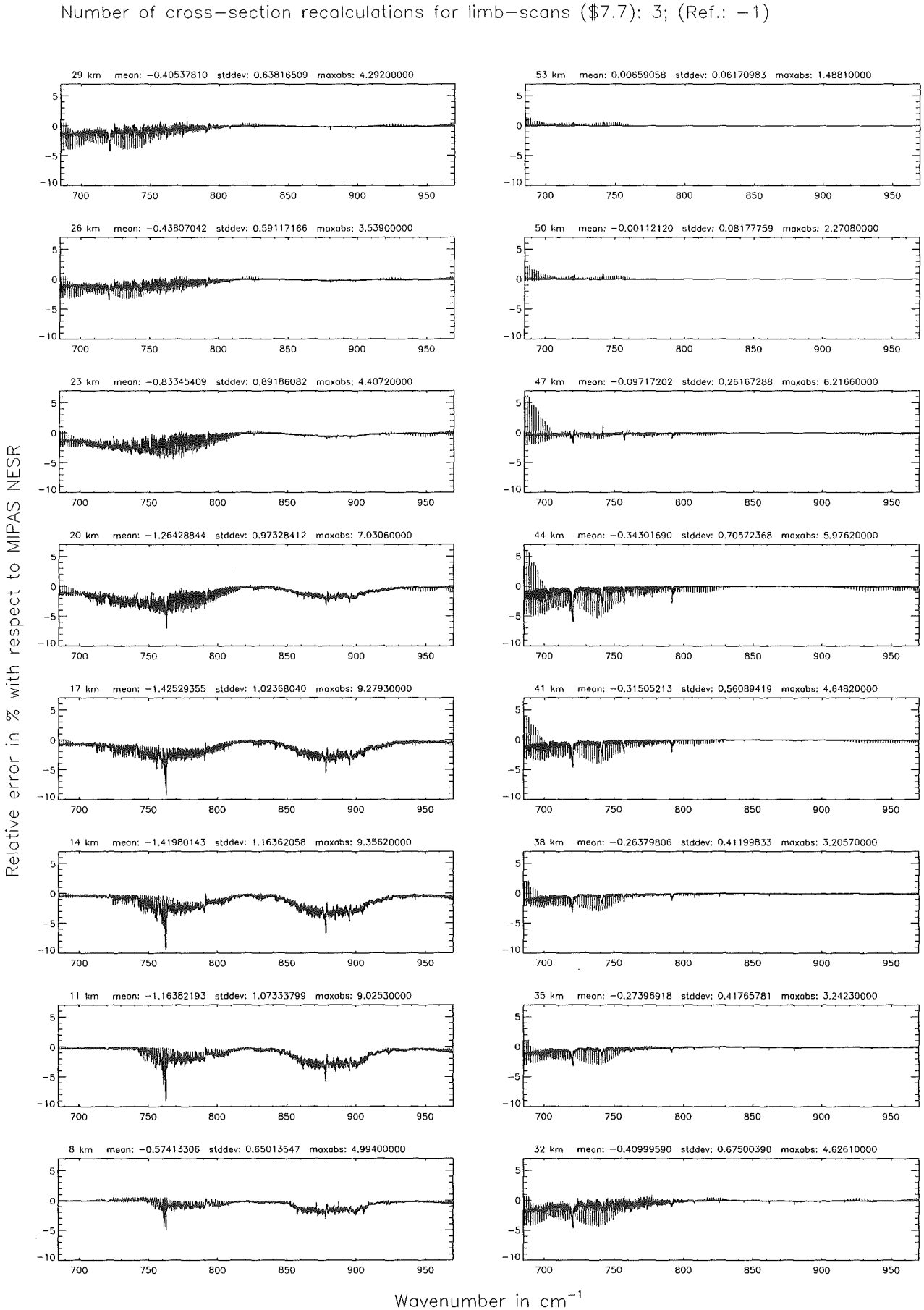


Figure 28: Relative error [%] with respect to MIPAS NESR

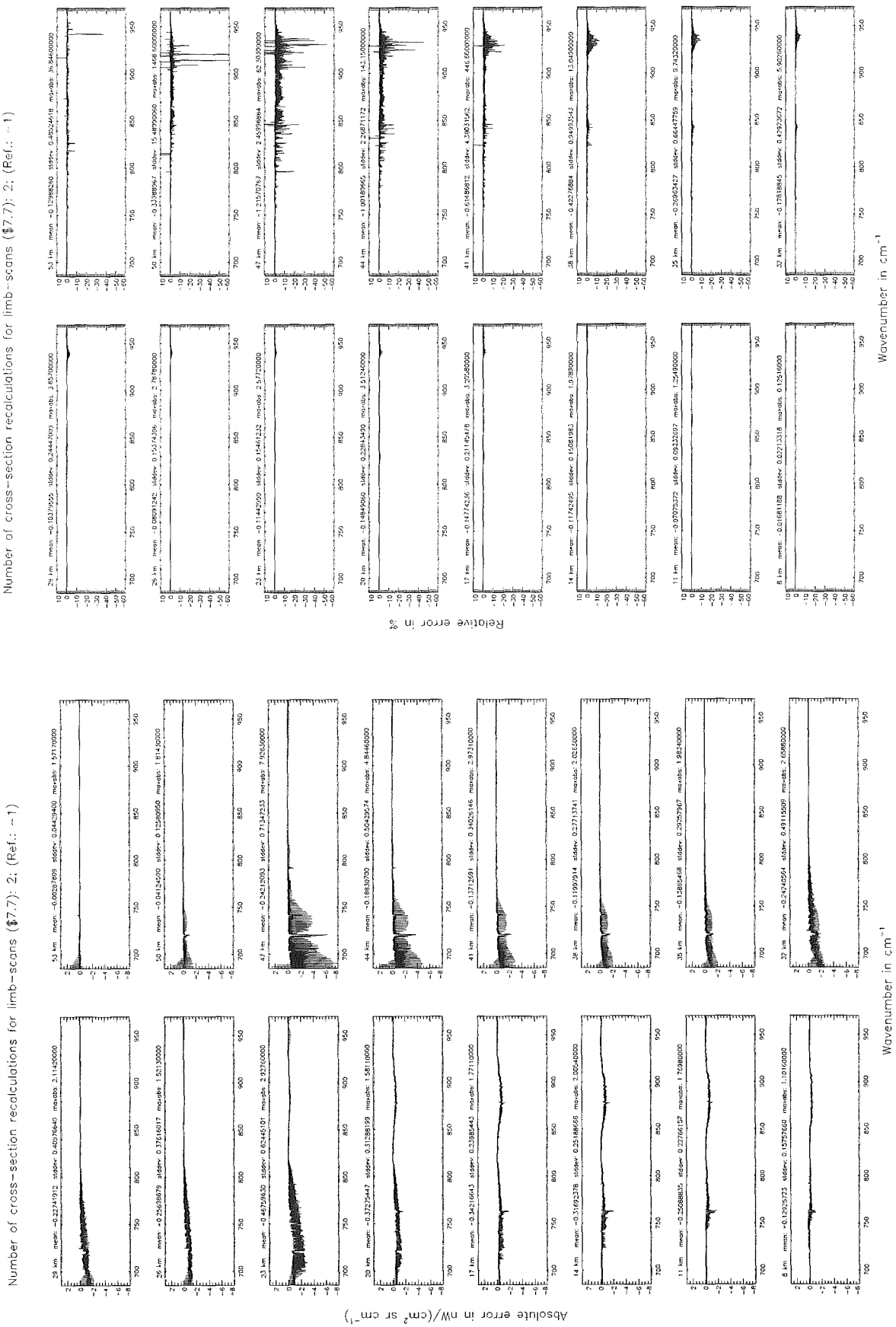


Figure 29: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

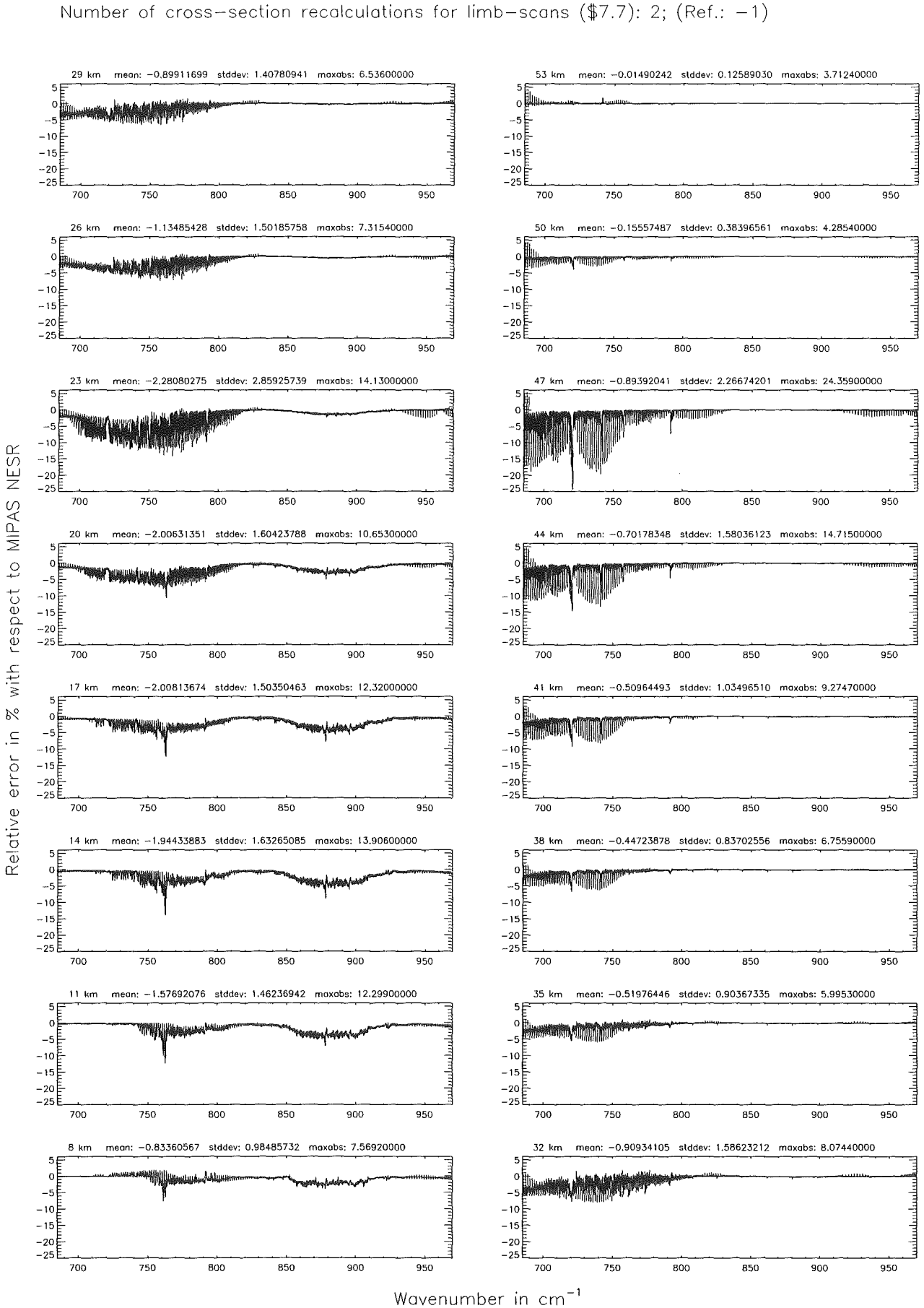


Figure 30: Relative error [%] with respect to MIPAS NESR

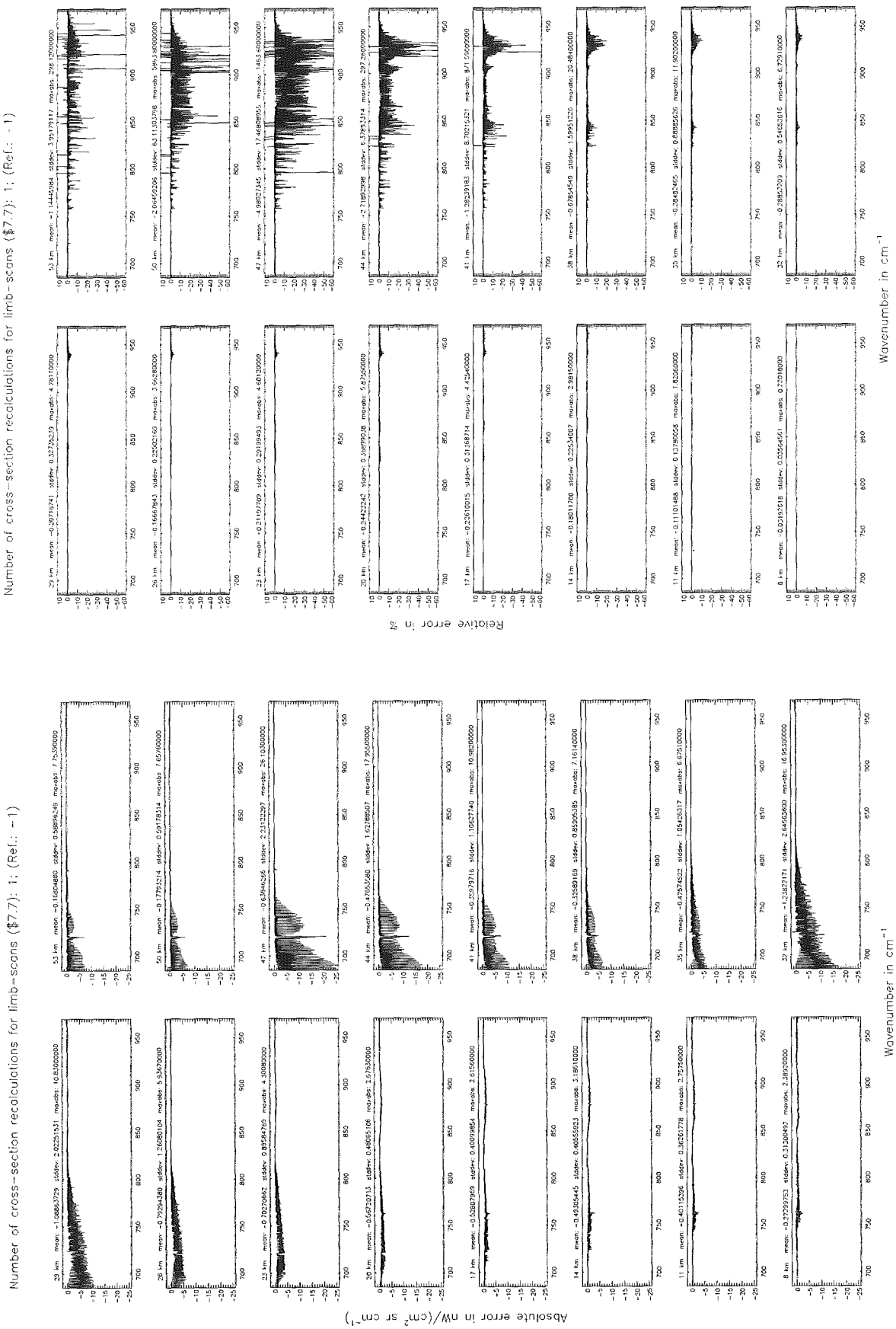


Figure 31: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

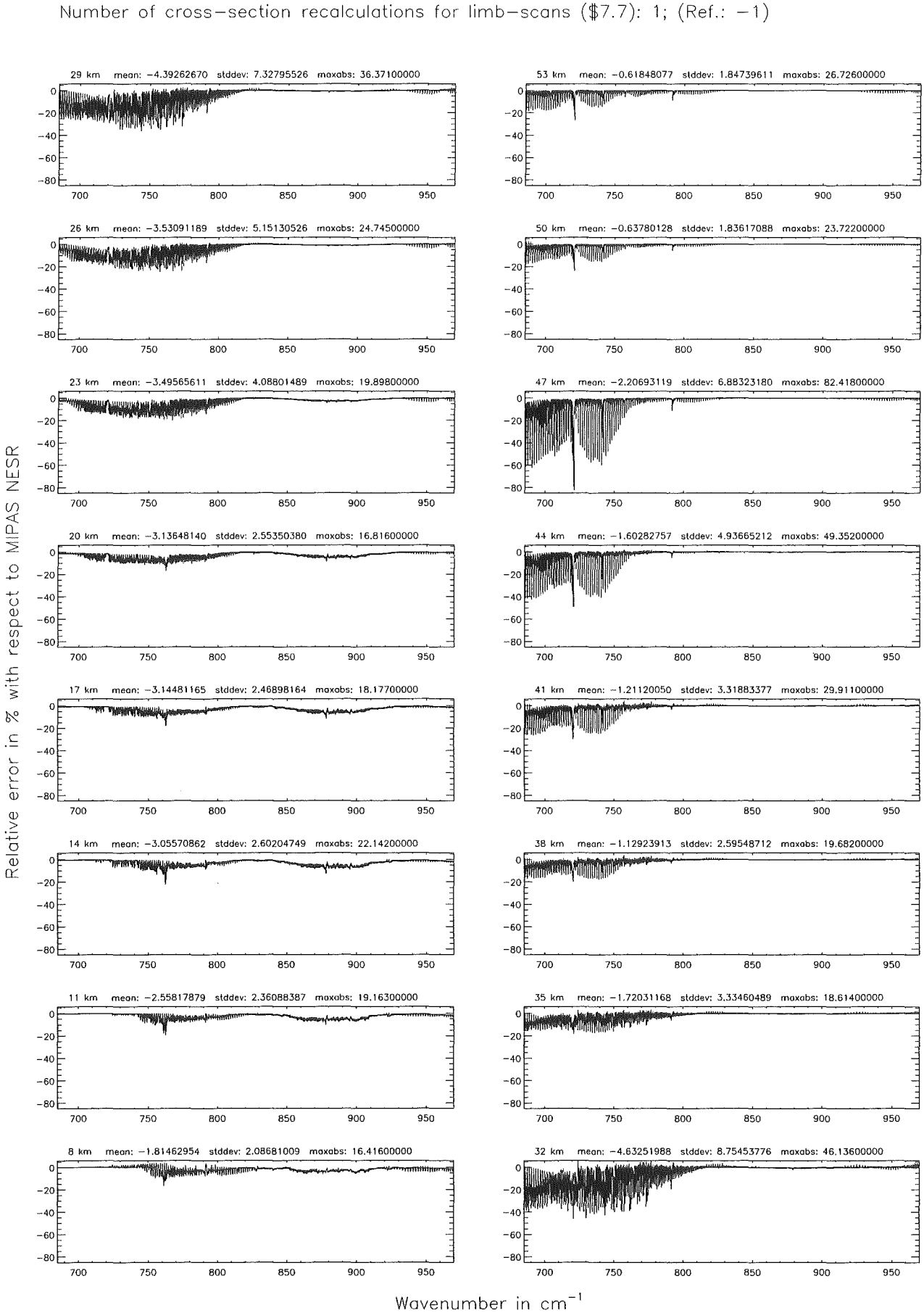


Figure 32: Relative error [%] with respect to MIPAS NESR

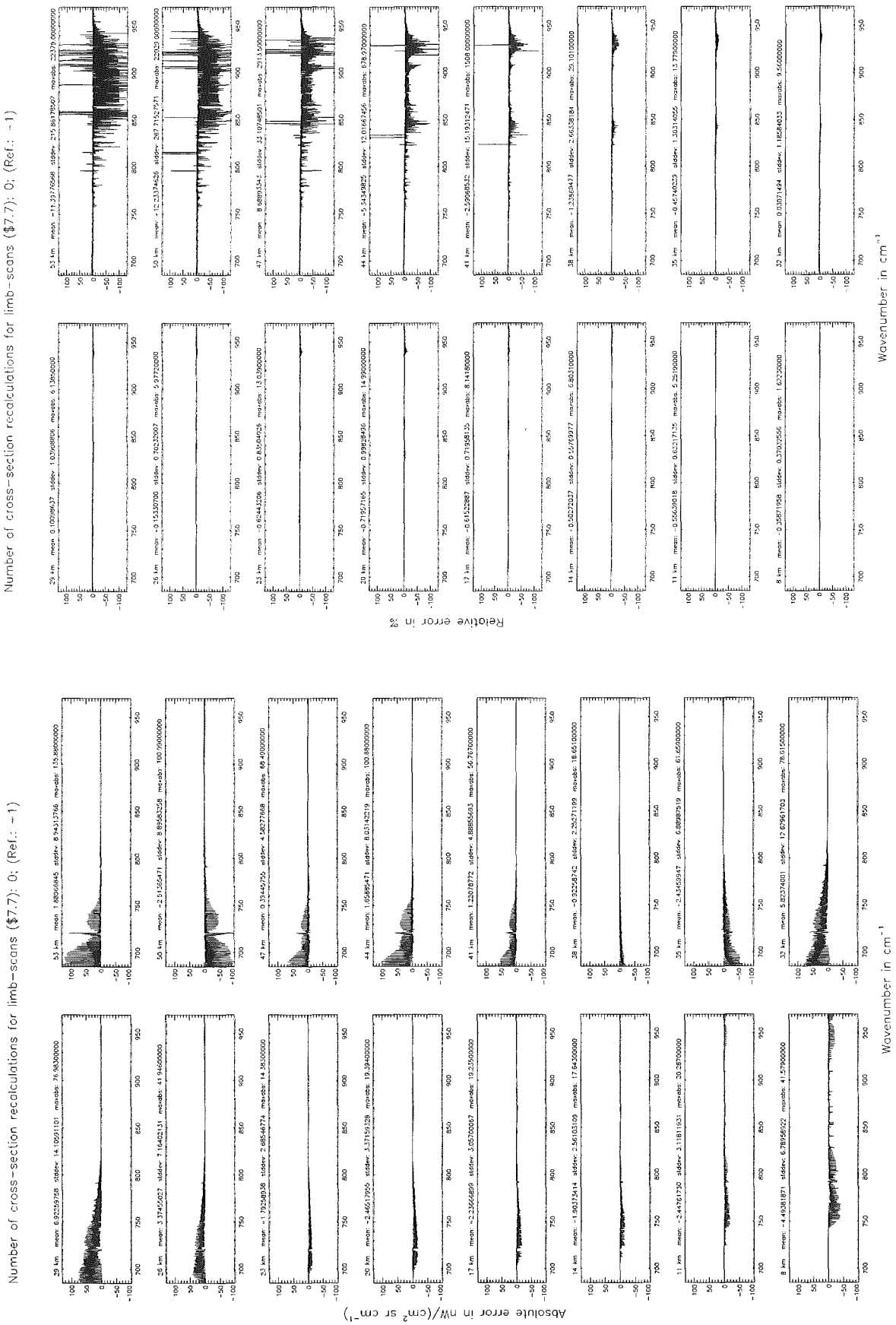


Figure 33: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

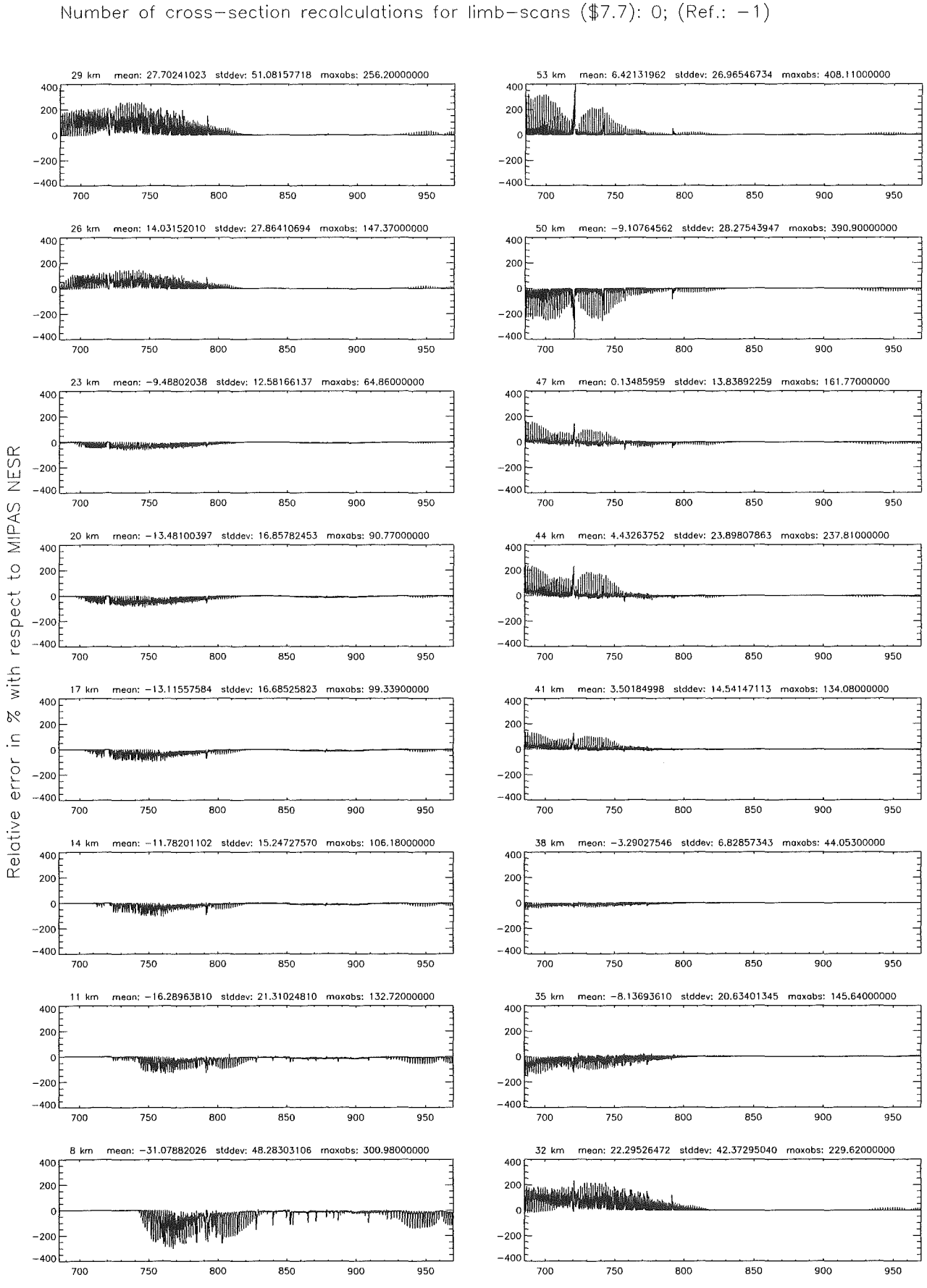


Figure 34: Relative error [%] with respect to MIPAS NESR

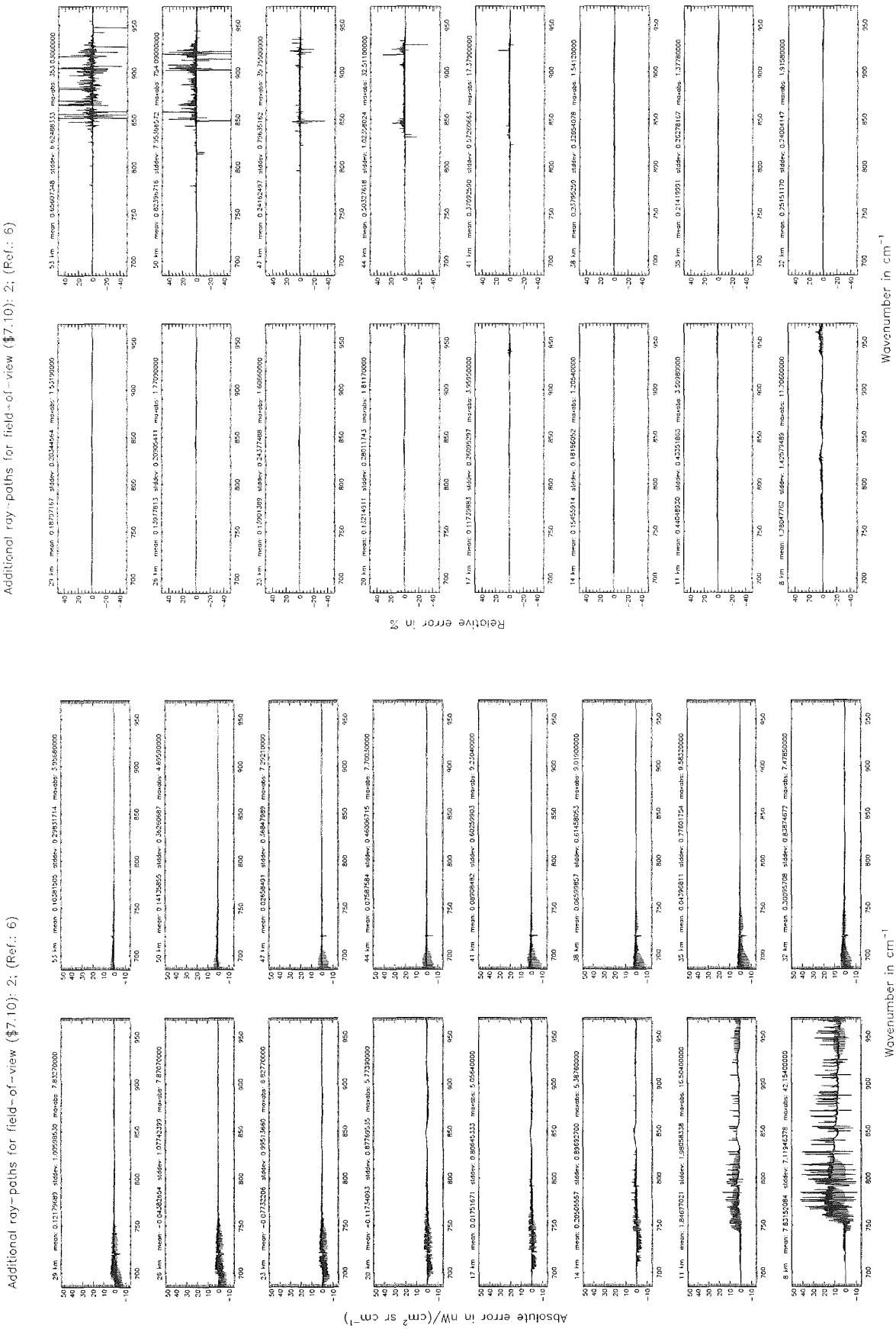


Figure 35: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

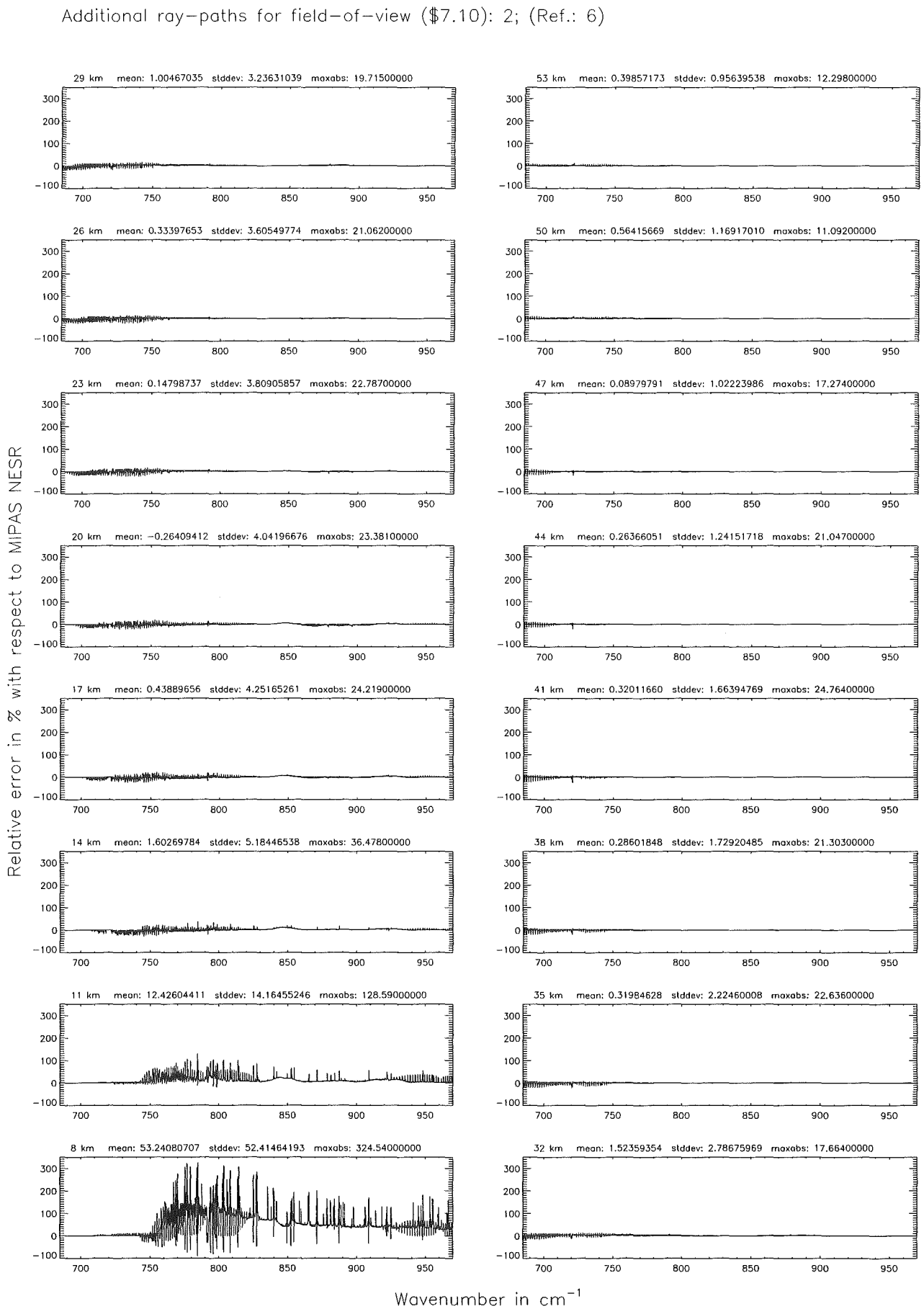


Figure 36: Relative error [%] with respect to MIPAS NESR

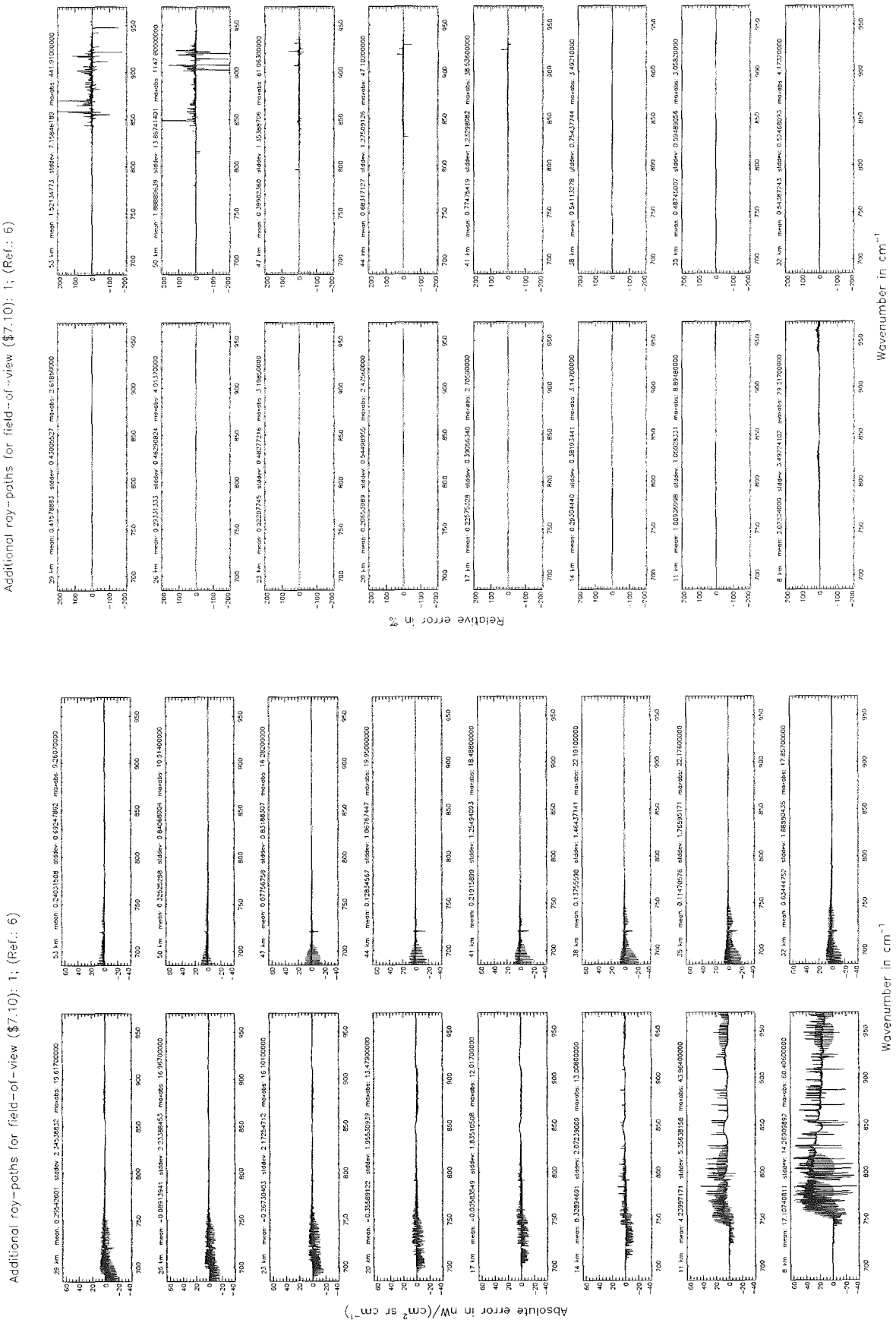


Figure 37: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

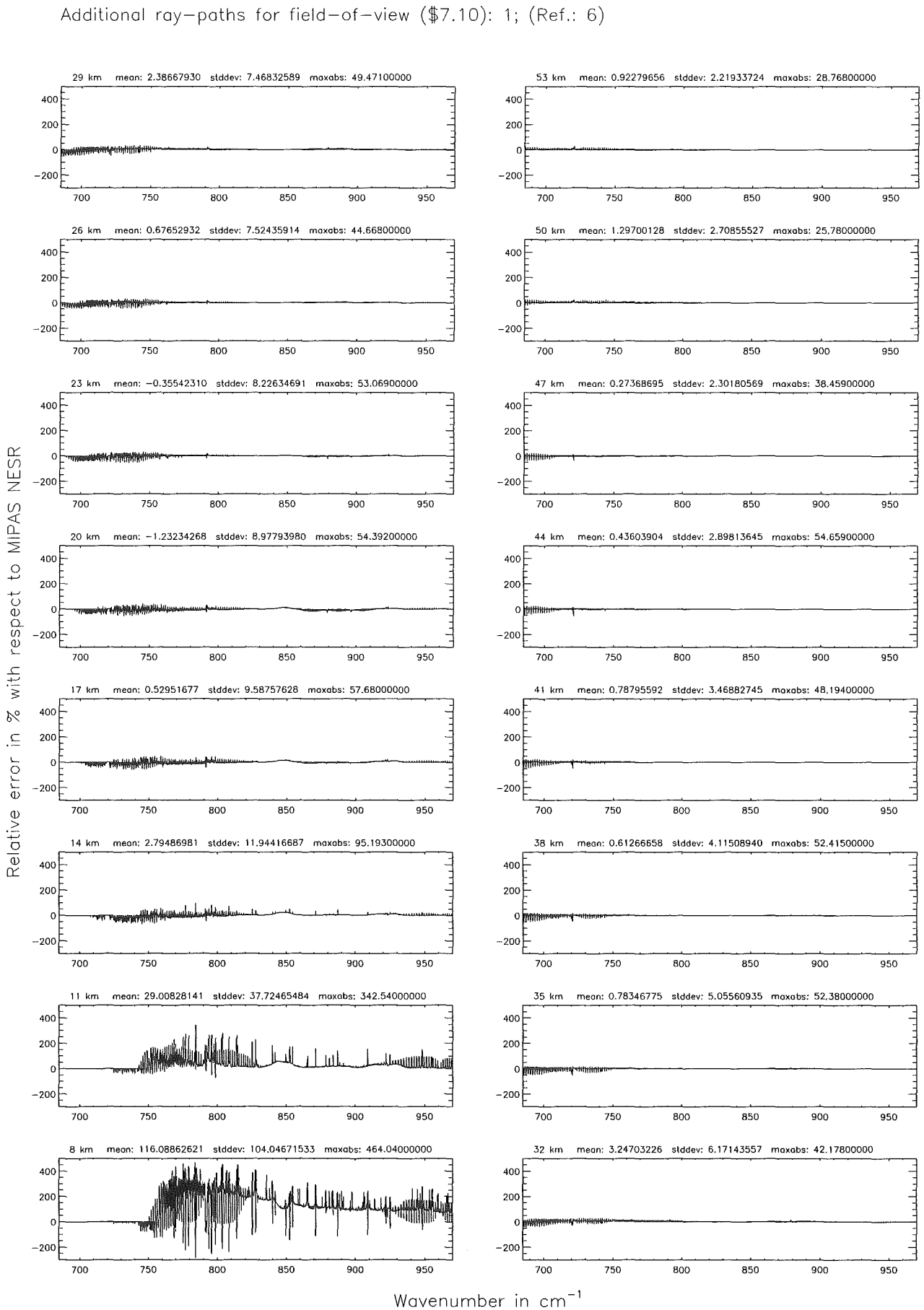


Figure 38: Relative error [%] with respect to MIPAS NESR

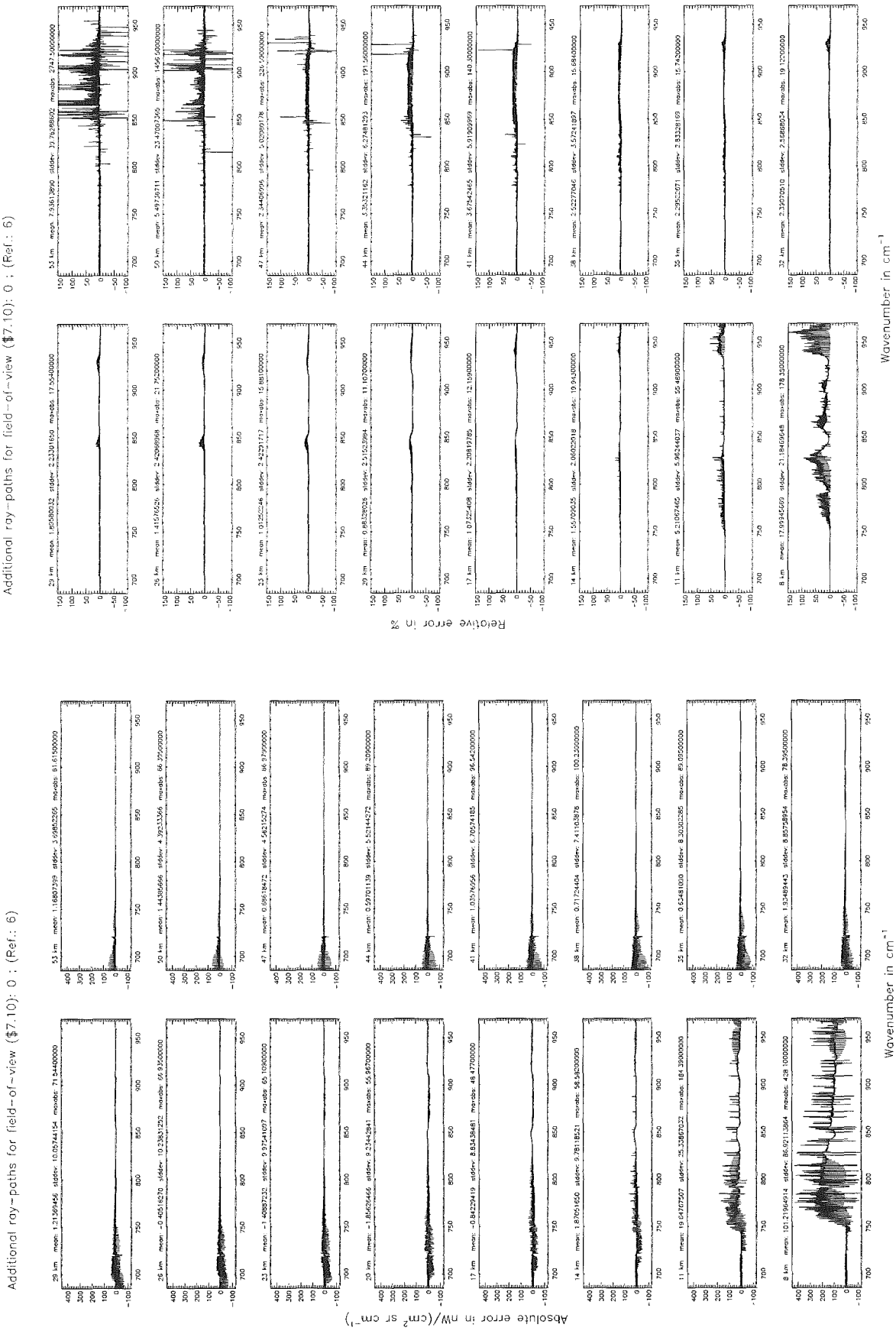


Figure 39: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

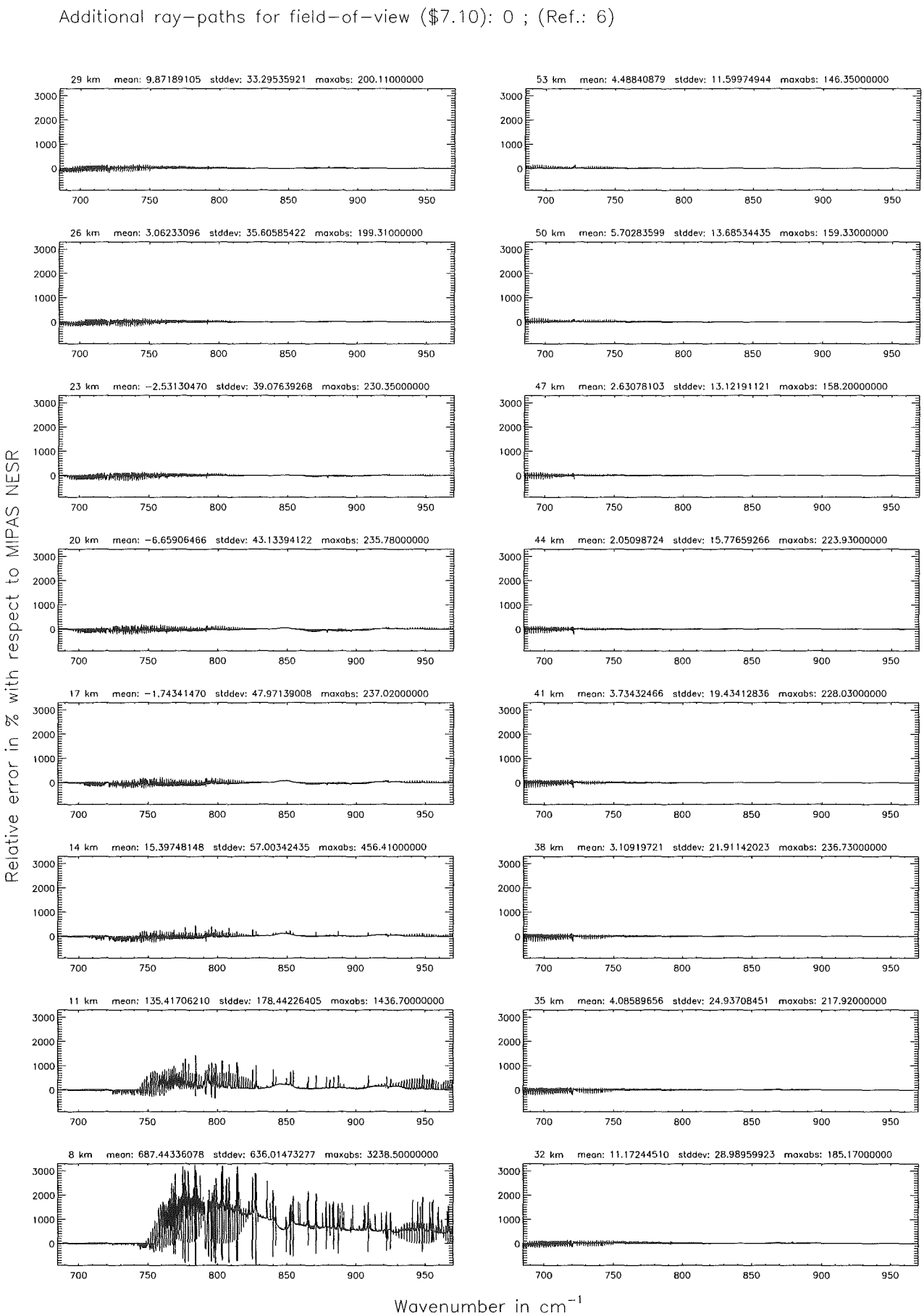


Figure 40: Relative error [%] with respect to MIPAS NESR

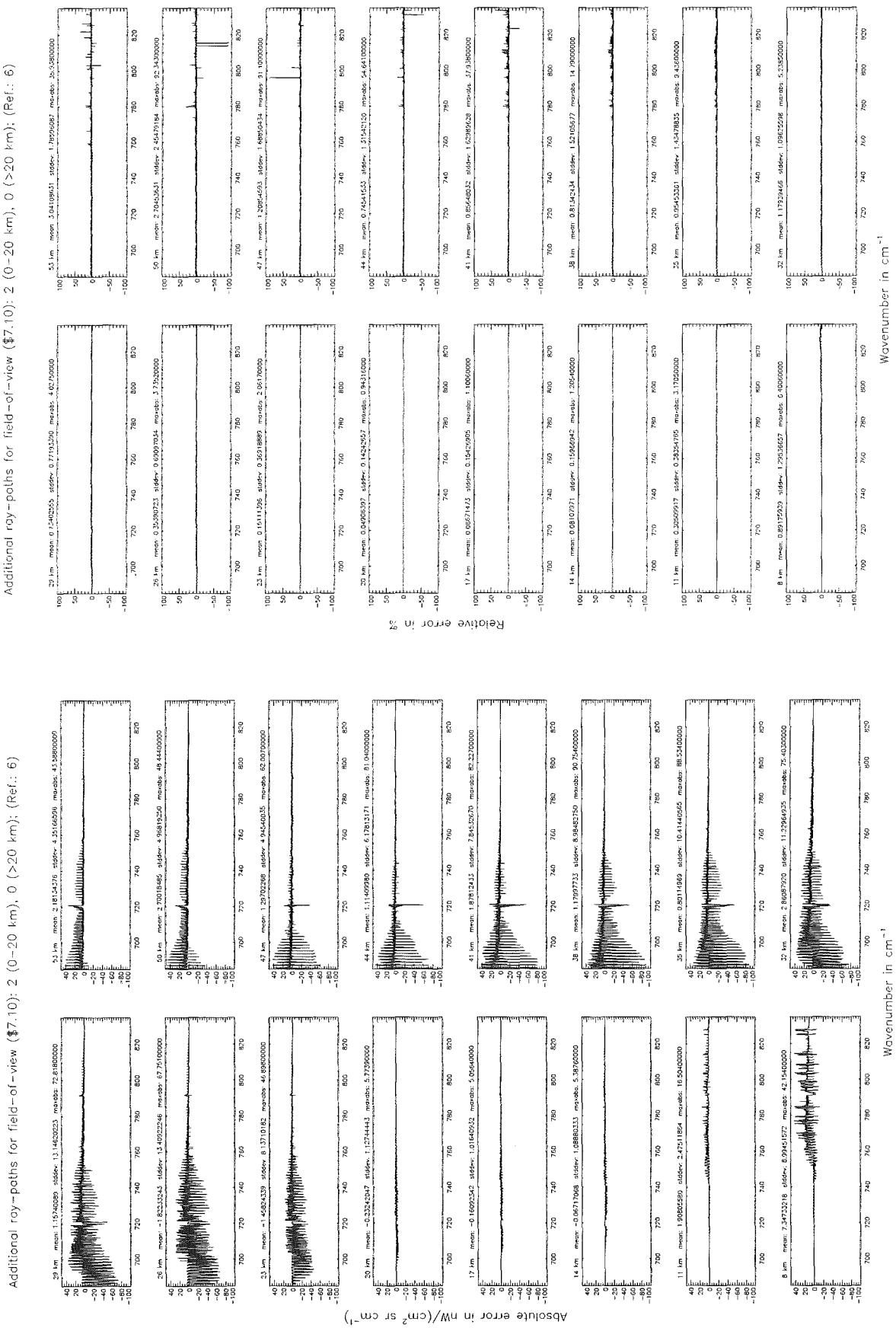


Figure 41: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

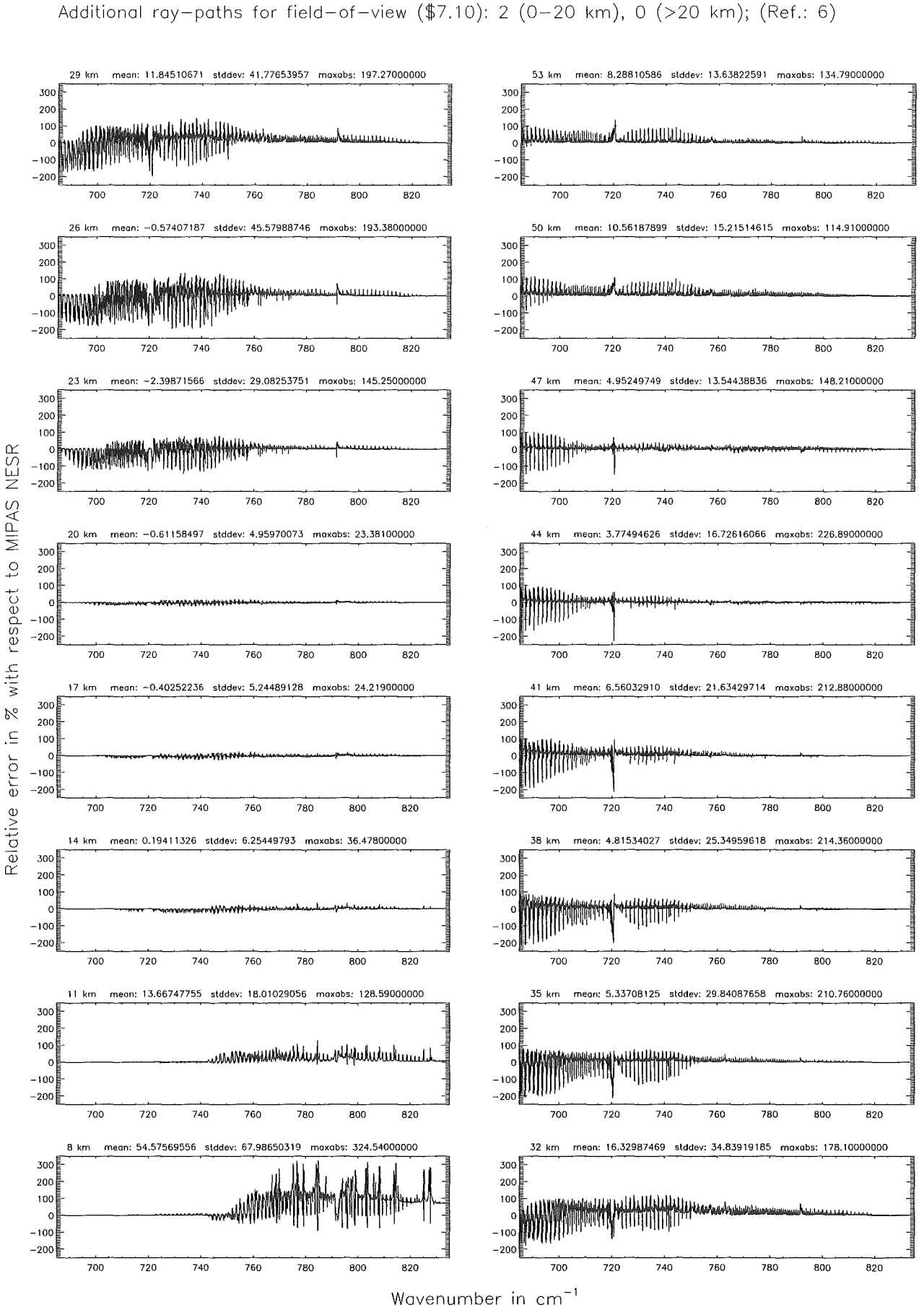


Figure 42: Relative error [%] with respect to MIPAS NESR

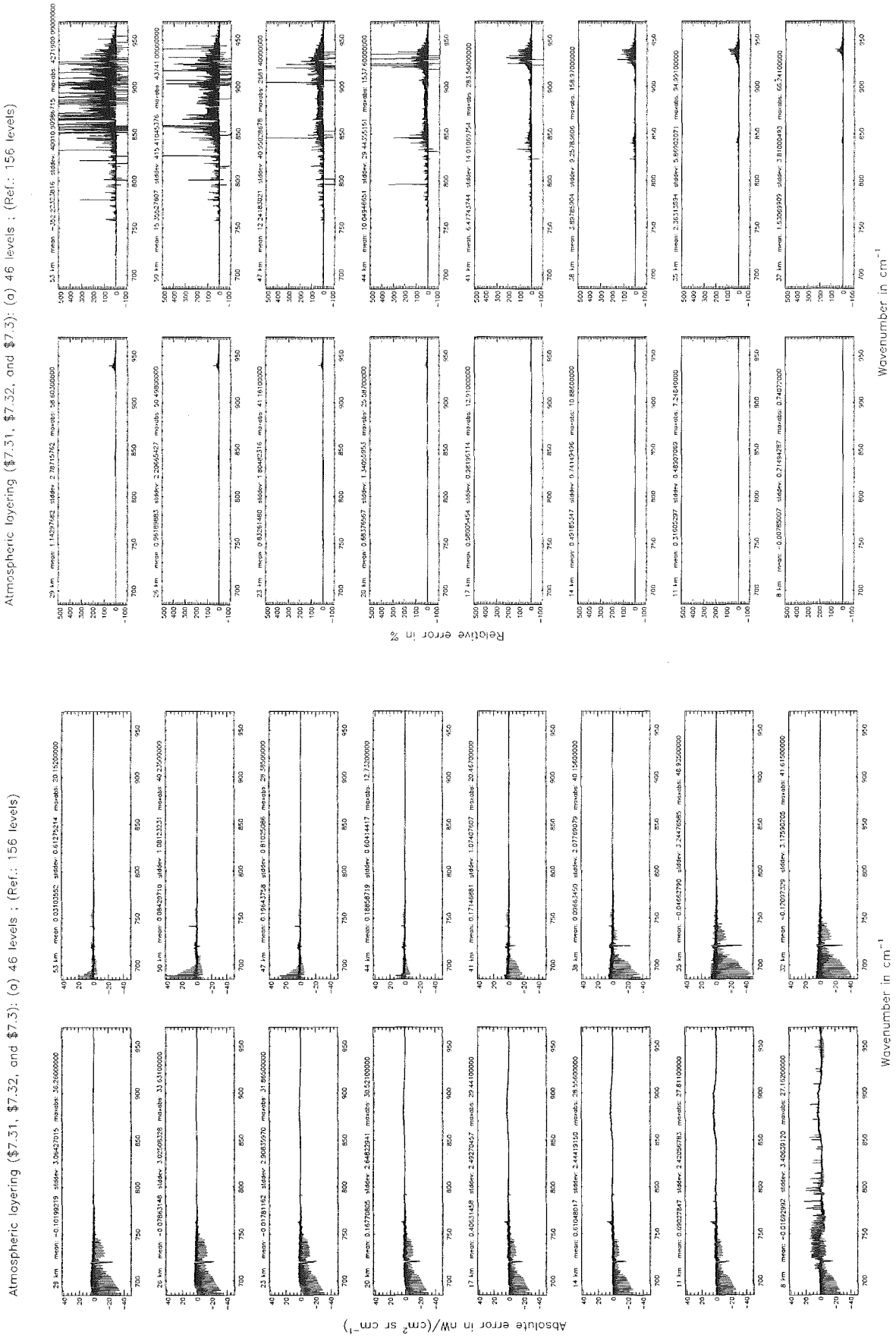


Figure 43: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

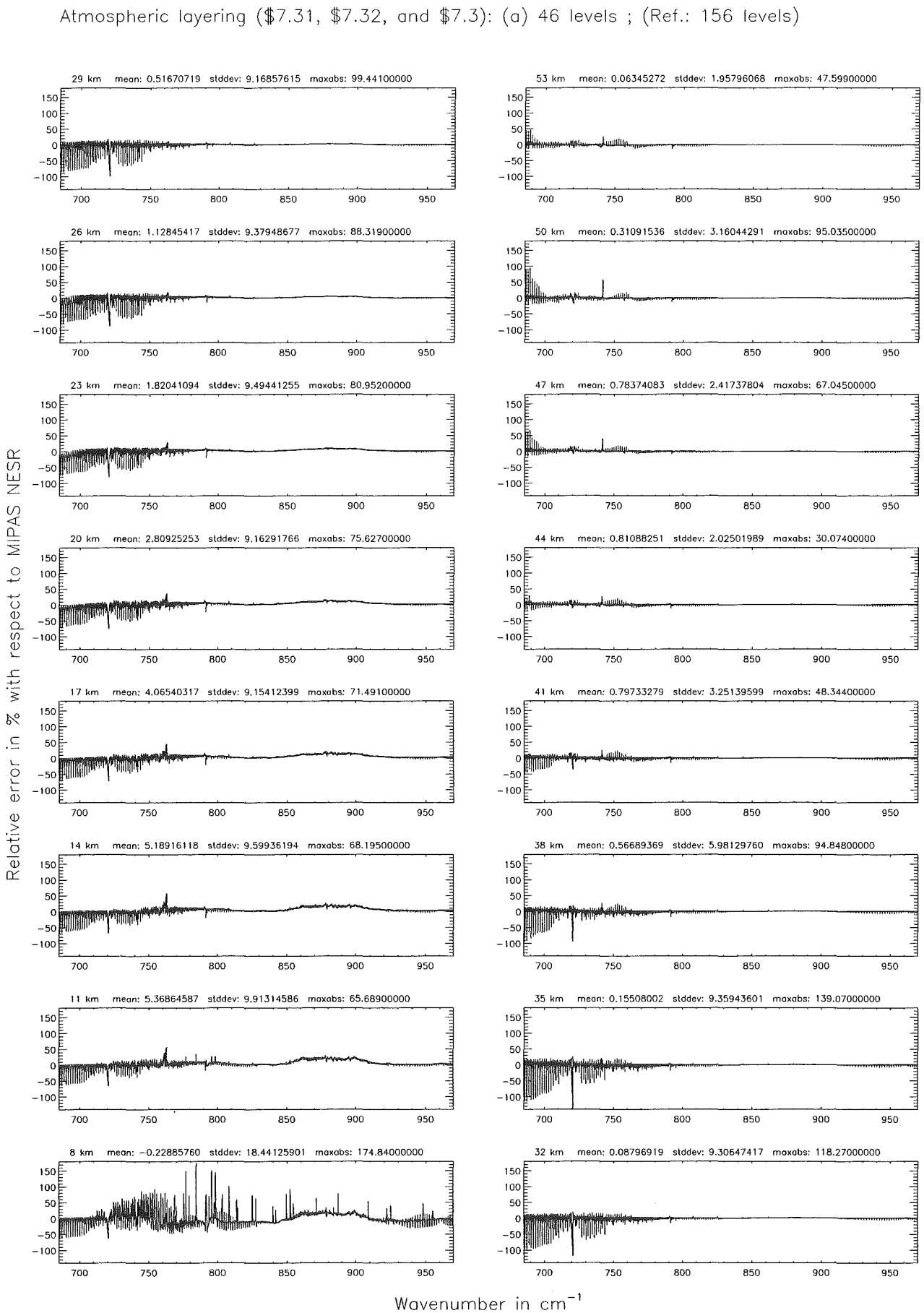


Figure 44: Relative error [%] with respect to MIPAS NESR

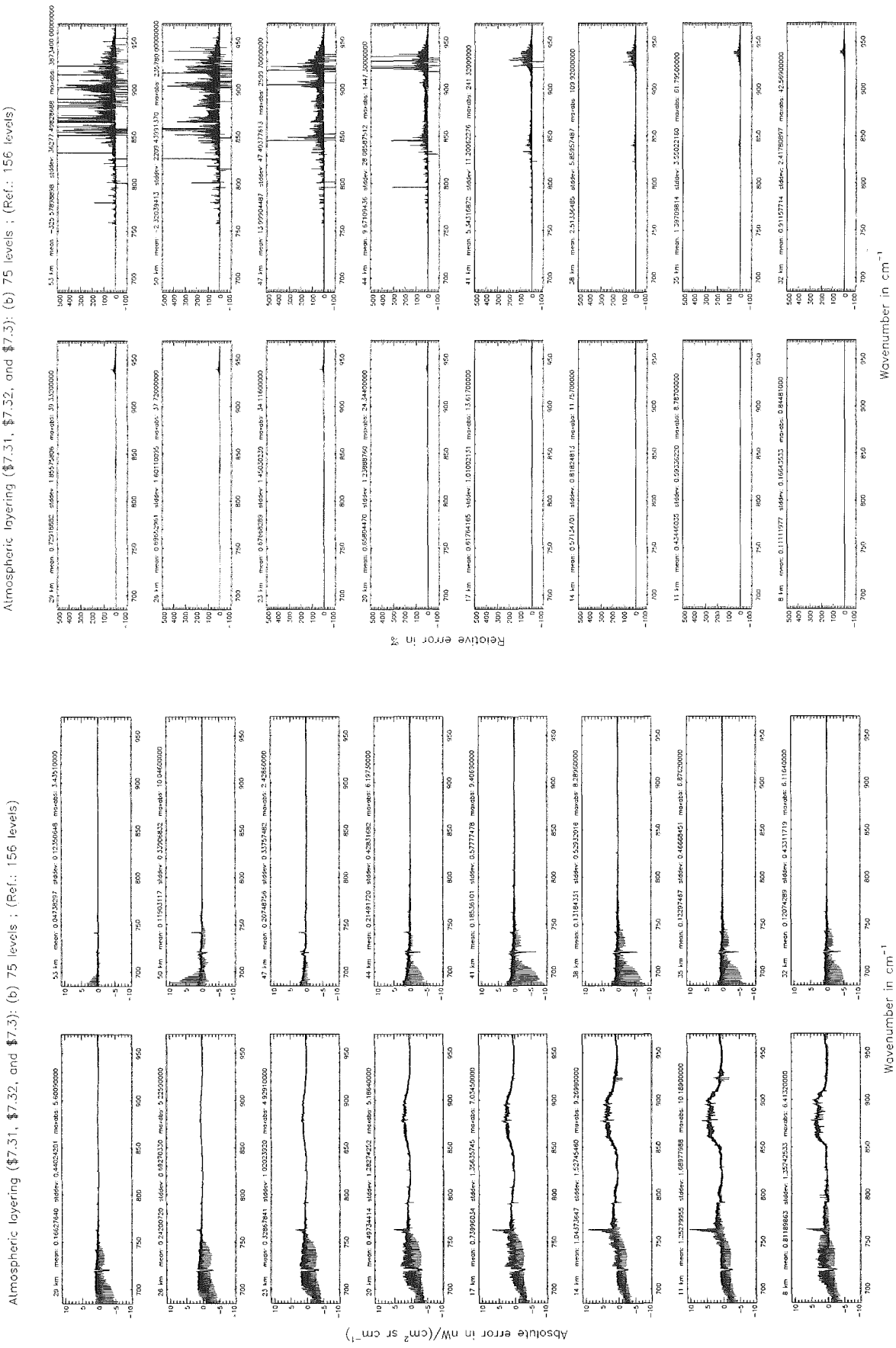


Figure 45: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

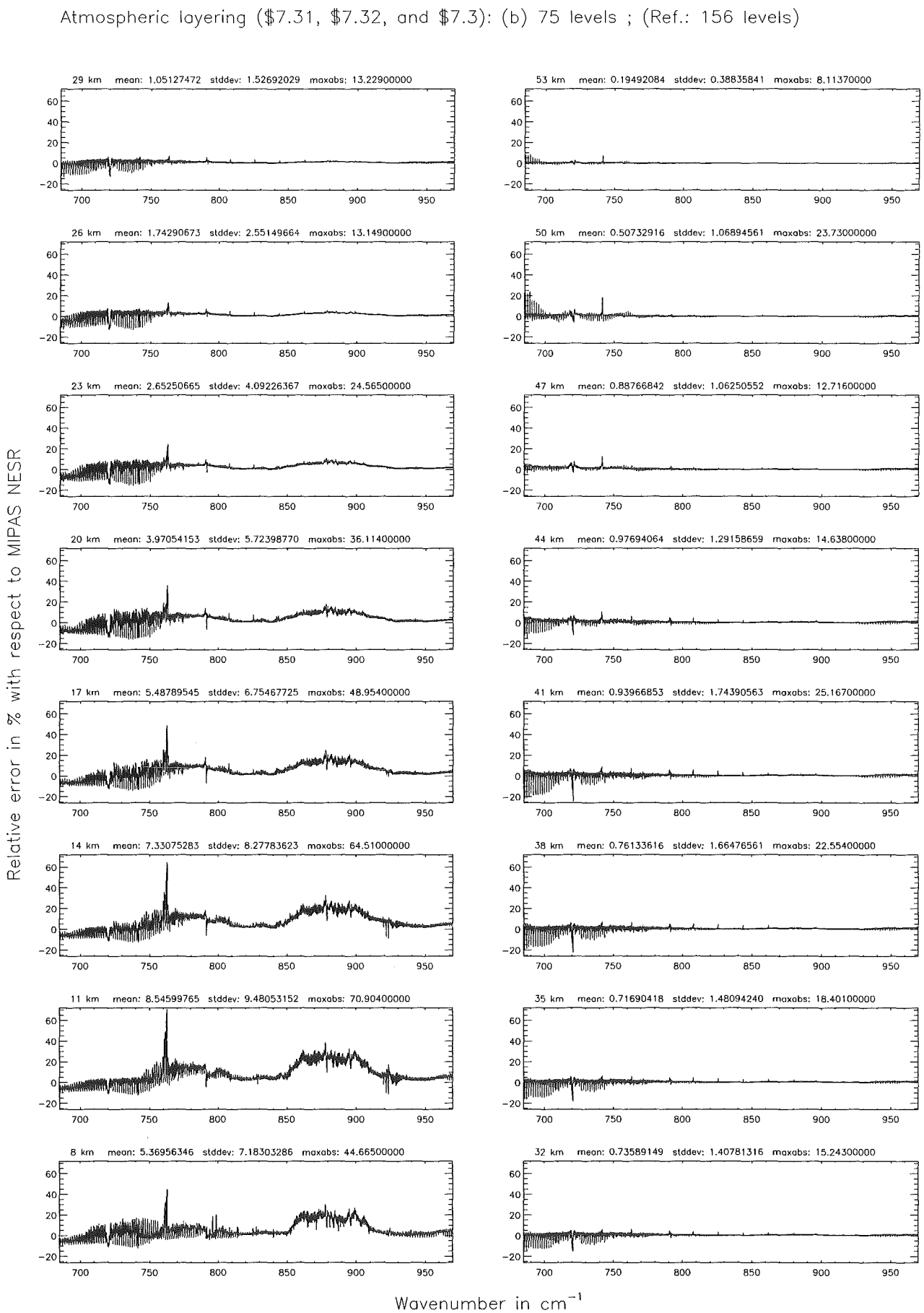


Figure 46: Relative error [%] with respect to MIPAS NESR

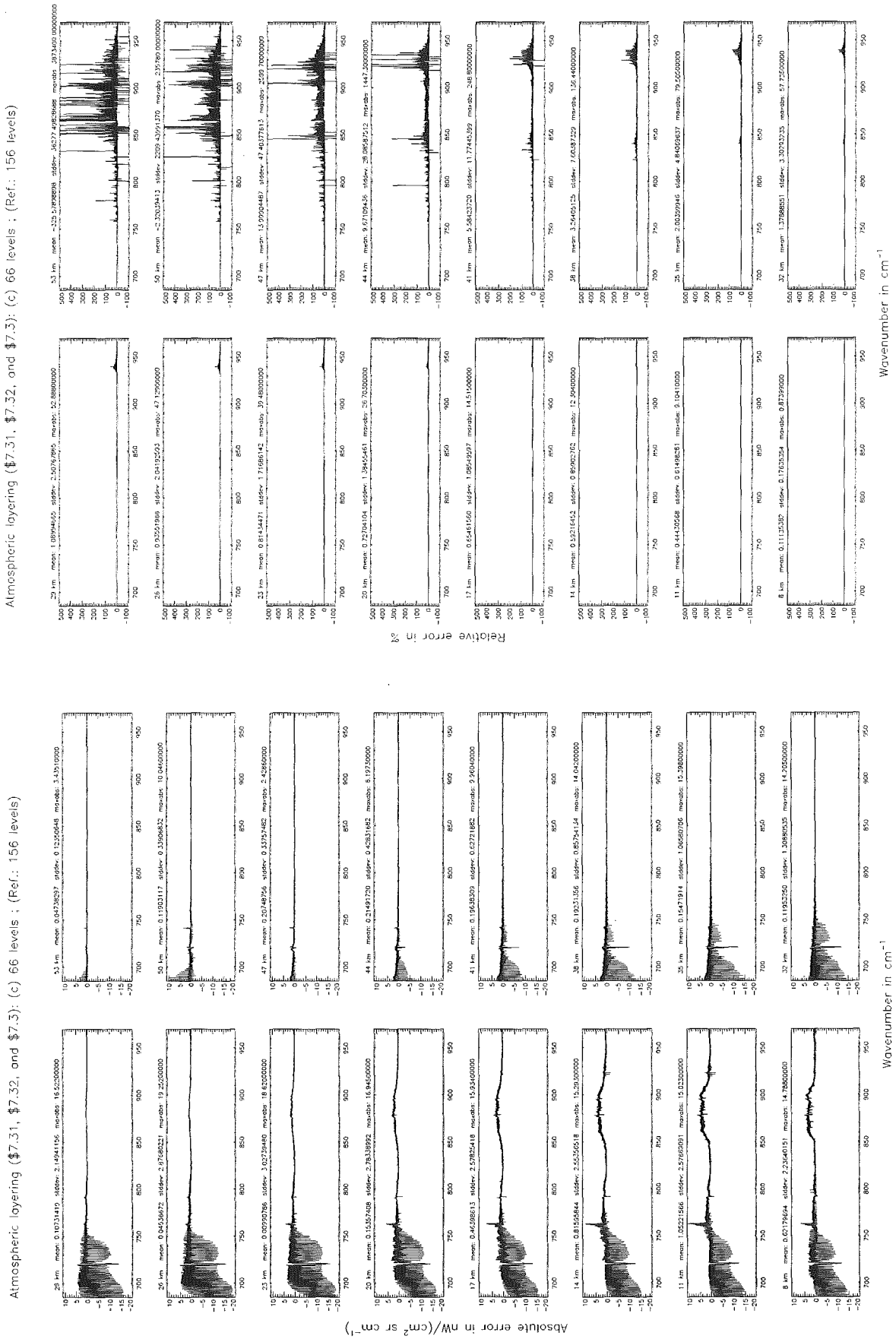


Figure 47: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

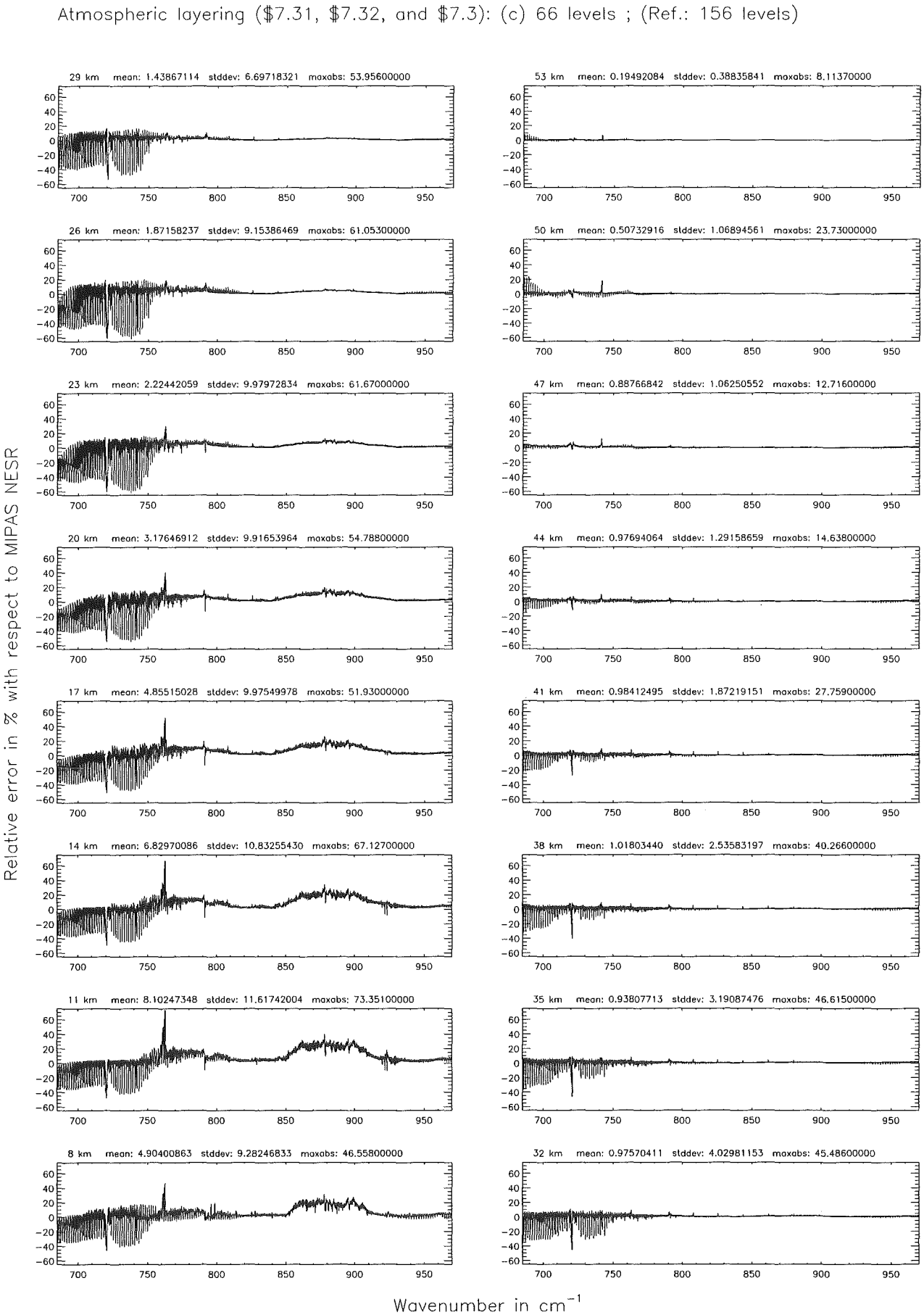


Figure 48: Relative error [%] with respect to MIPAS NESR

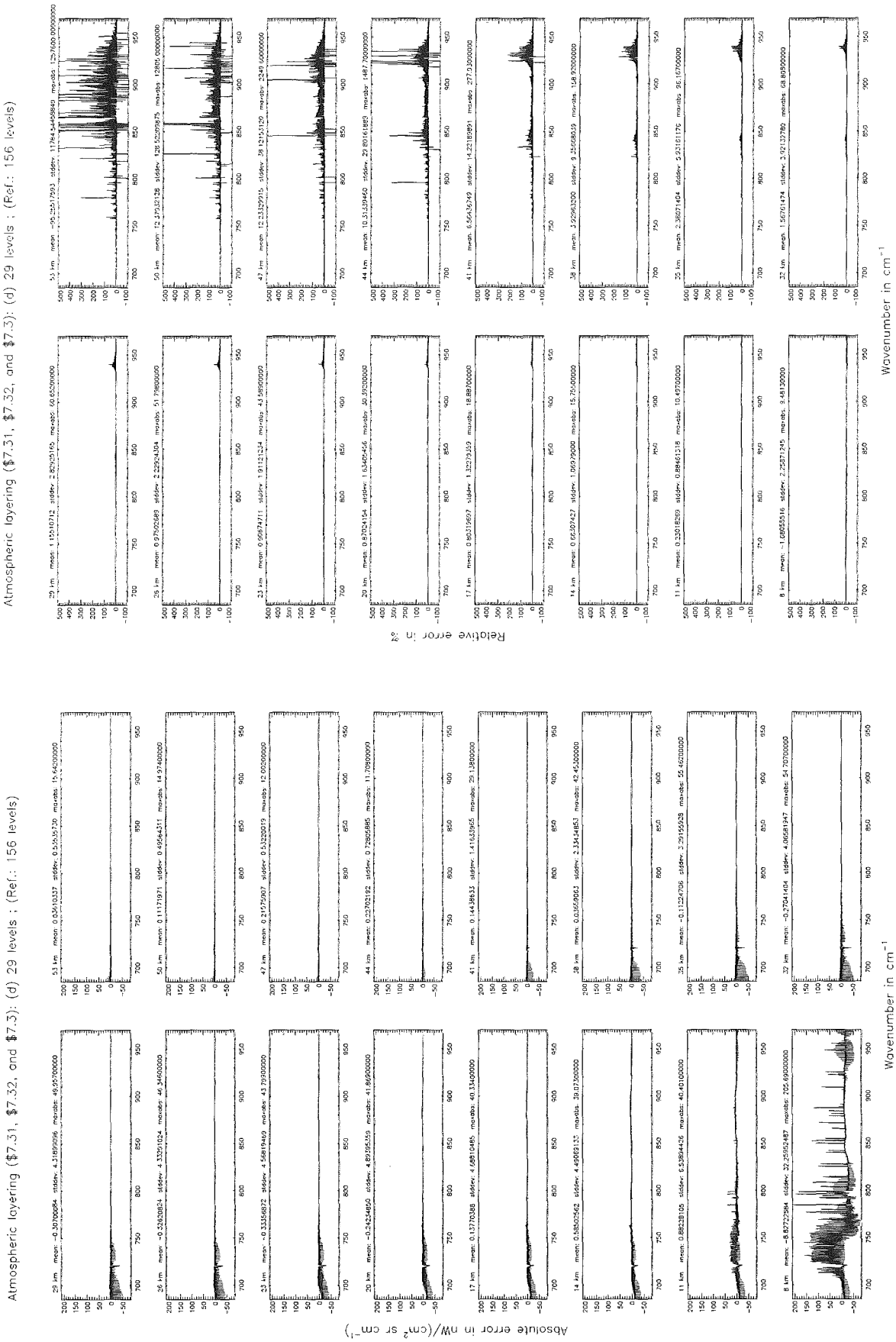


Figure 49: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

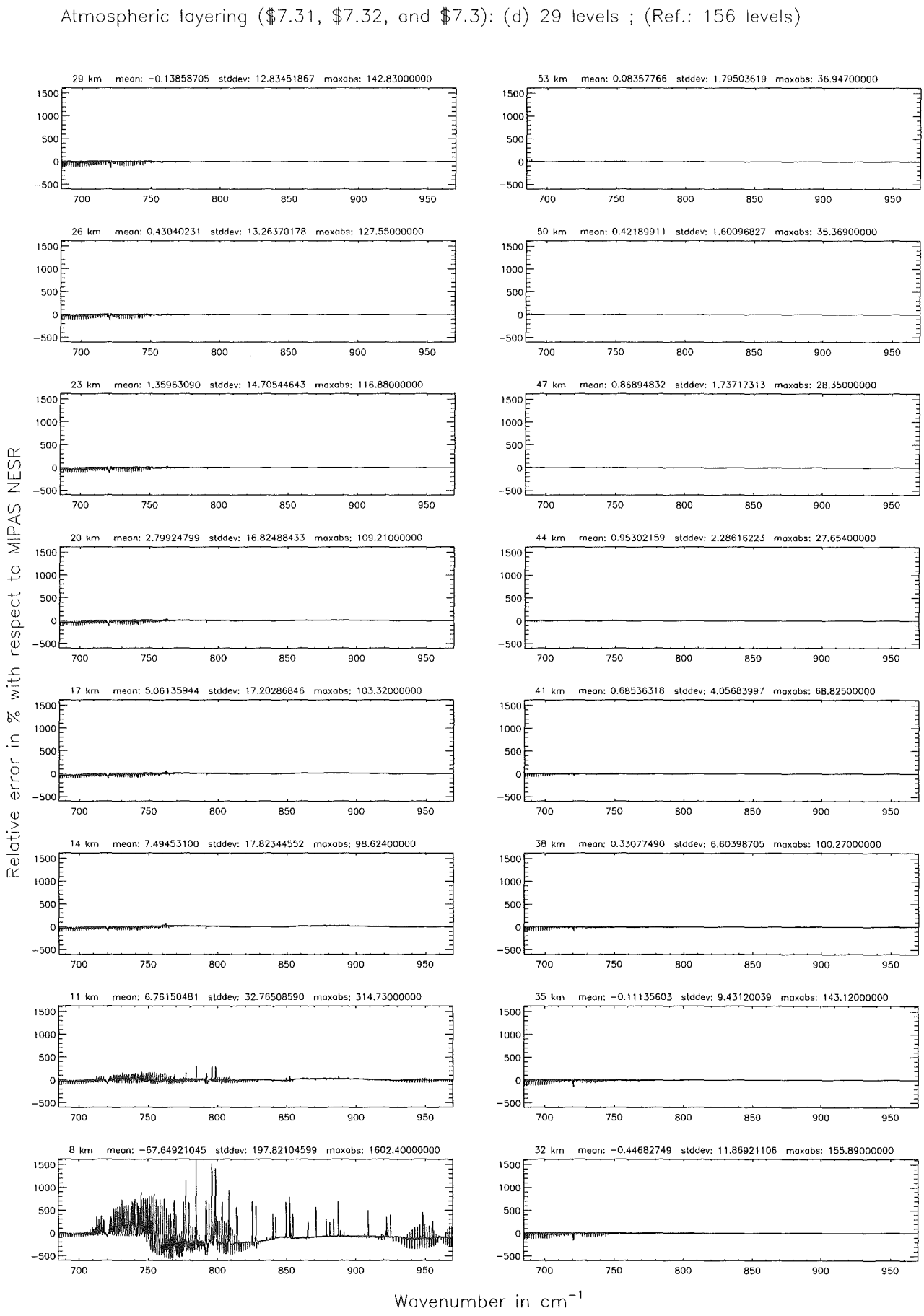


Figure 50: Relative error [%] with respect to MIPAS NESR

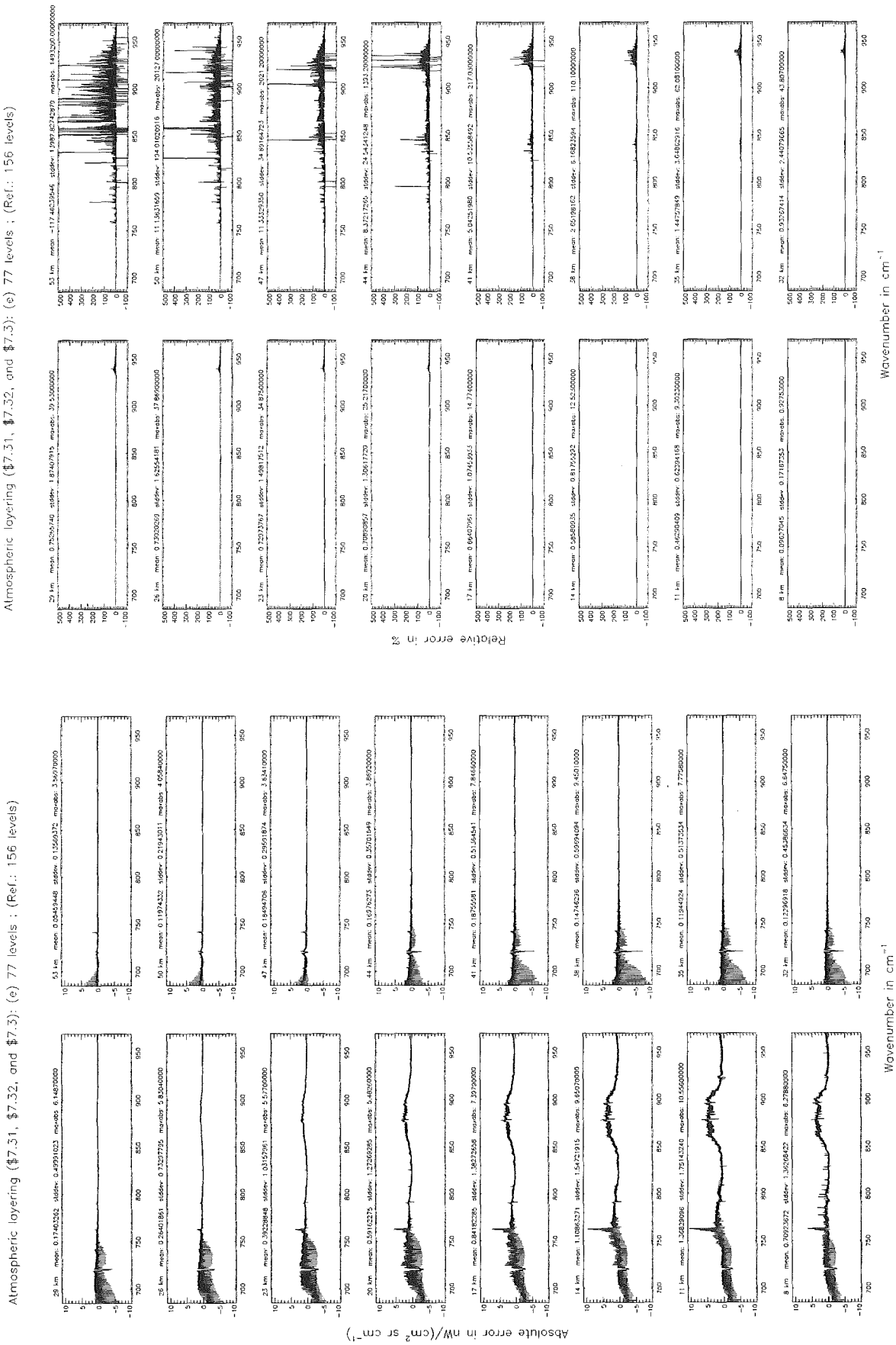


Figure 51: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

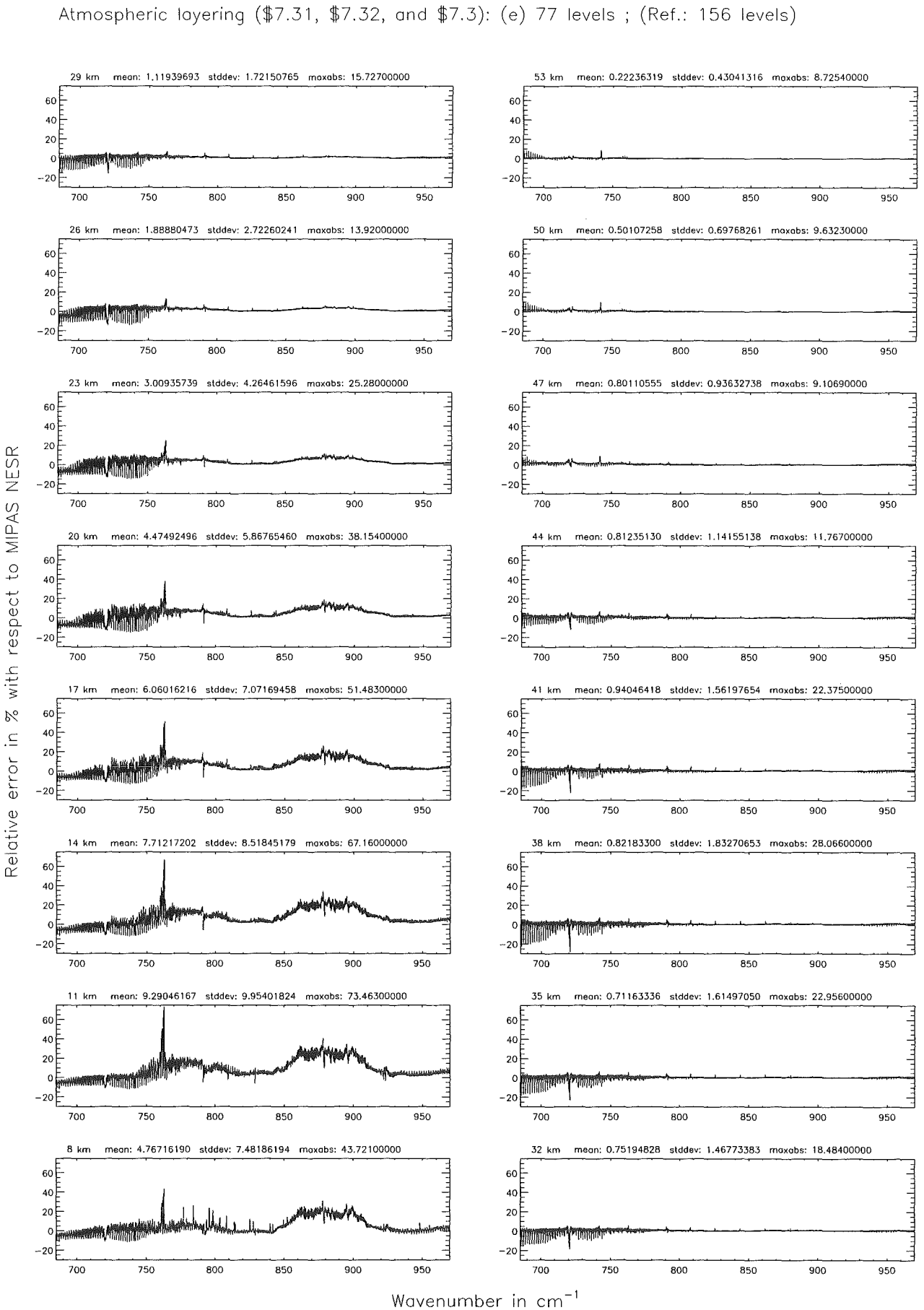


Figure 52: Relative error [%] with respect to MIPAS NESR

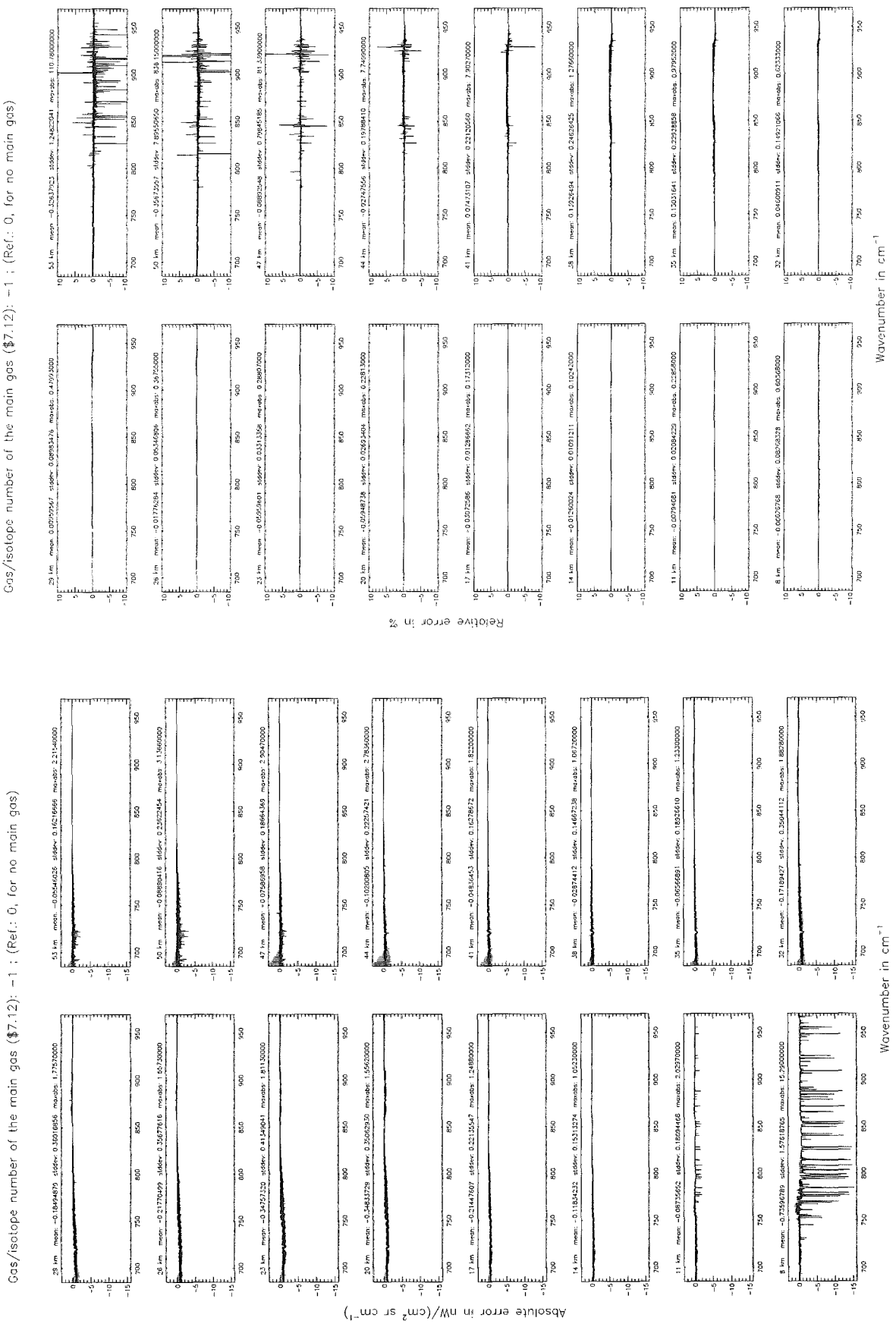


Figure 53: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

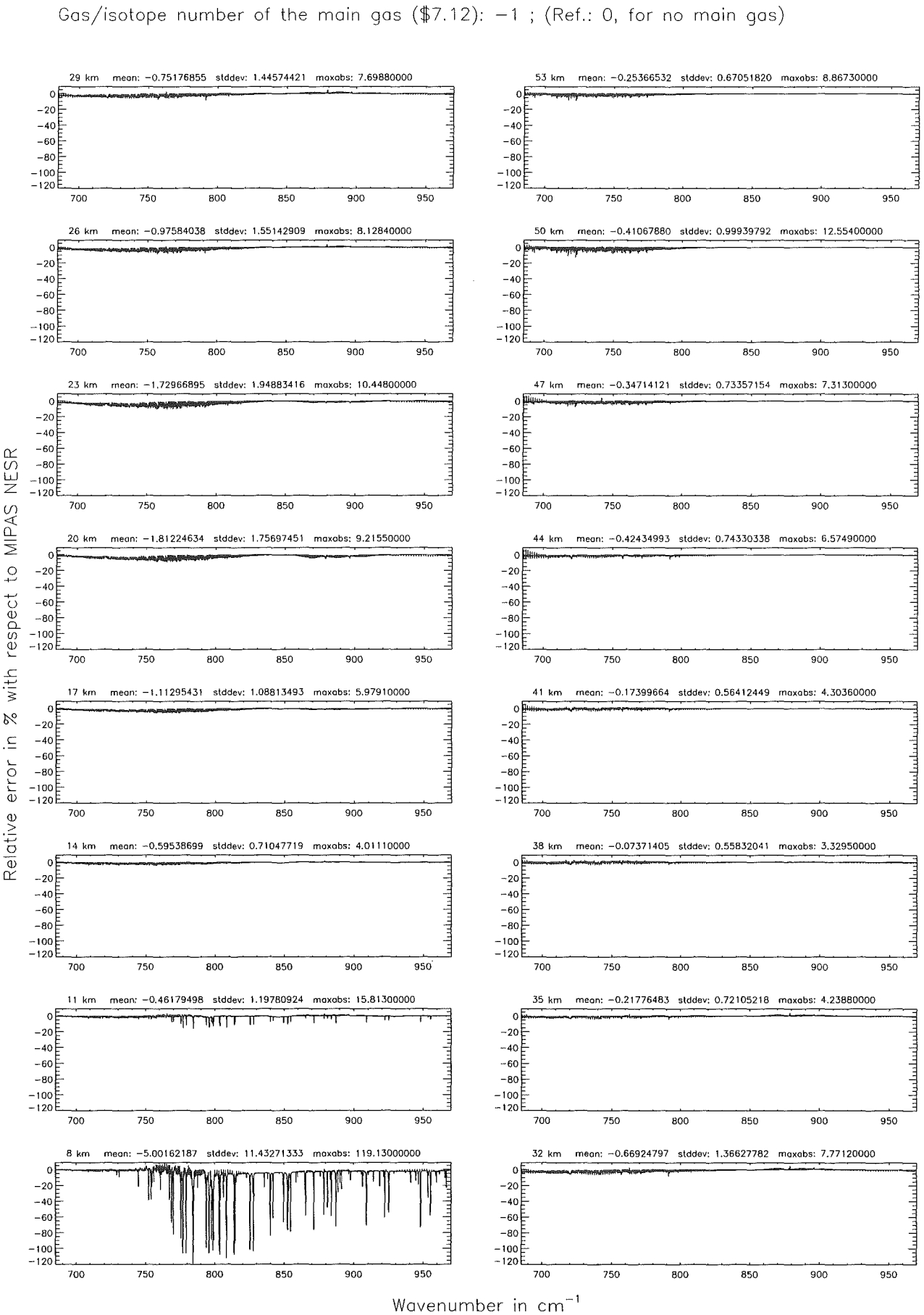


Figure 54: Relative error [%] with respect to MIPAS NESR

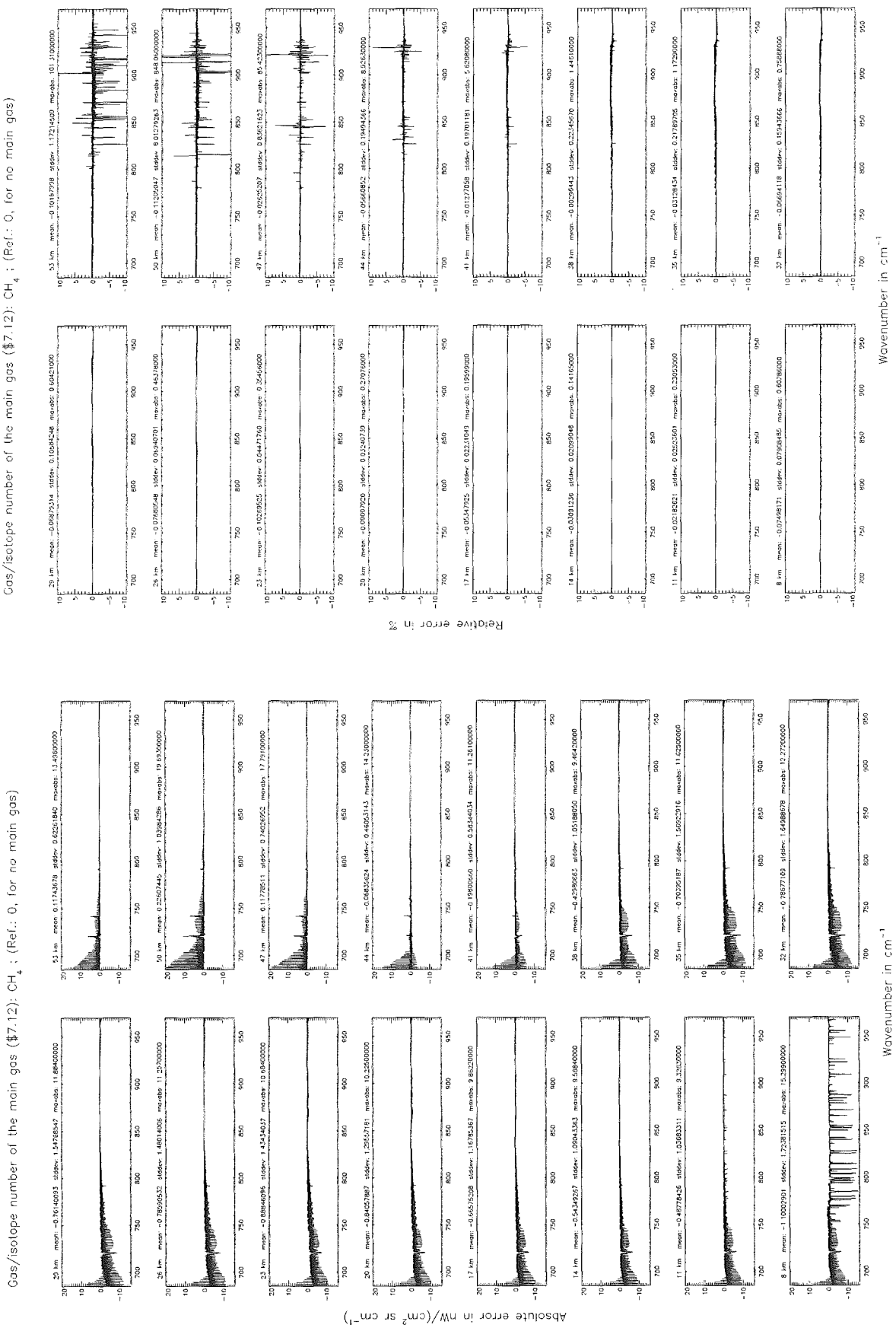


Figure 55: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

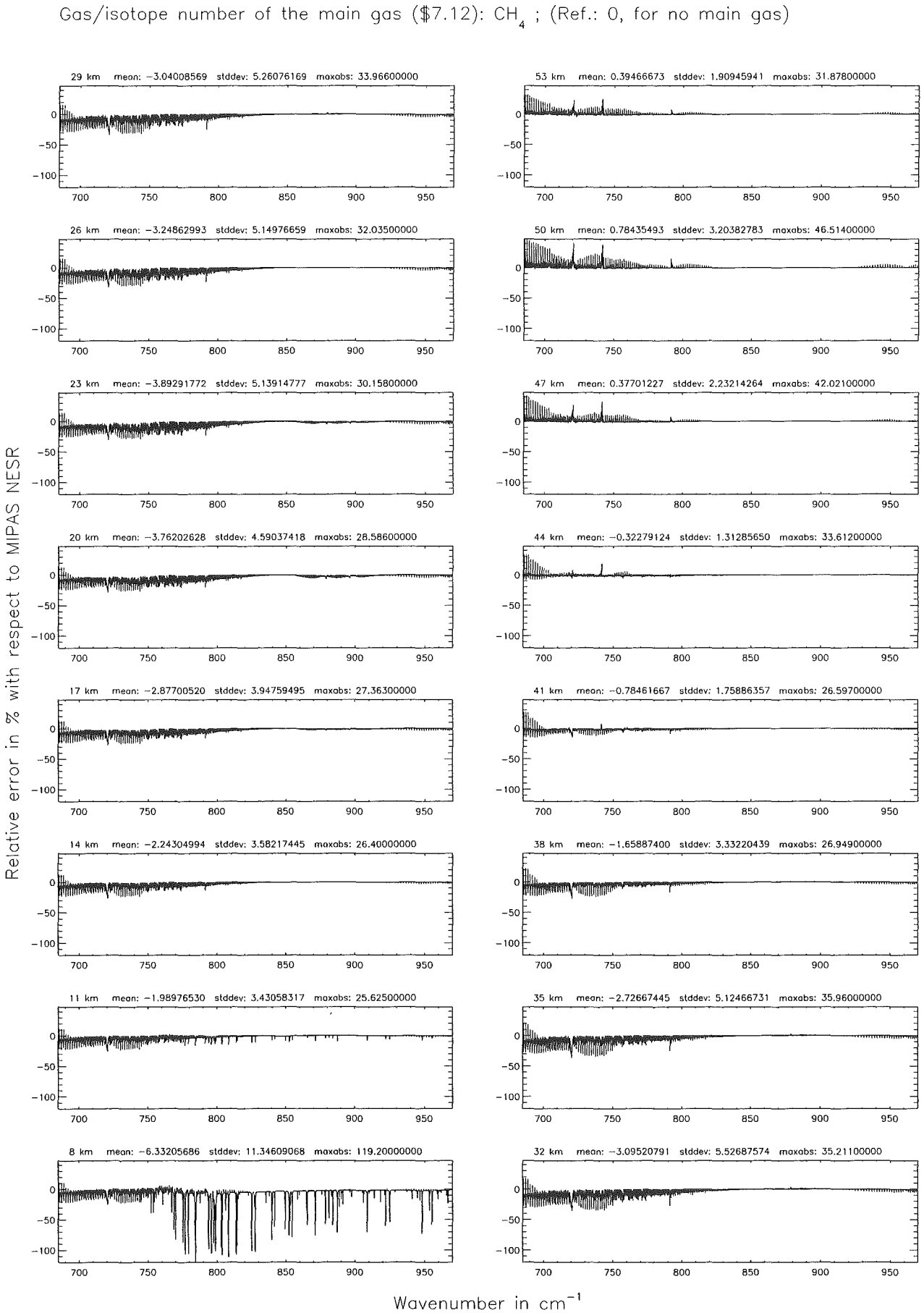


Figure 56: Relative error [%] with respect to MIPAS NESR

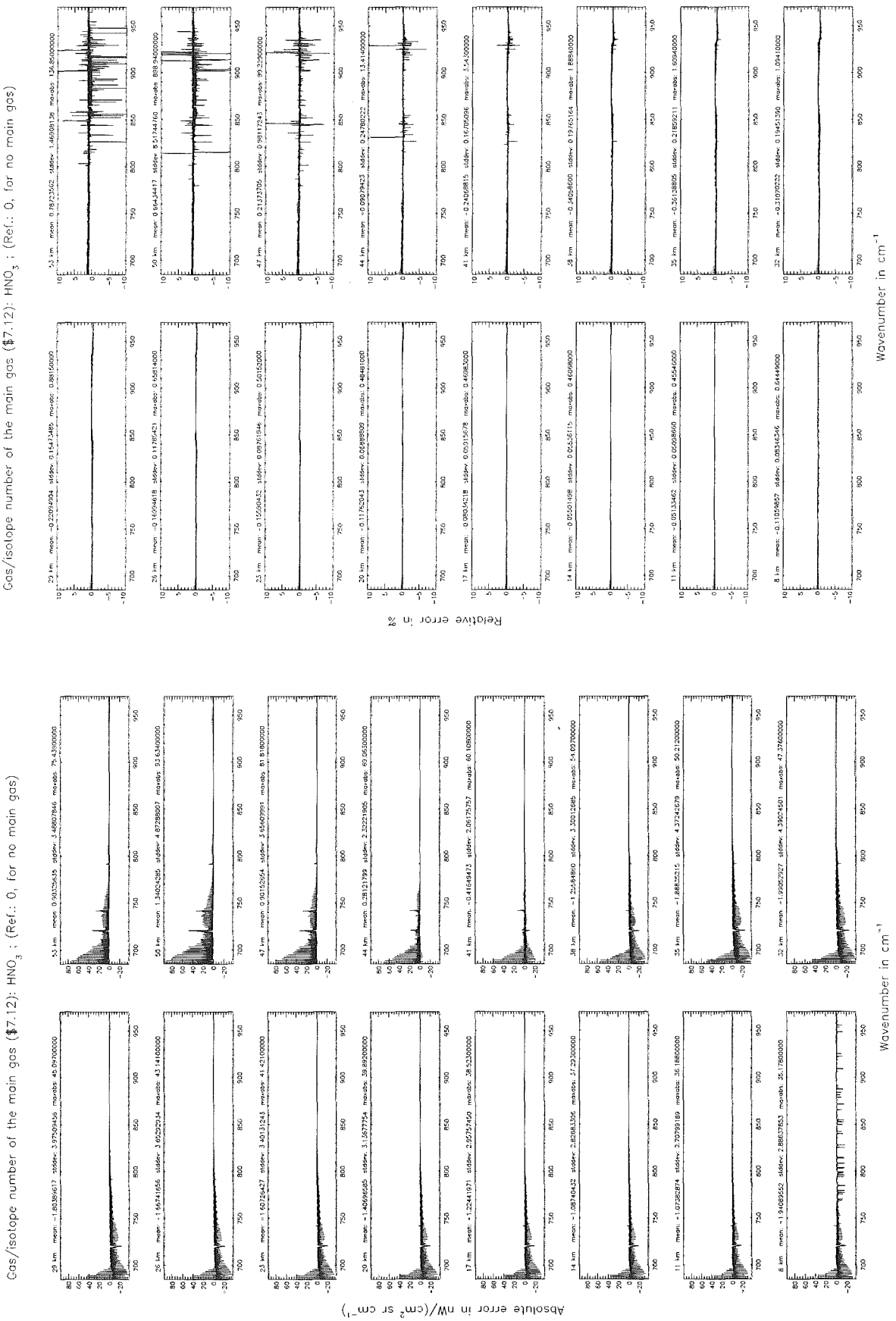


Figure 57: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

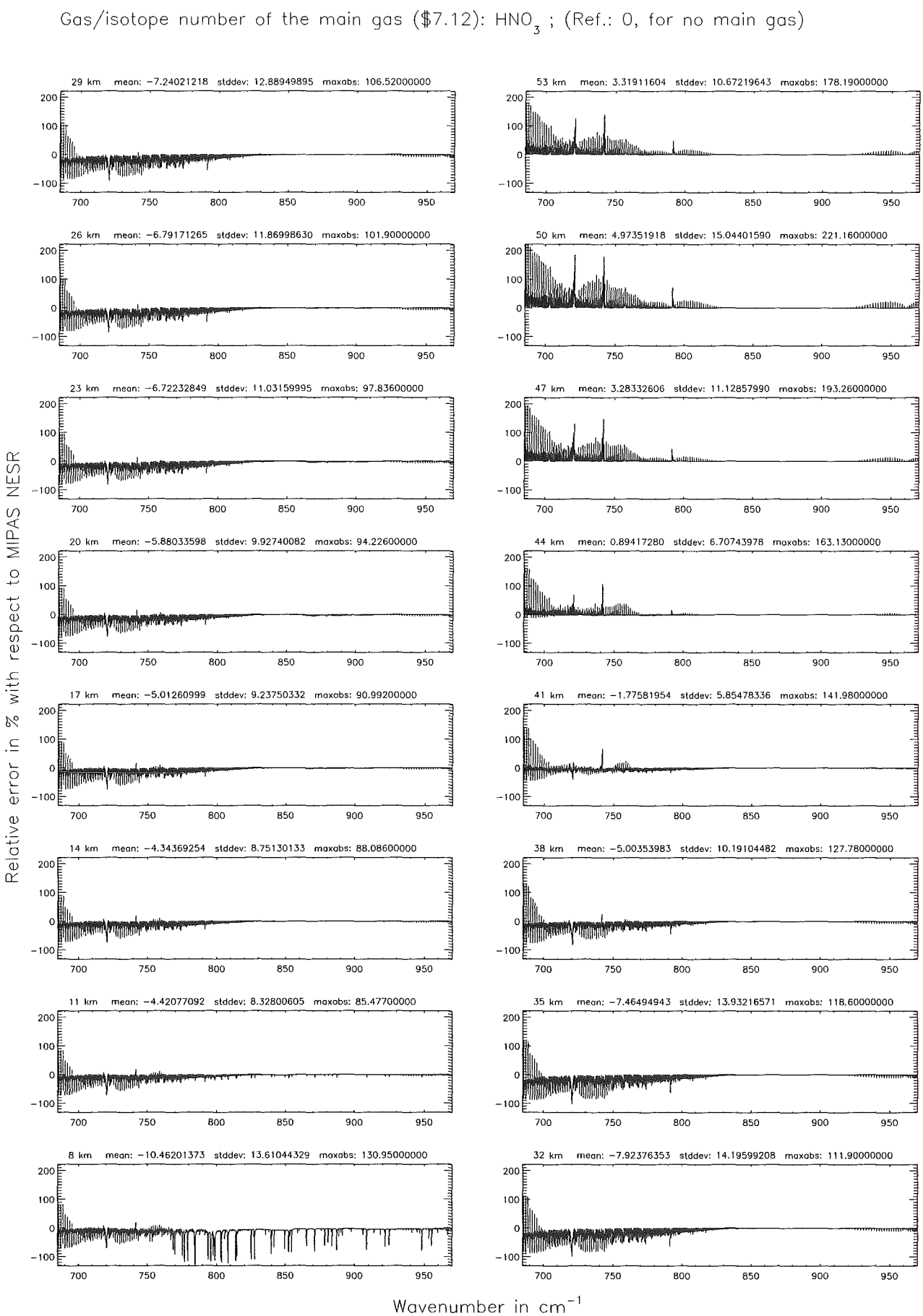


Figure 58: Relative error [%] with respect to MIPAS NESR

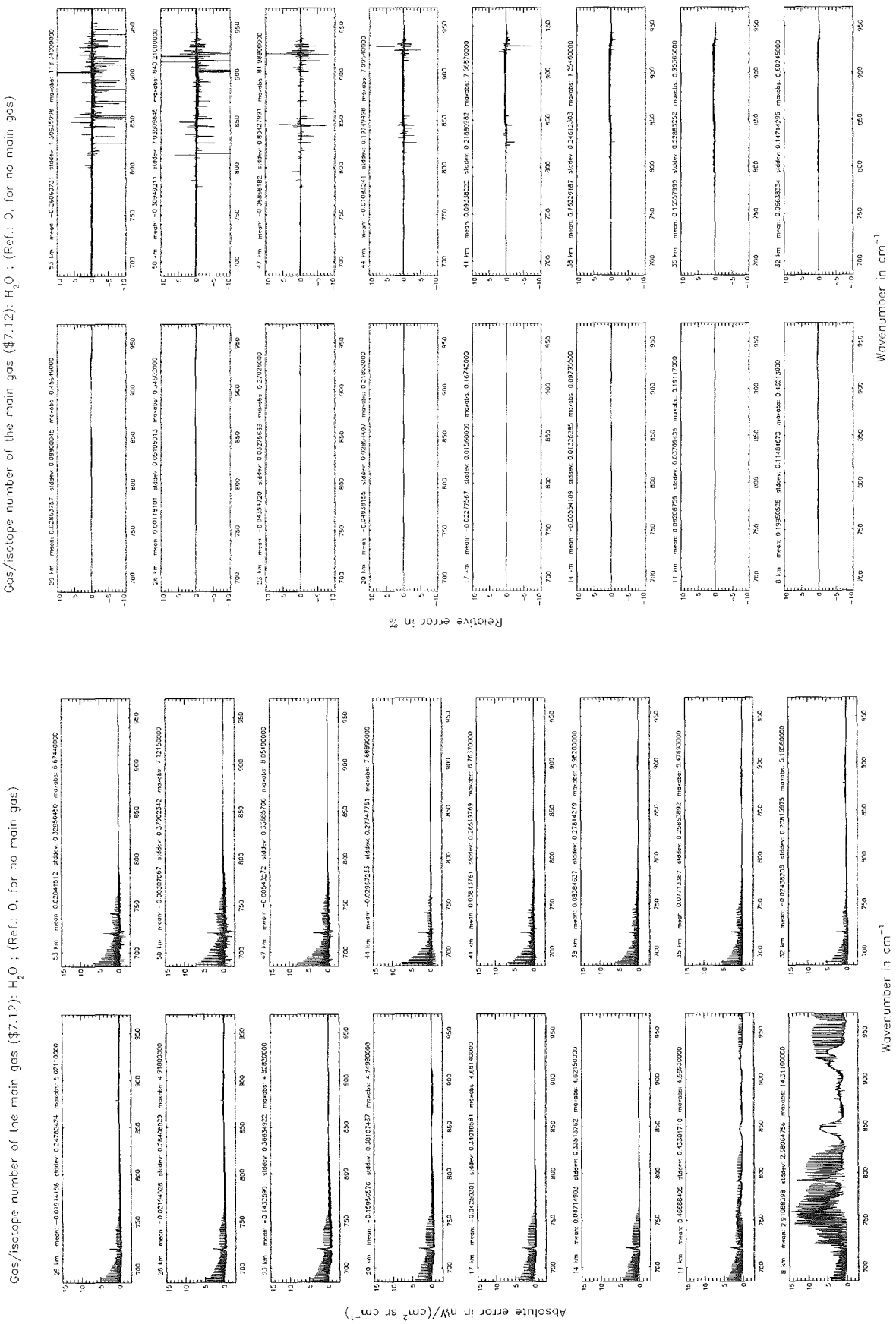


Figure 59: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

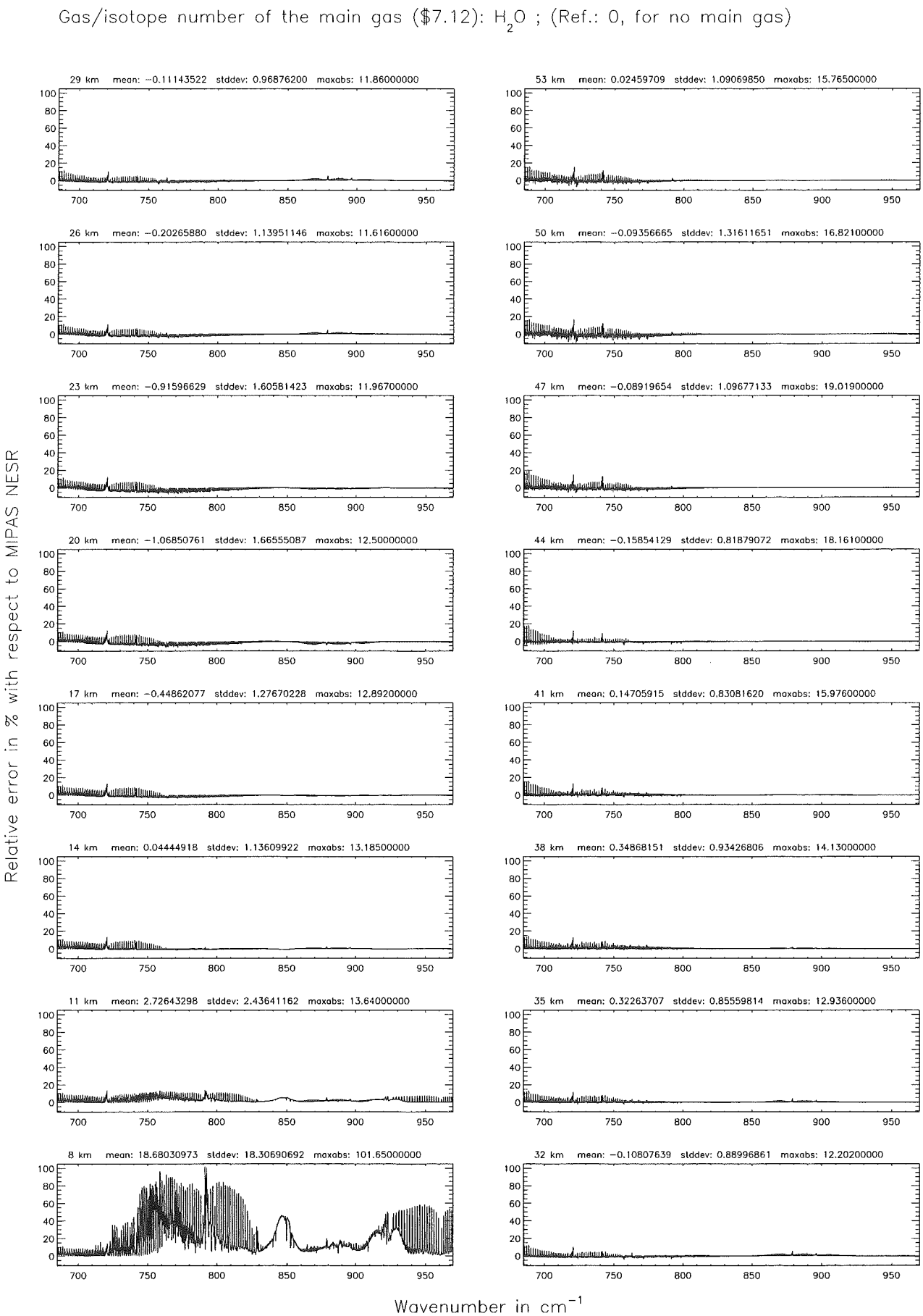


Figure 60: Relative error [%] with respect to MIPAS NESR

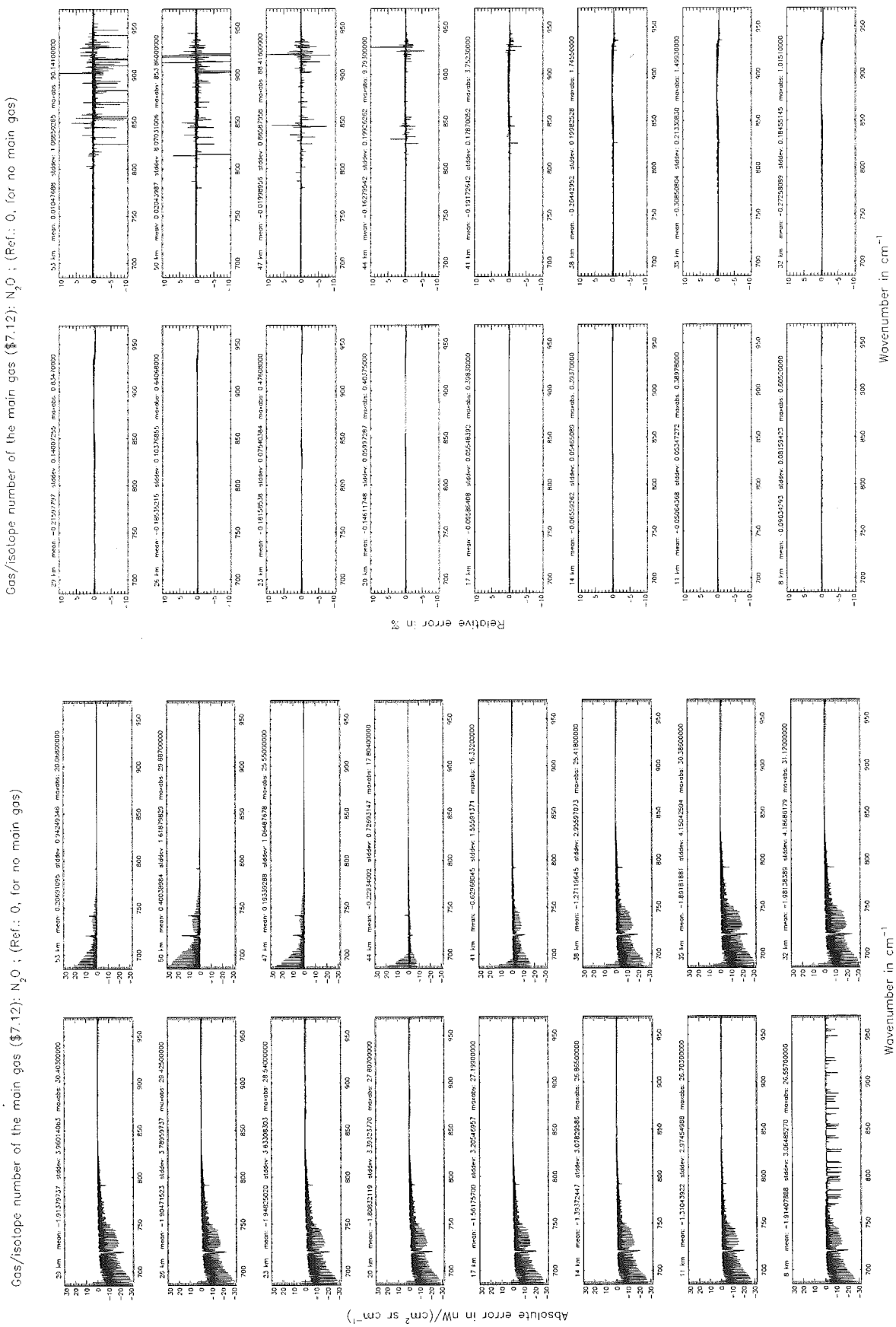


Figure 61: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

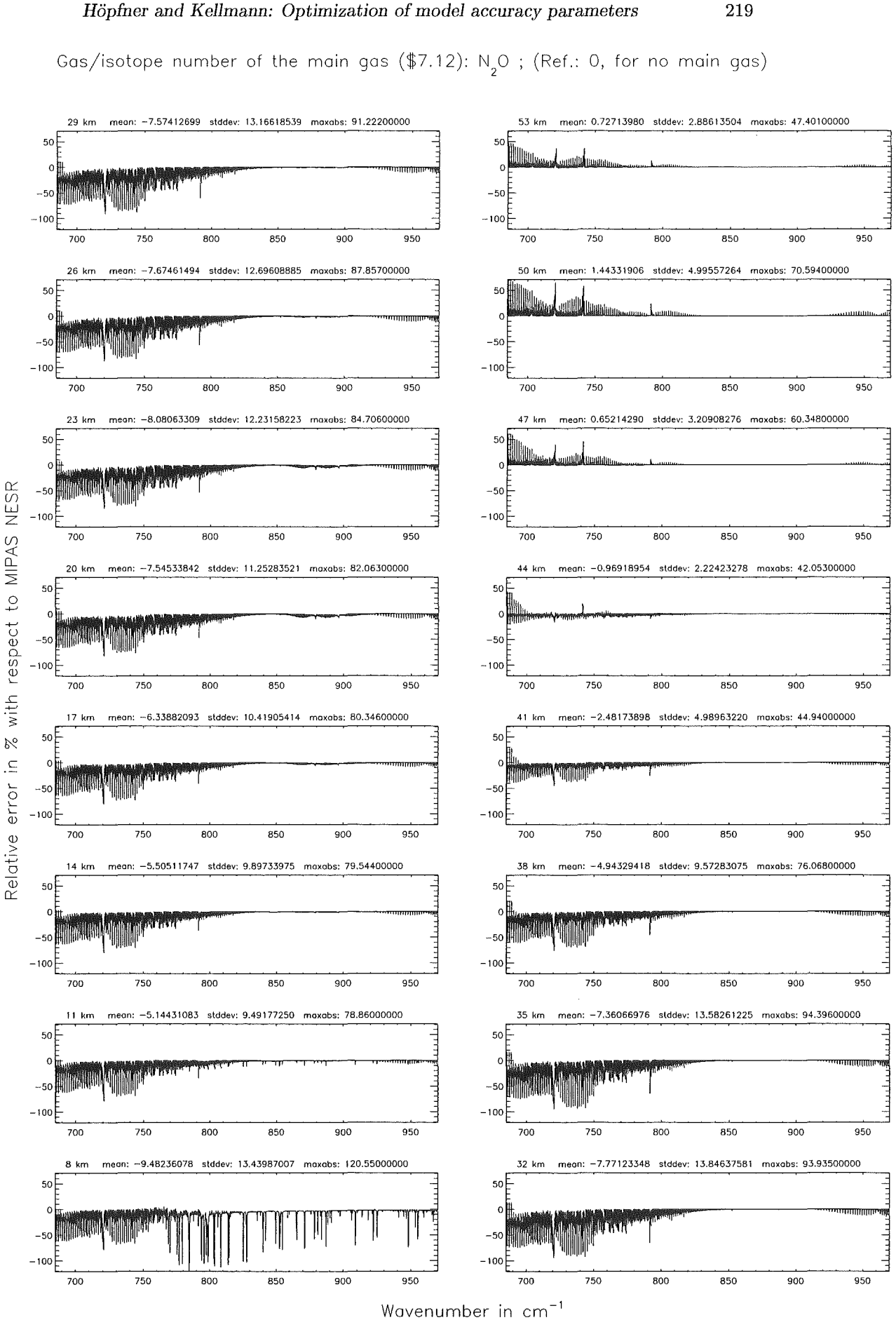


Figure 62: Relative error [%] with respect to MIPAS NESR

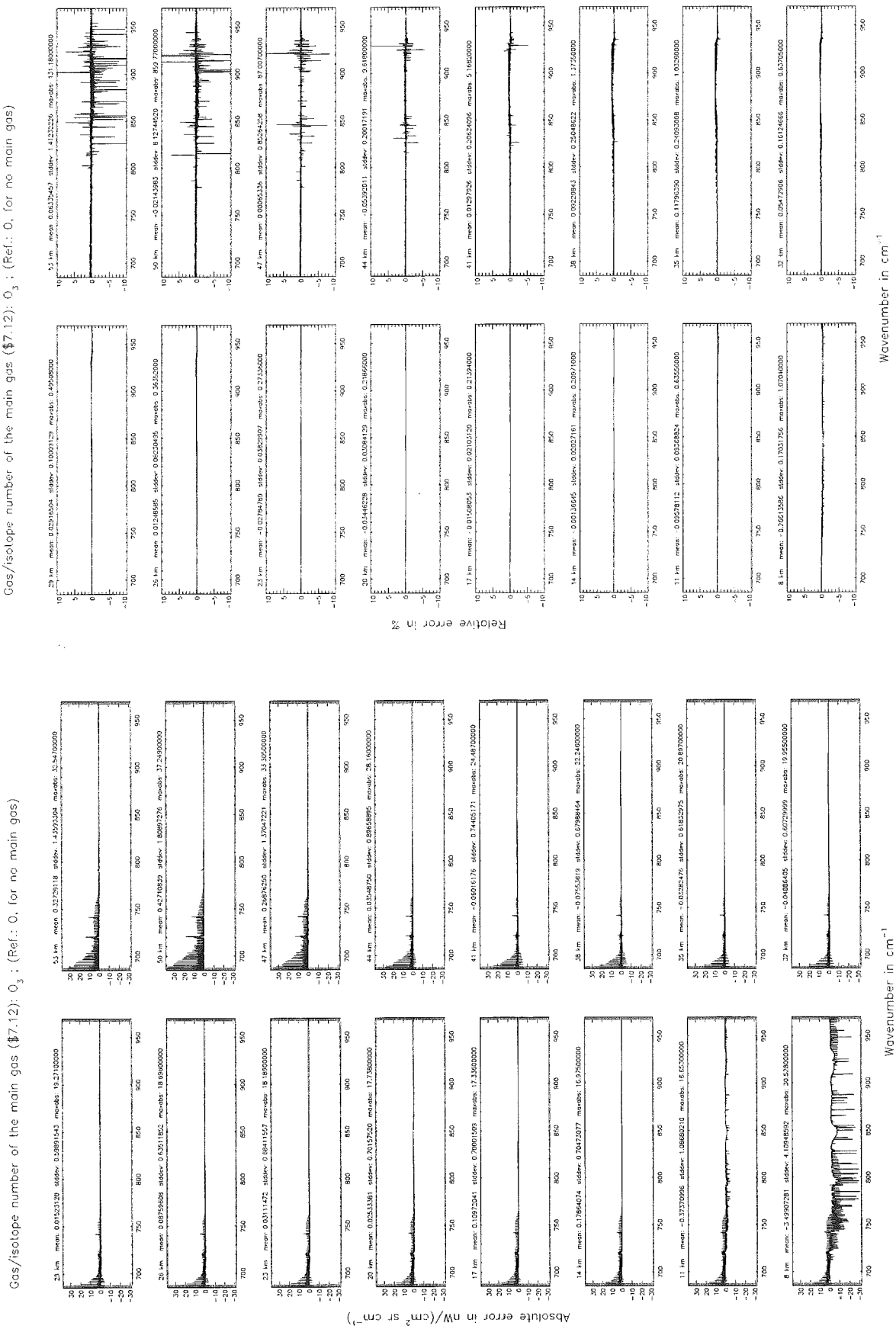


Figure 63: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

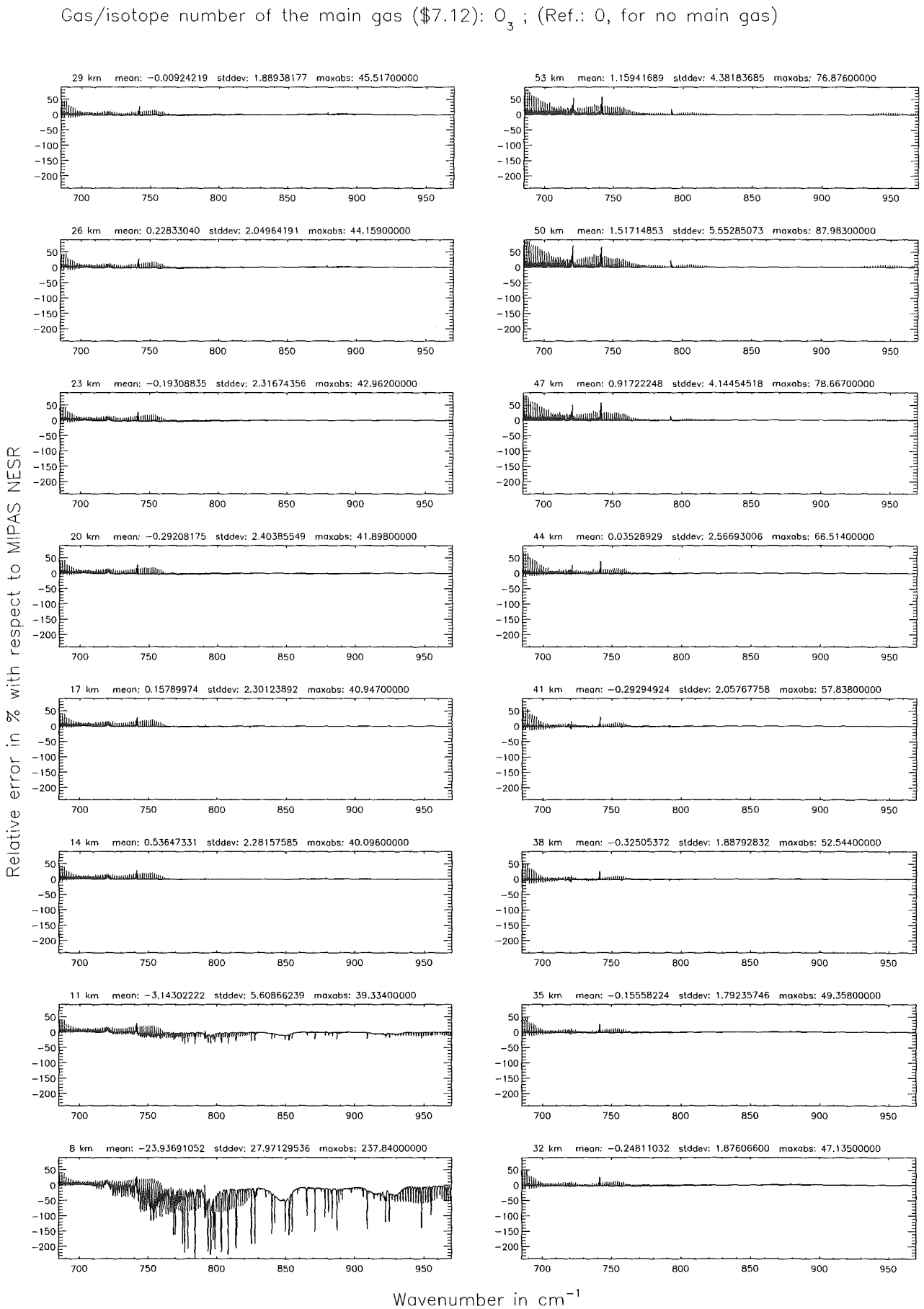


Figure 64: Relative error [%] with respect to MIPAS NESR

Band AB: 1020 – 1170 cm^{-1}

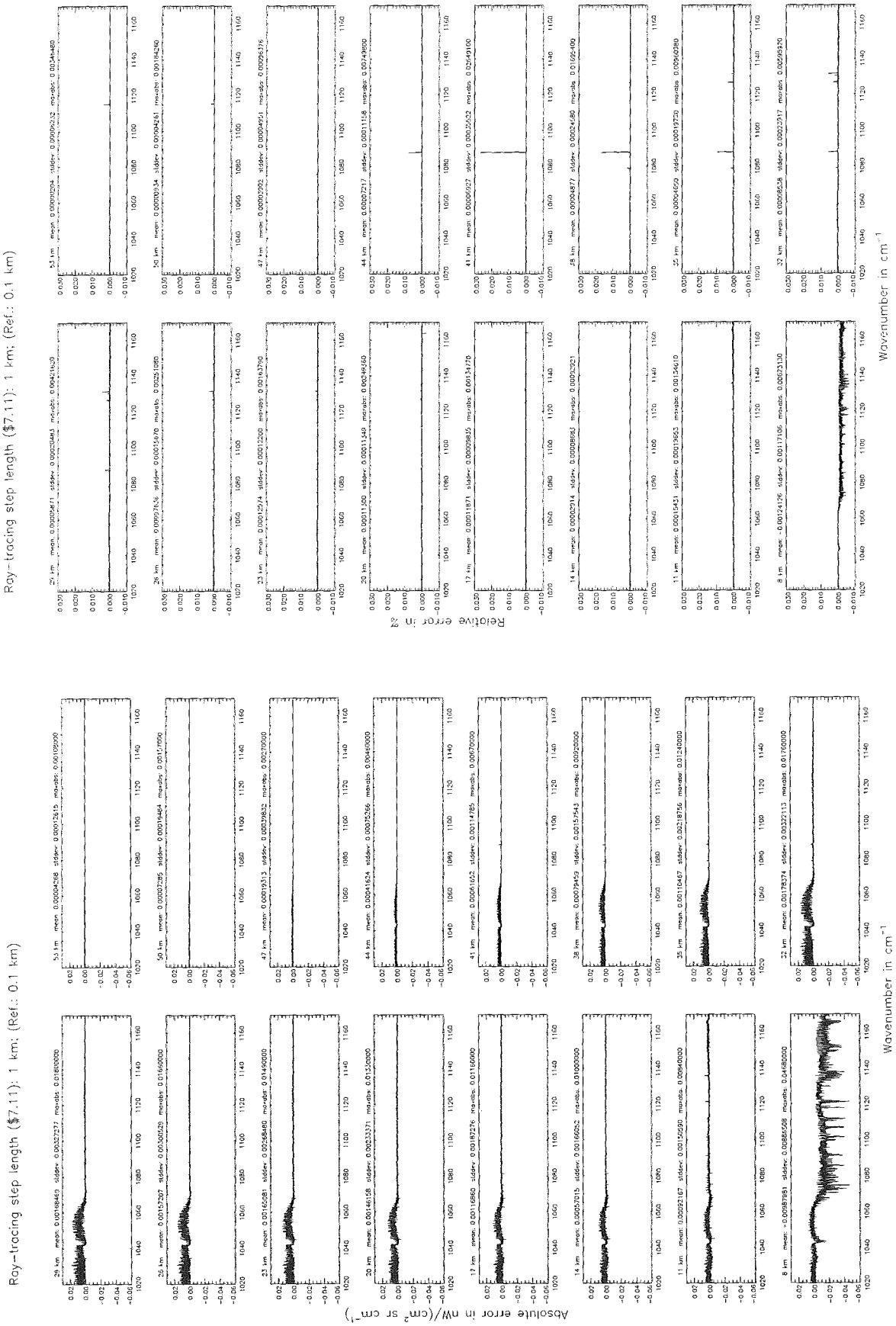


Figure 65: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

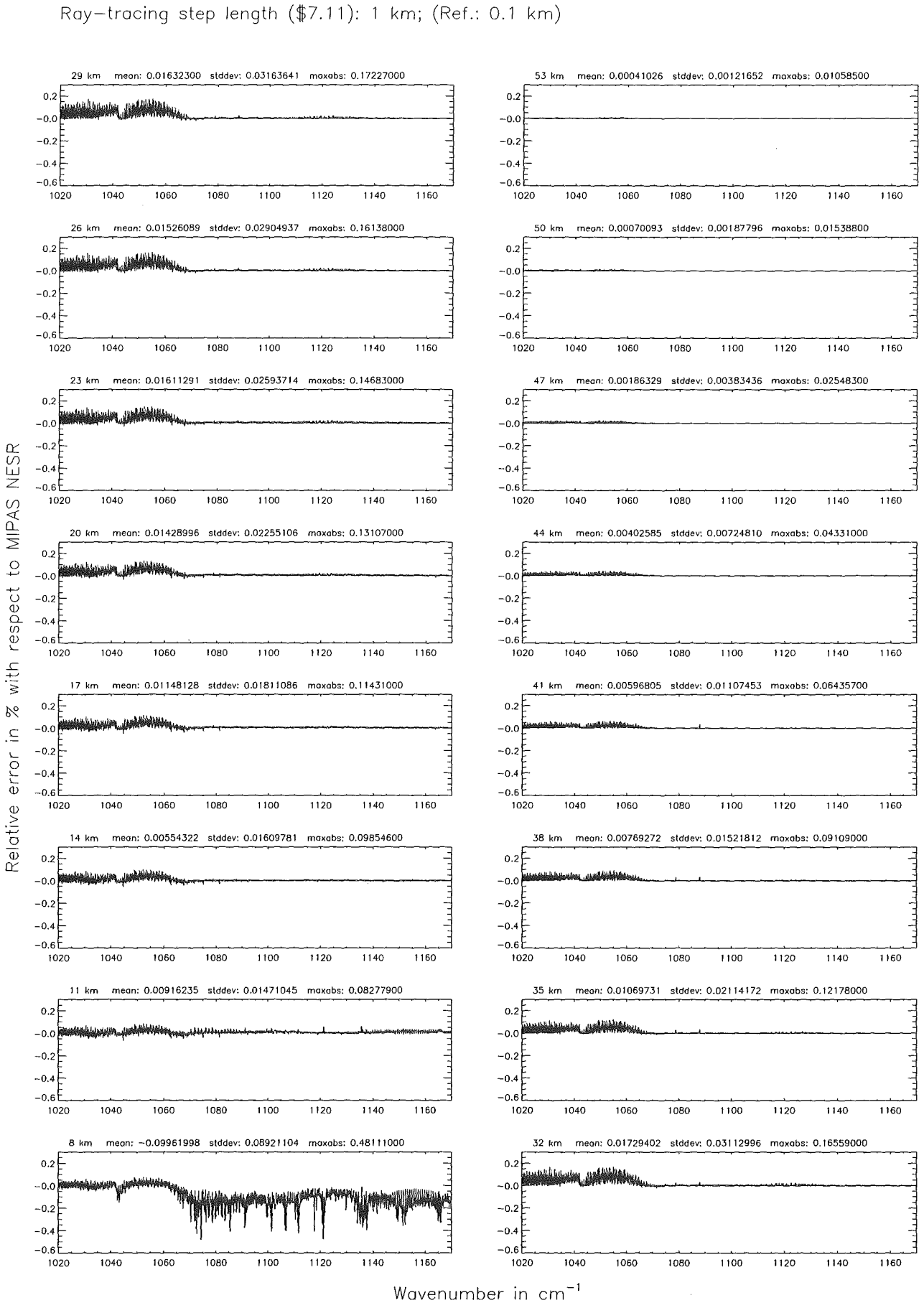


Figure 66: Relative error [%] with respect to MIPAS NESR

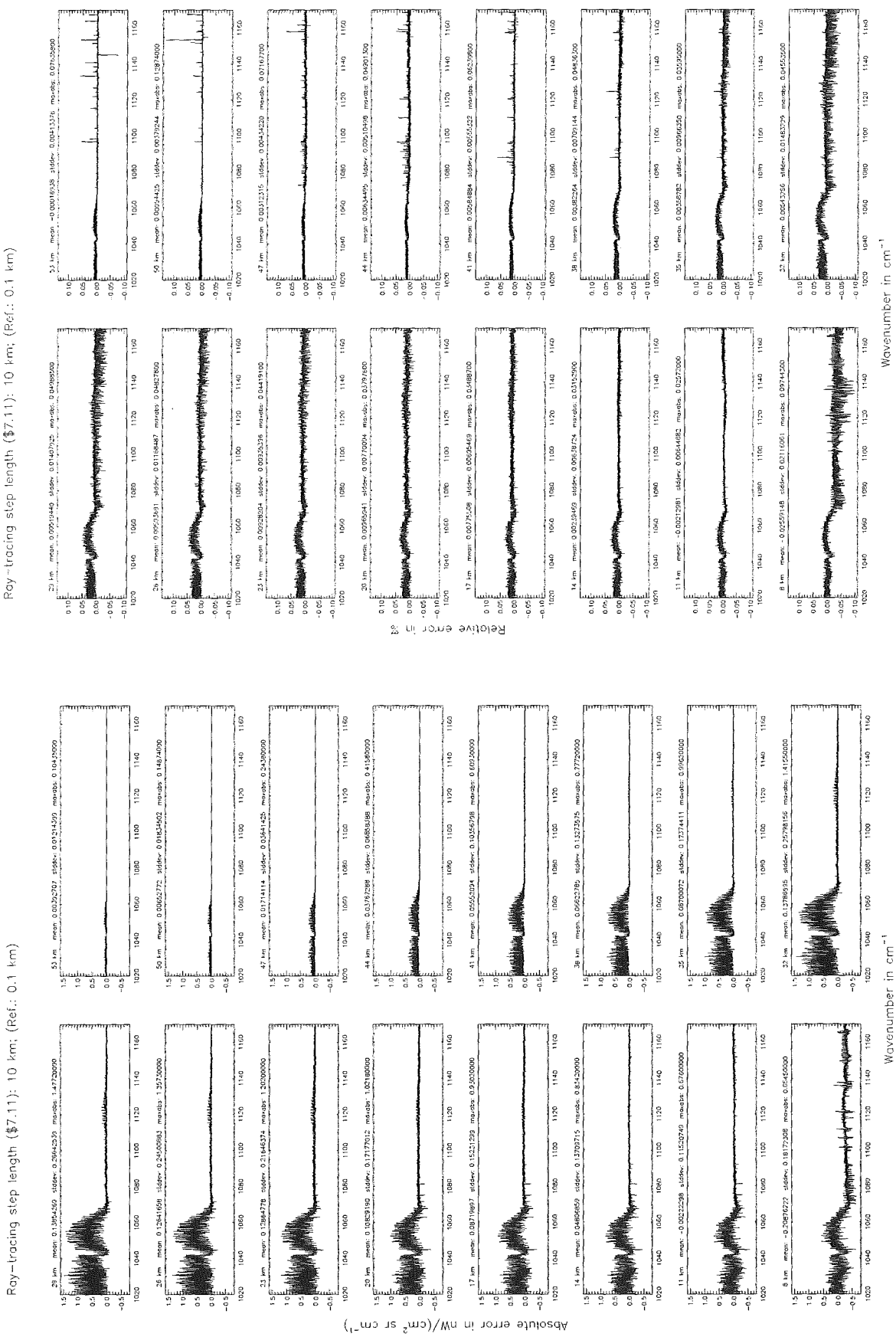


Figure 67: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

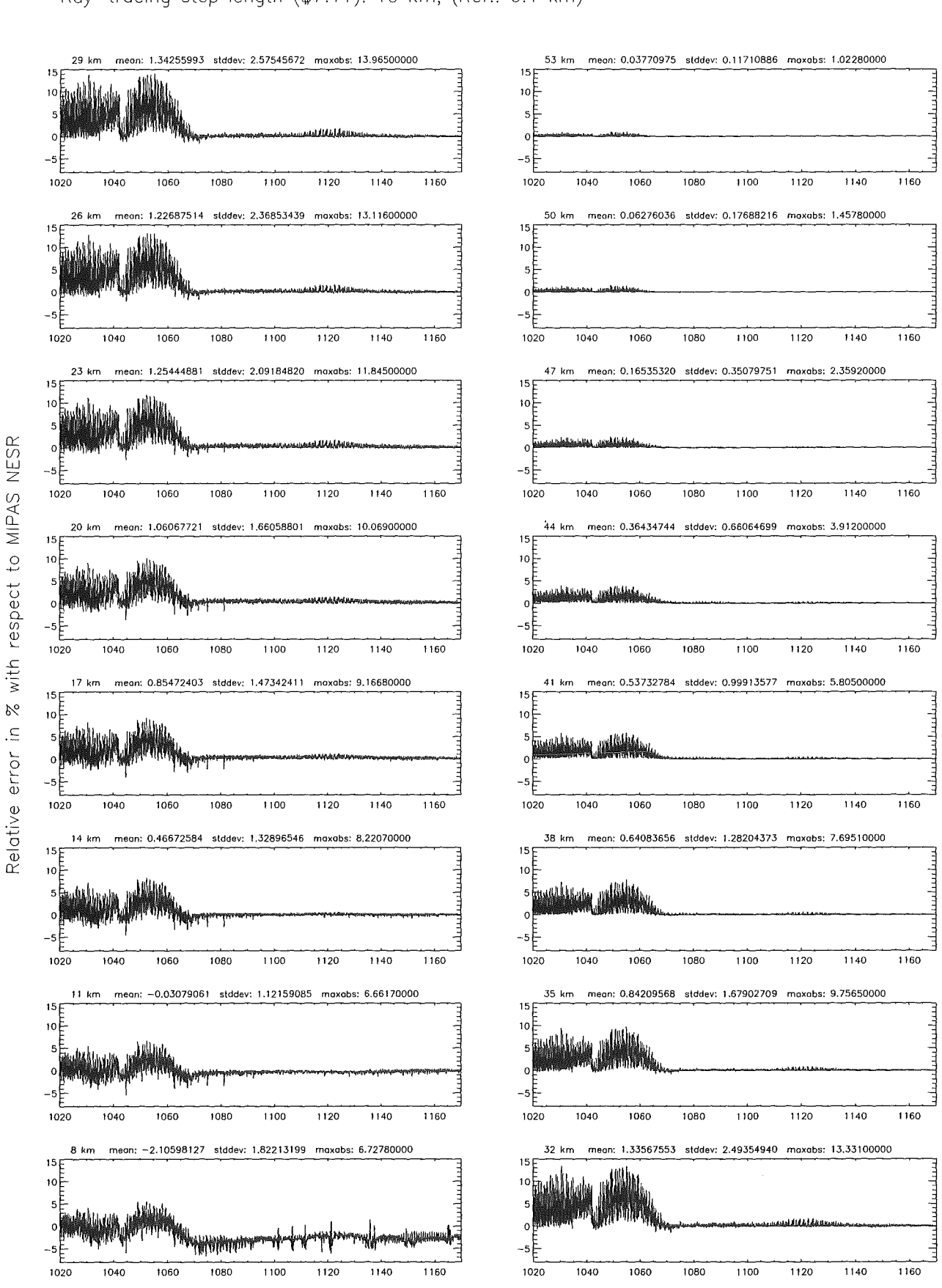


Figure 68: Relative error [%] with respect to MIPAS NESR

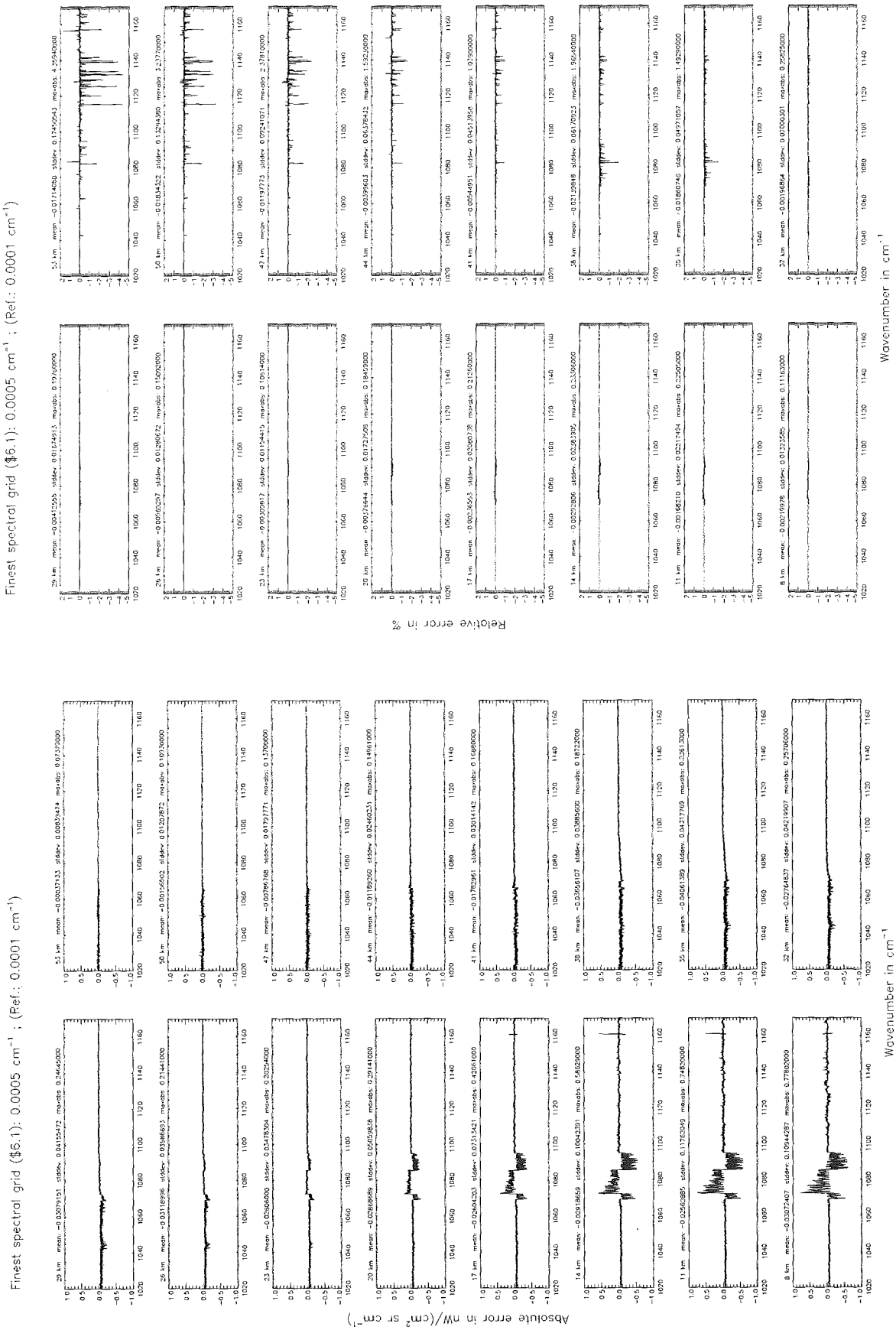


Figure 69: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

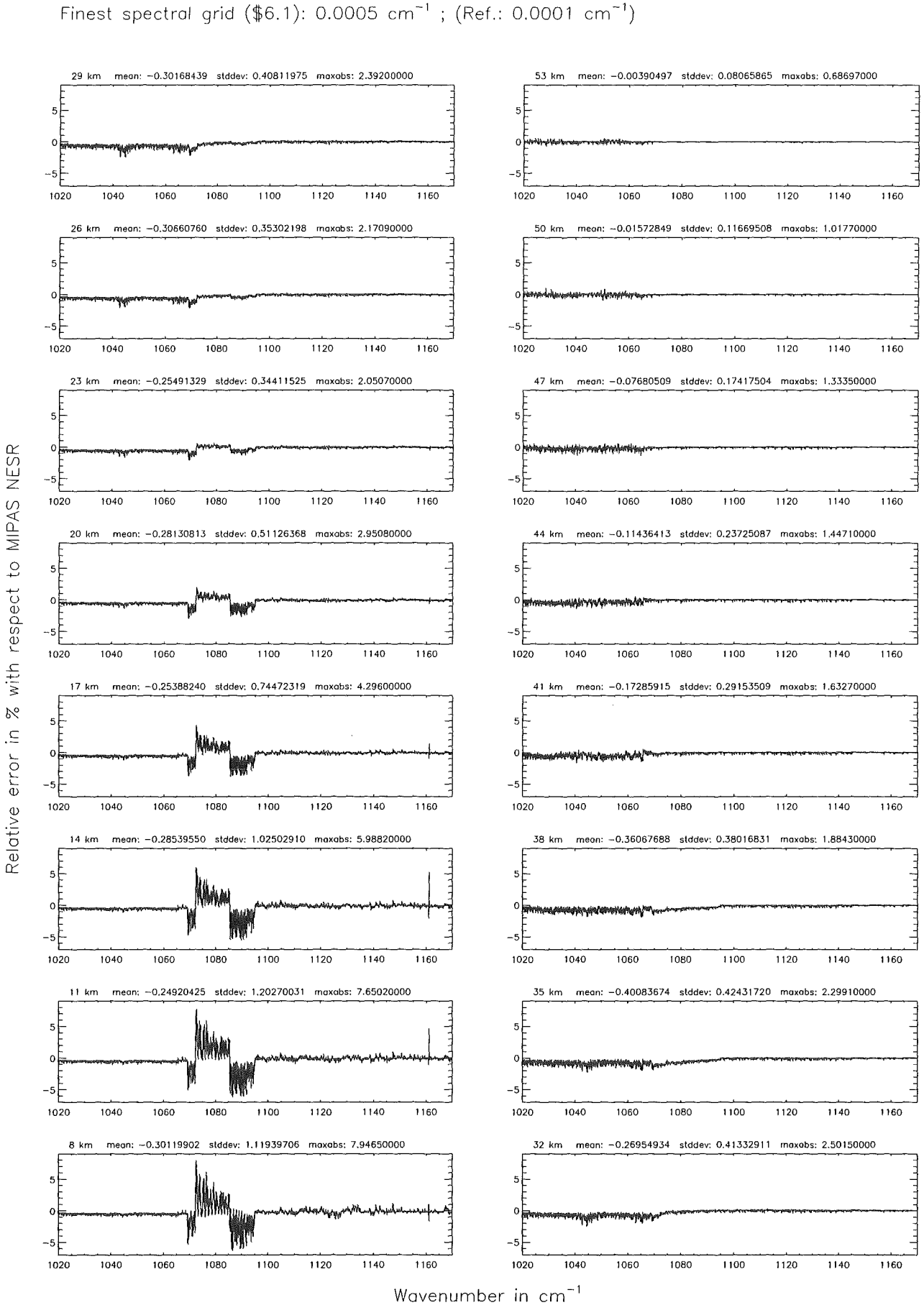


Figure 70: Relative error [%] with respect to MIPAS NESR

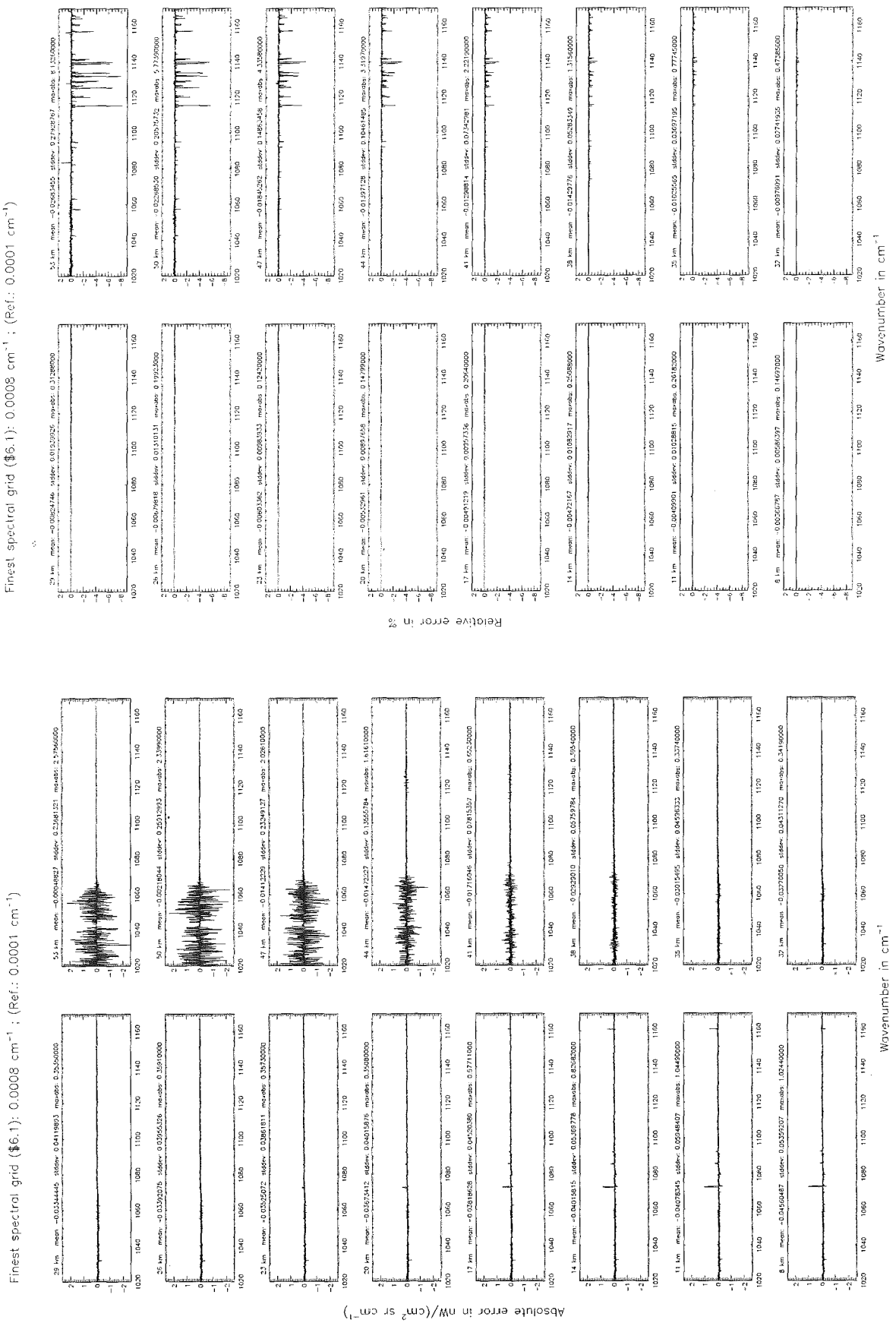


Figure 71: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

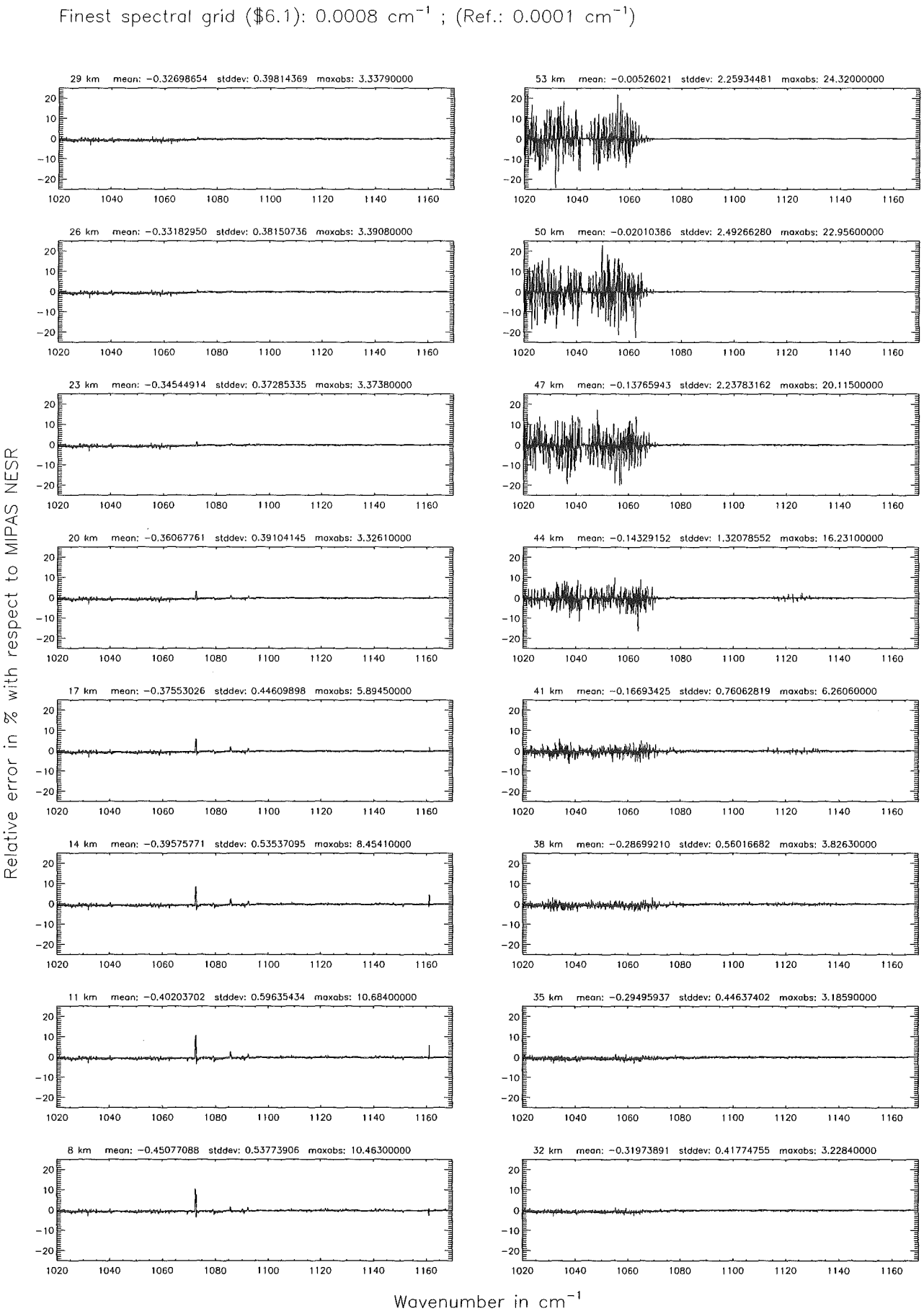


Figure 72: Relative error [%] with respect to MIPAS NESR

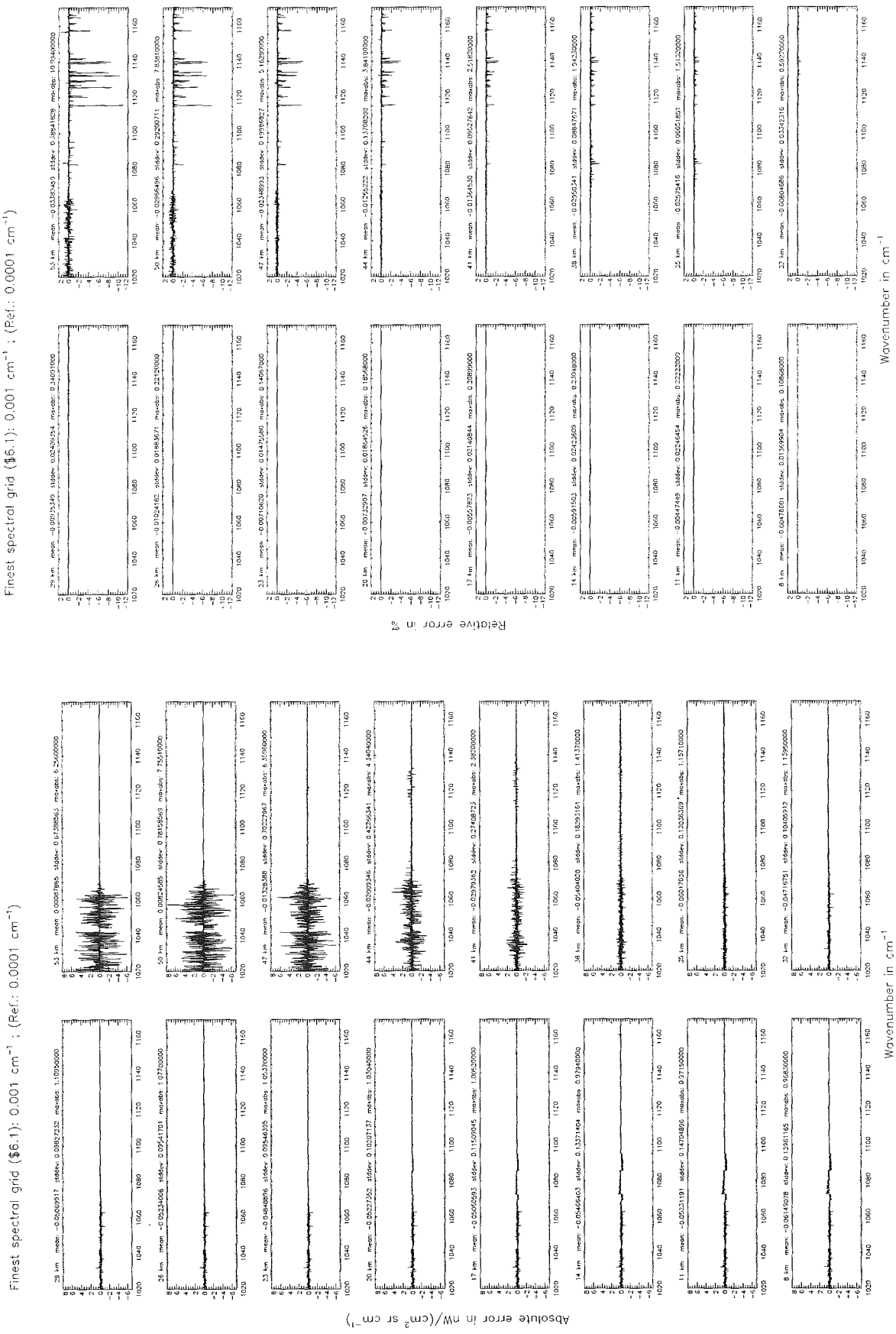


Figure 73: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

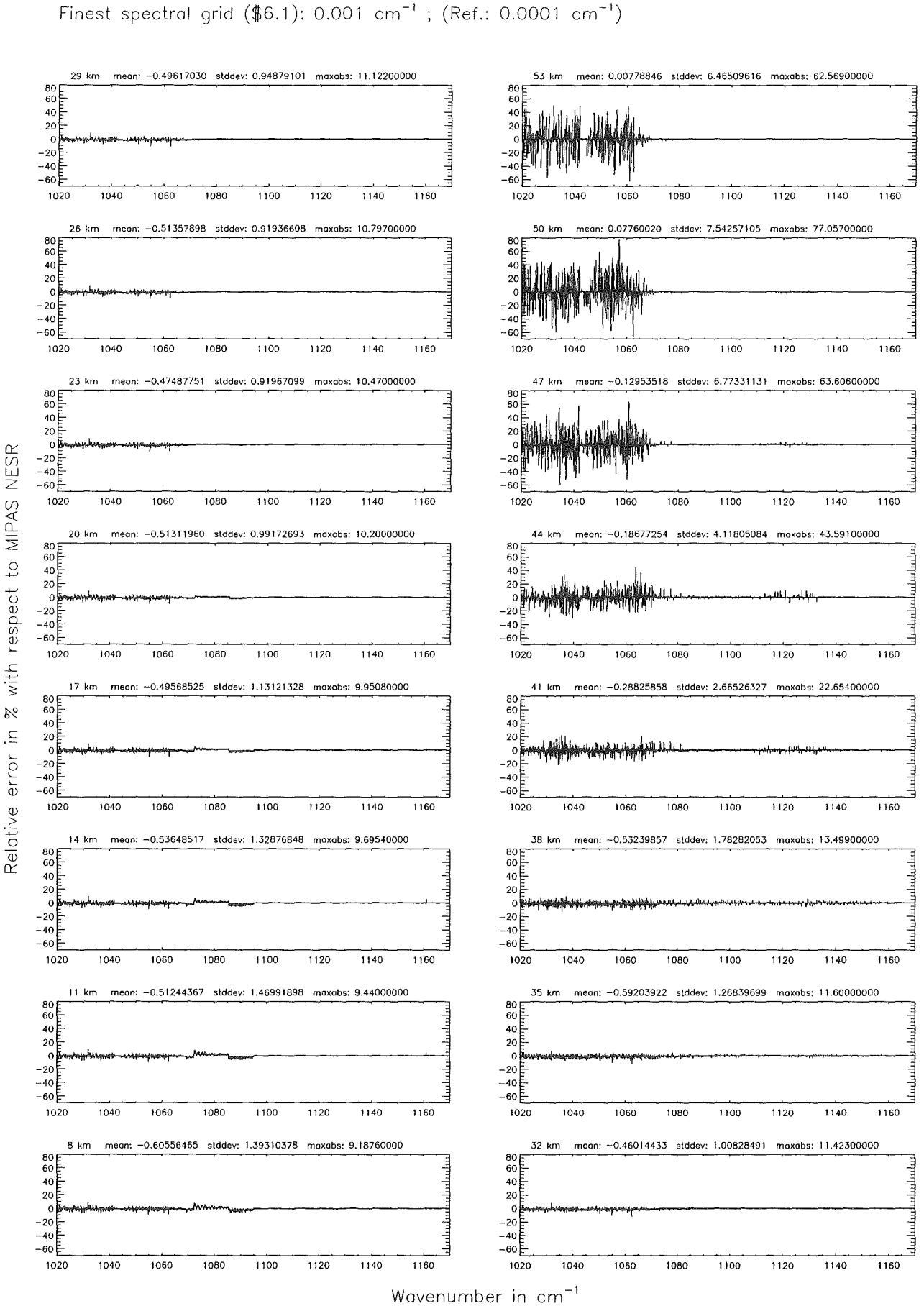


Figure 74: Relative error [%] with respect to MIPAS NESR

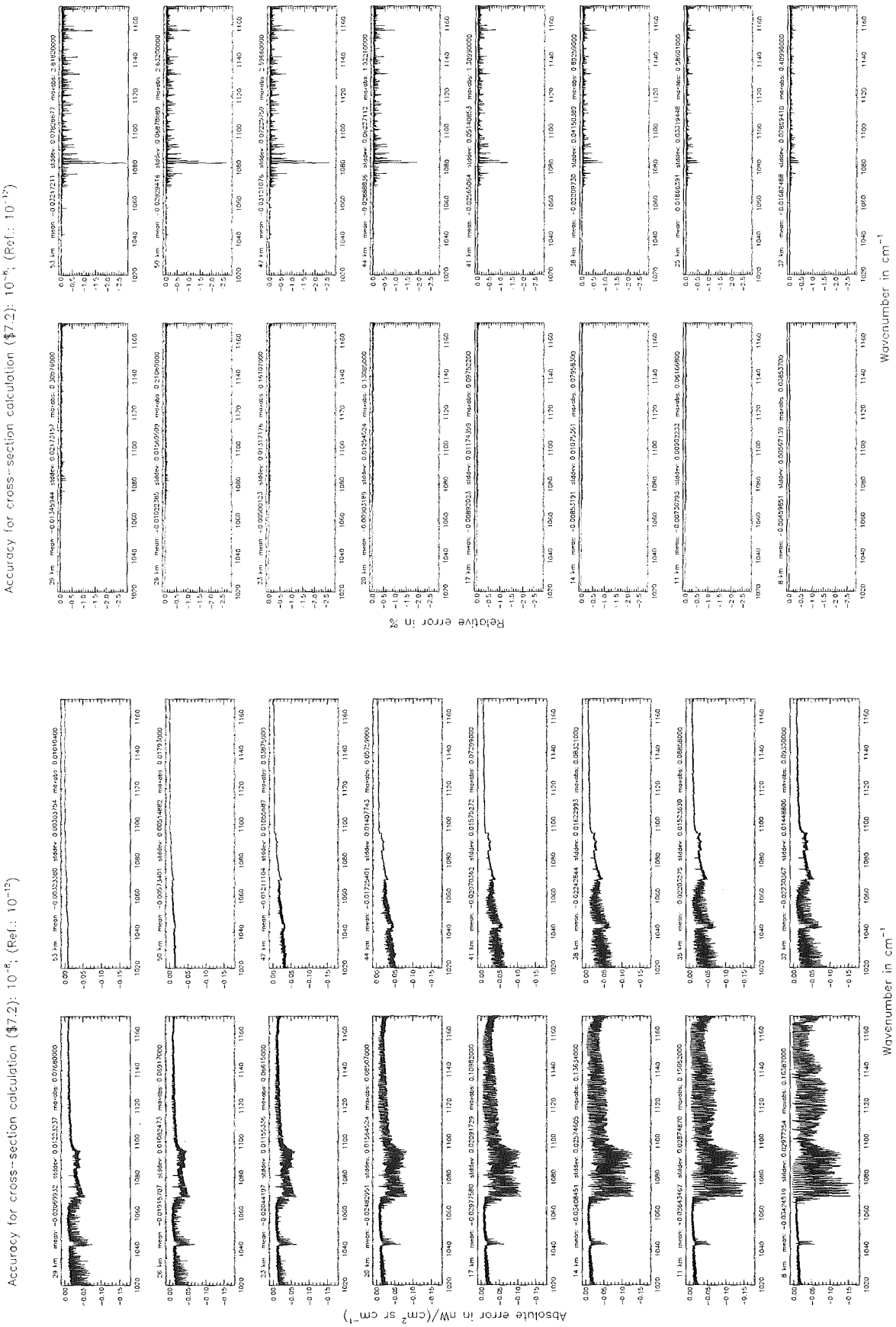


Figure 75: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

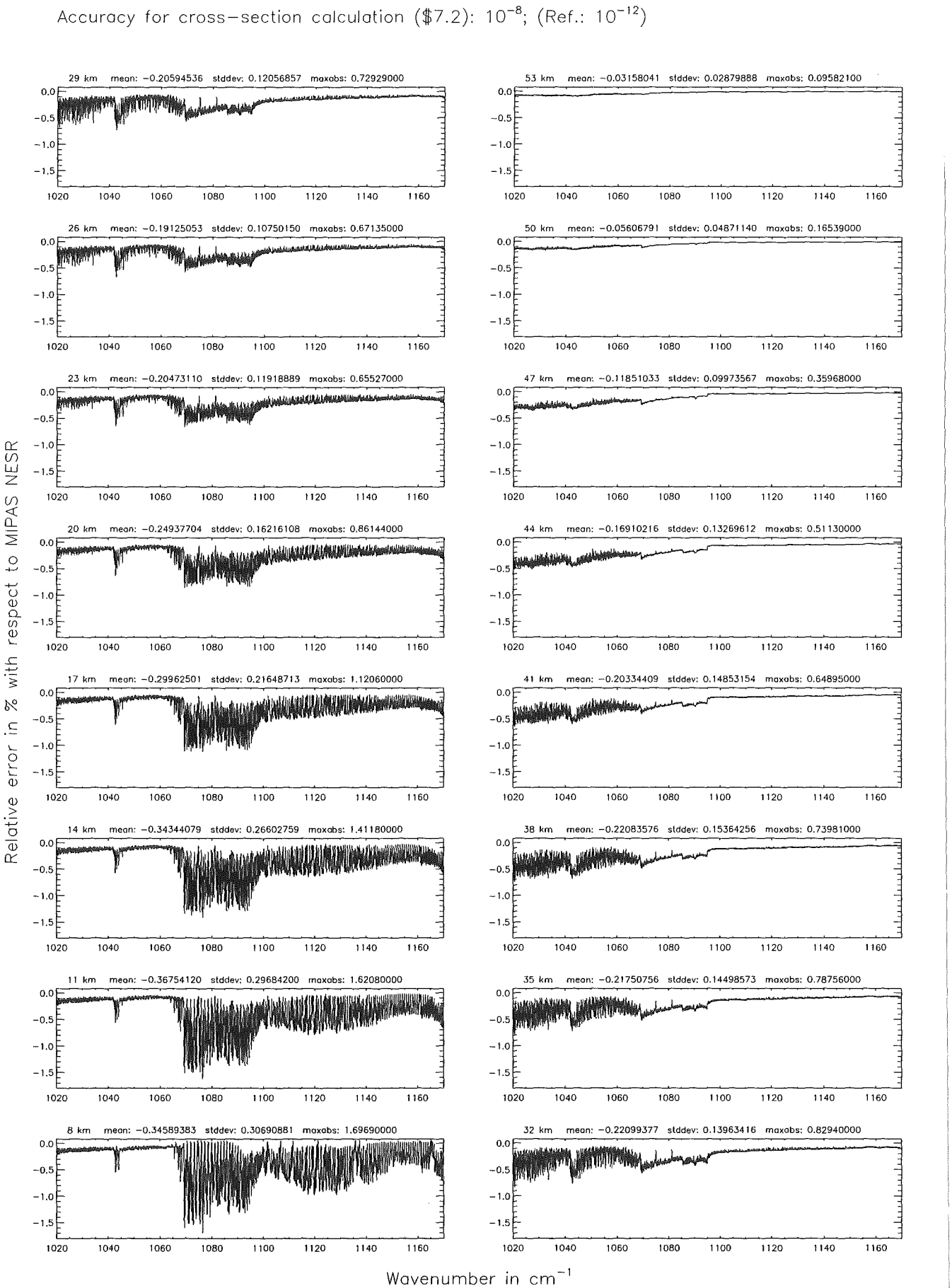


Figure 76: Relative error [%] with respect to MIPAS NESR

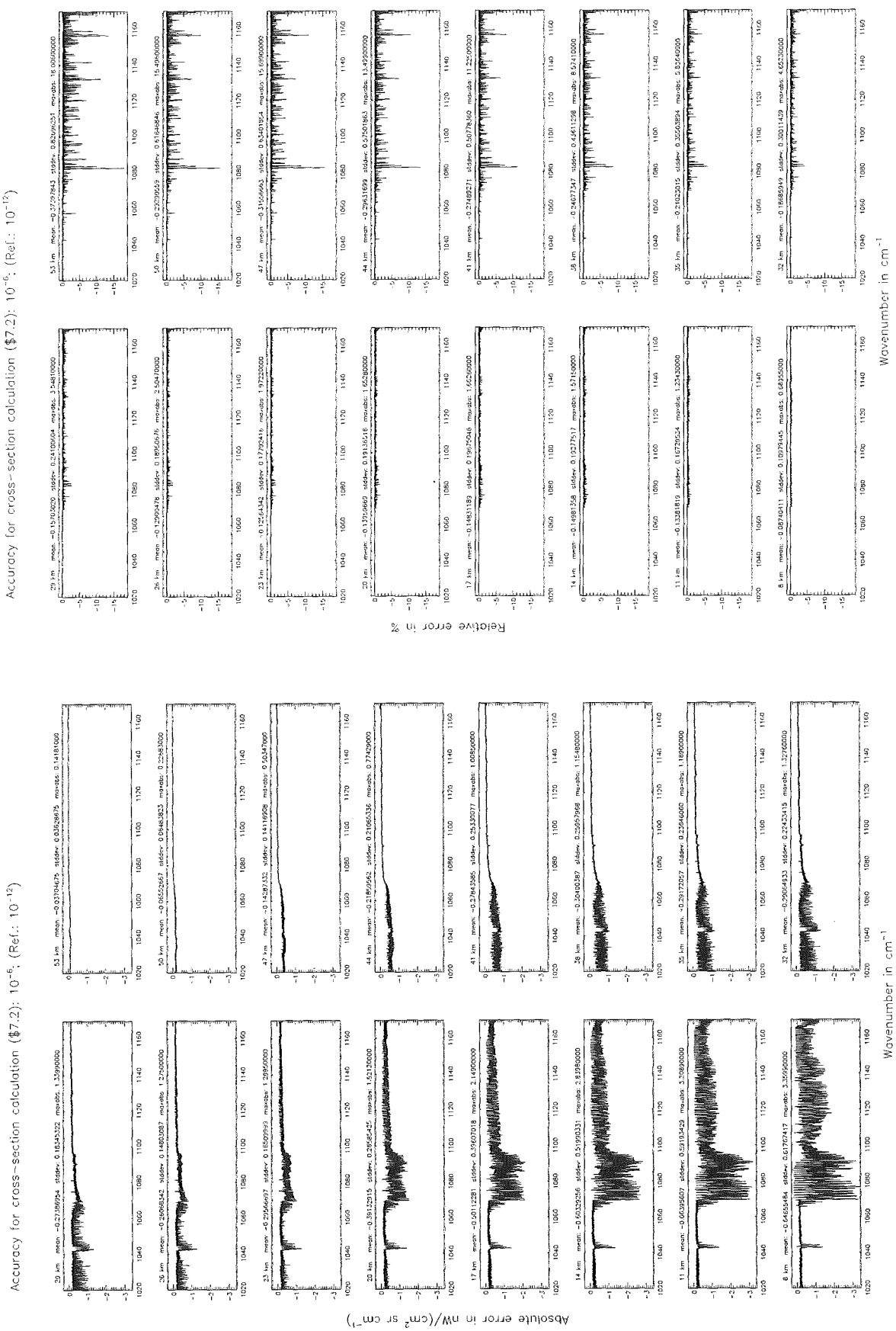


Figure 77: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

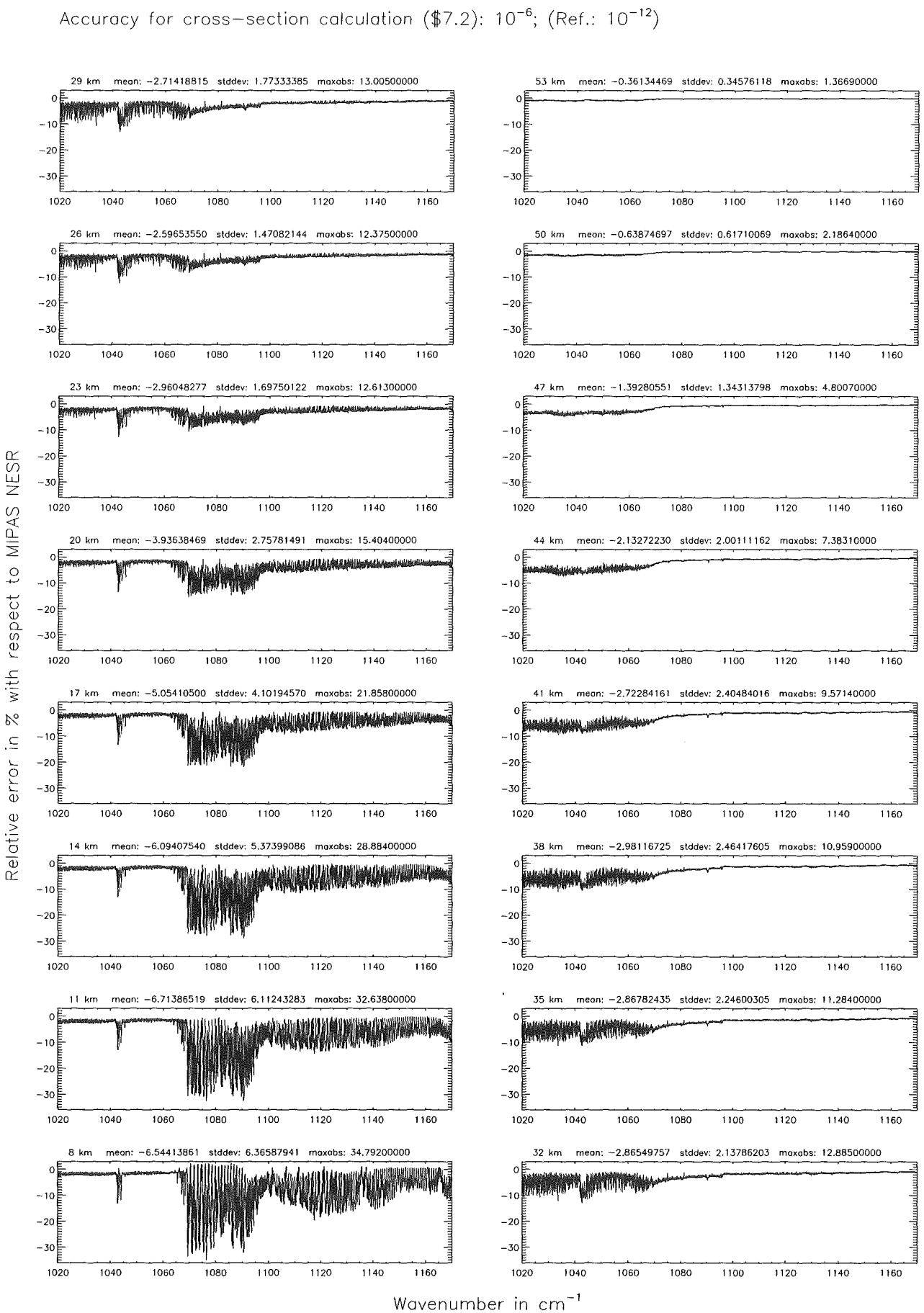
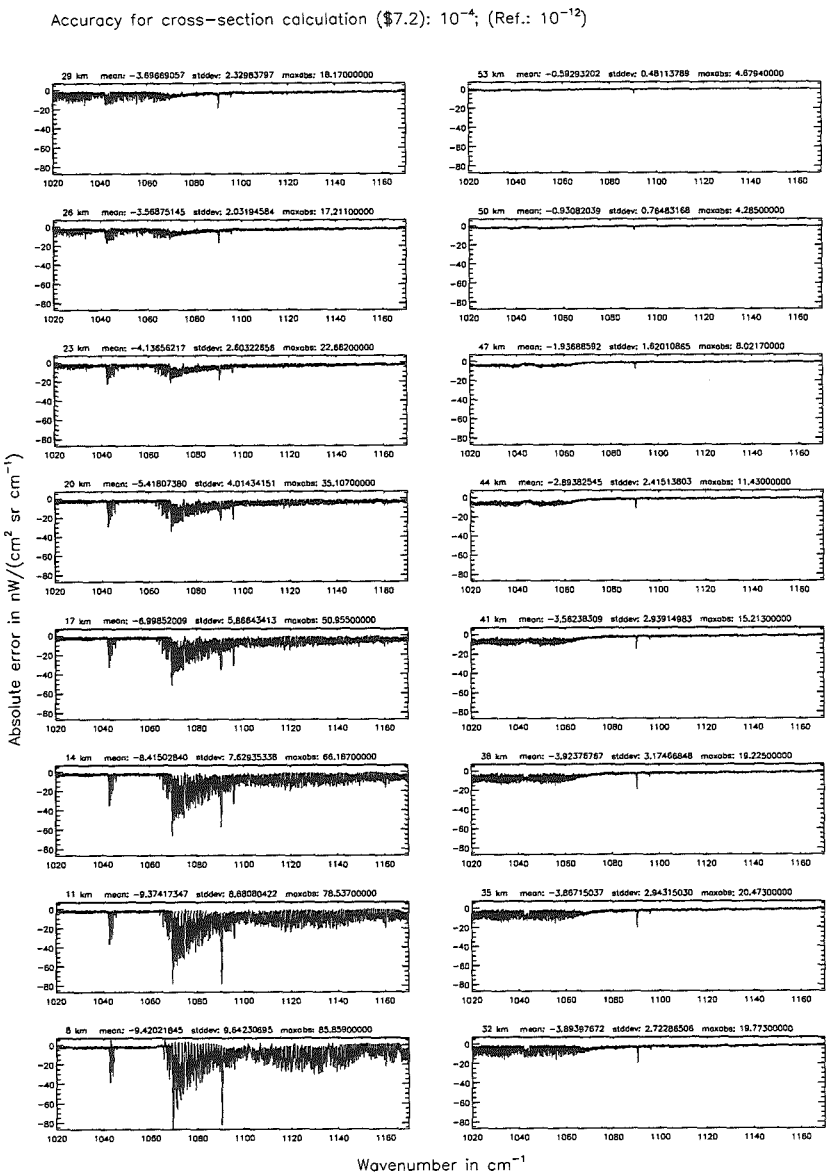


Figure 78: Relative error [%] with respect to MIPAS NESR



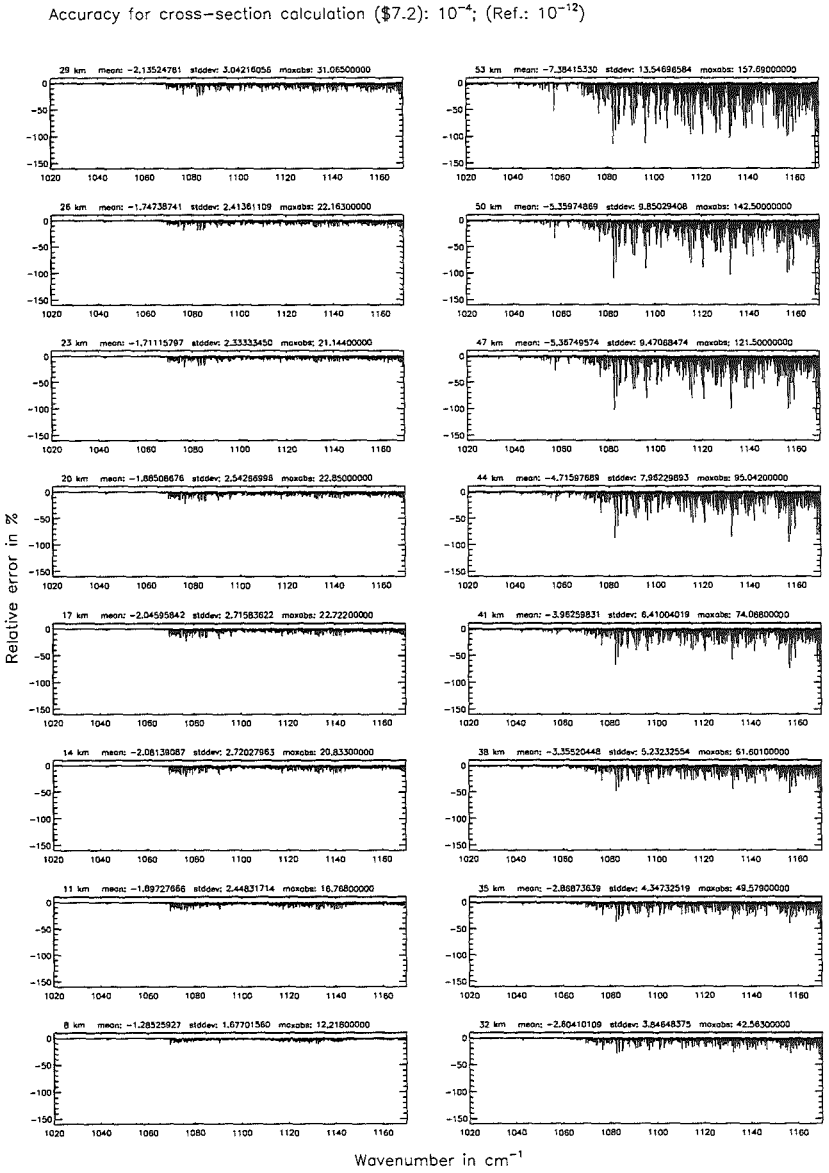


Figure 79: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

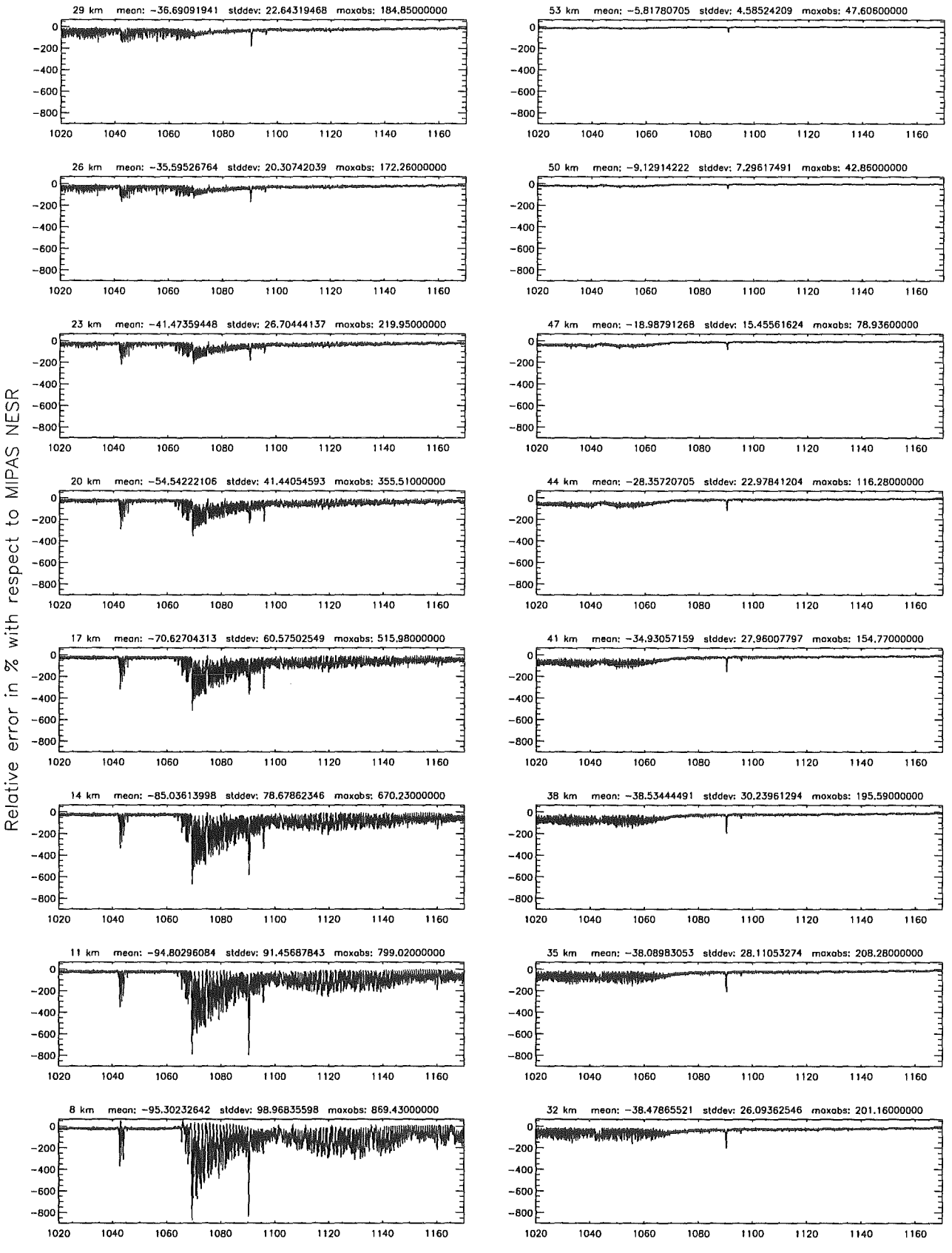
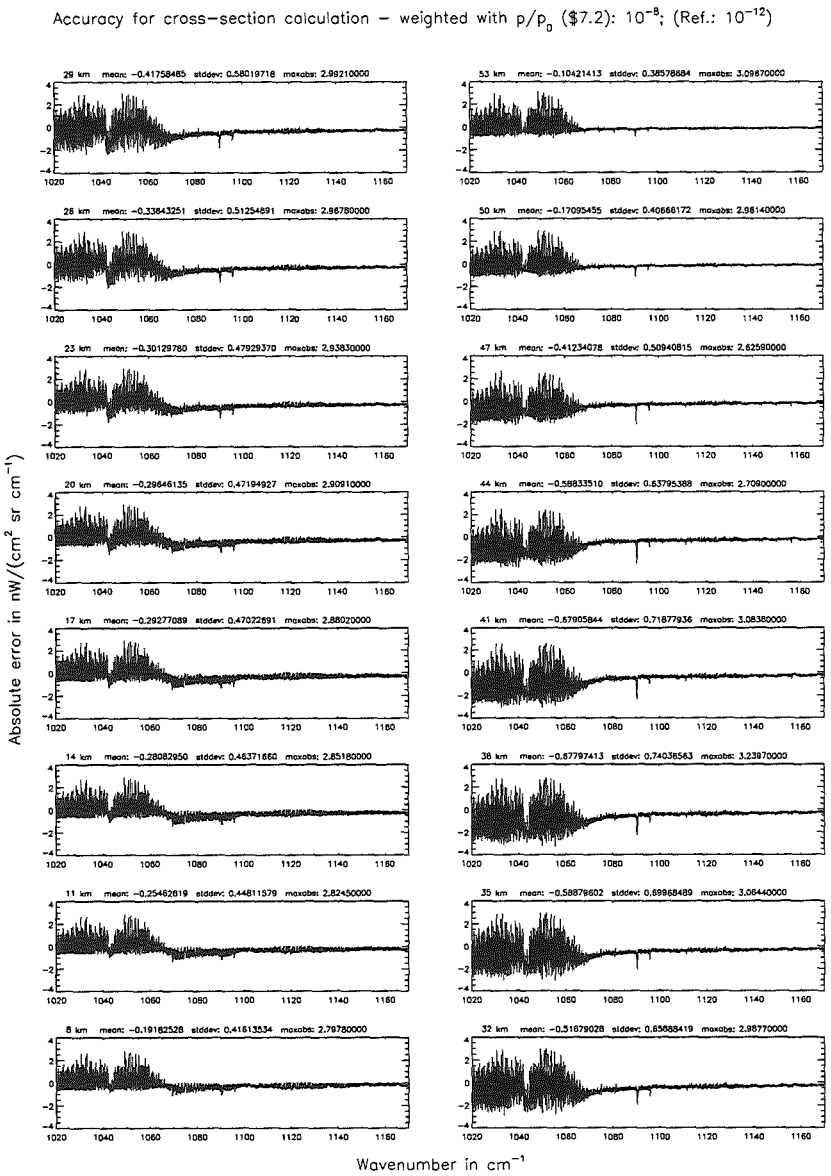


Figure 80: Relative error [%] with respect to MIPAS NESR



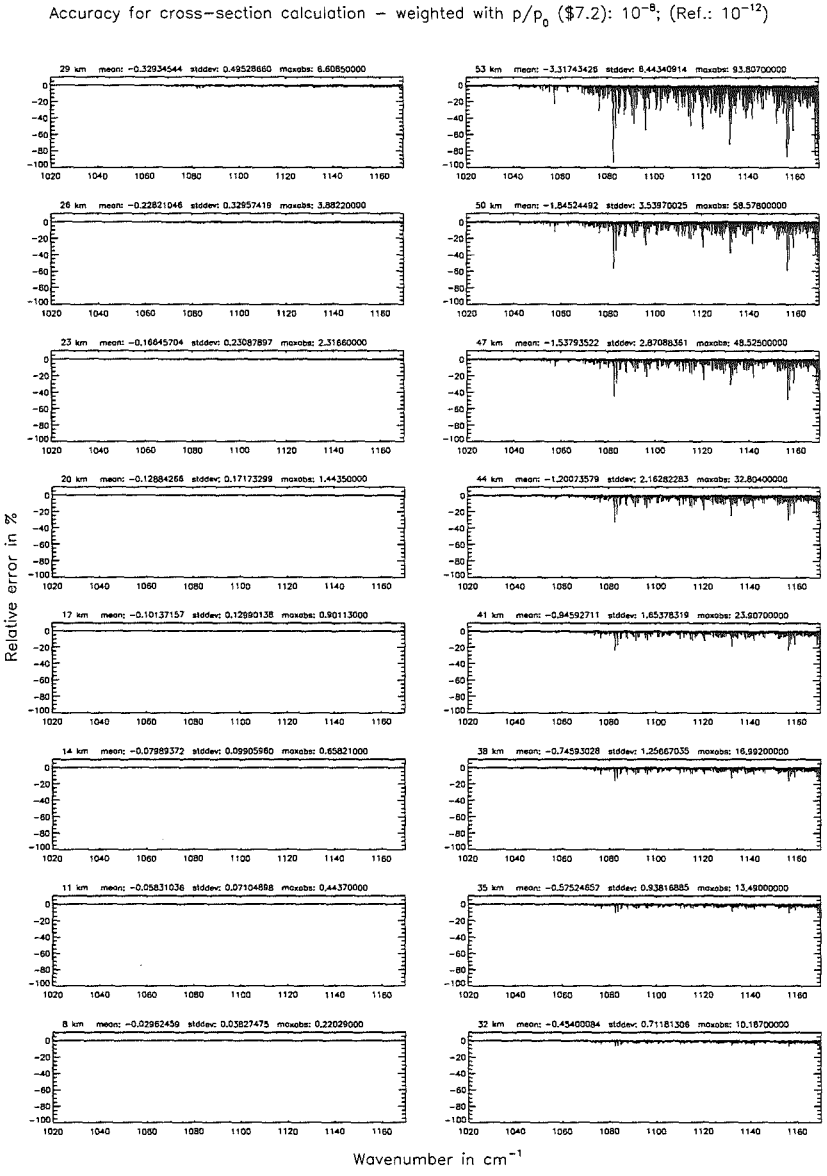


Figure 81: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

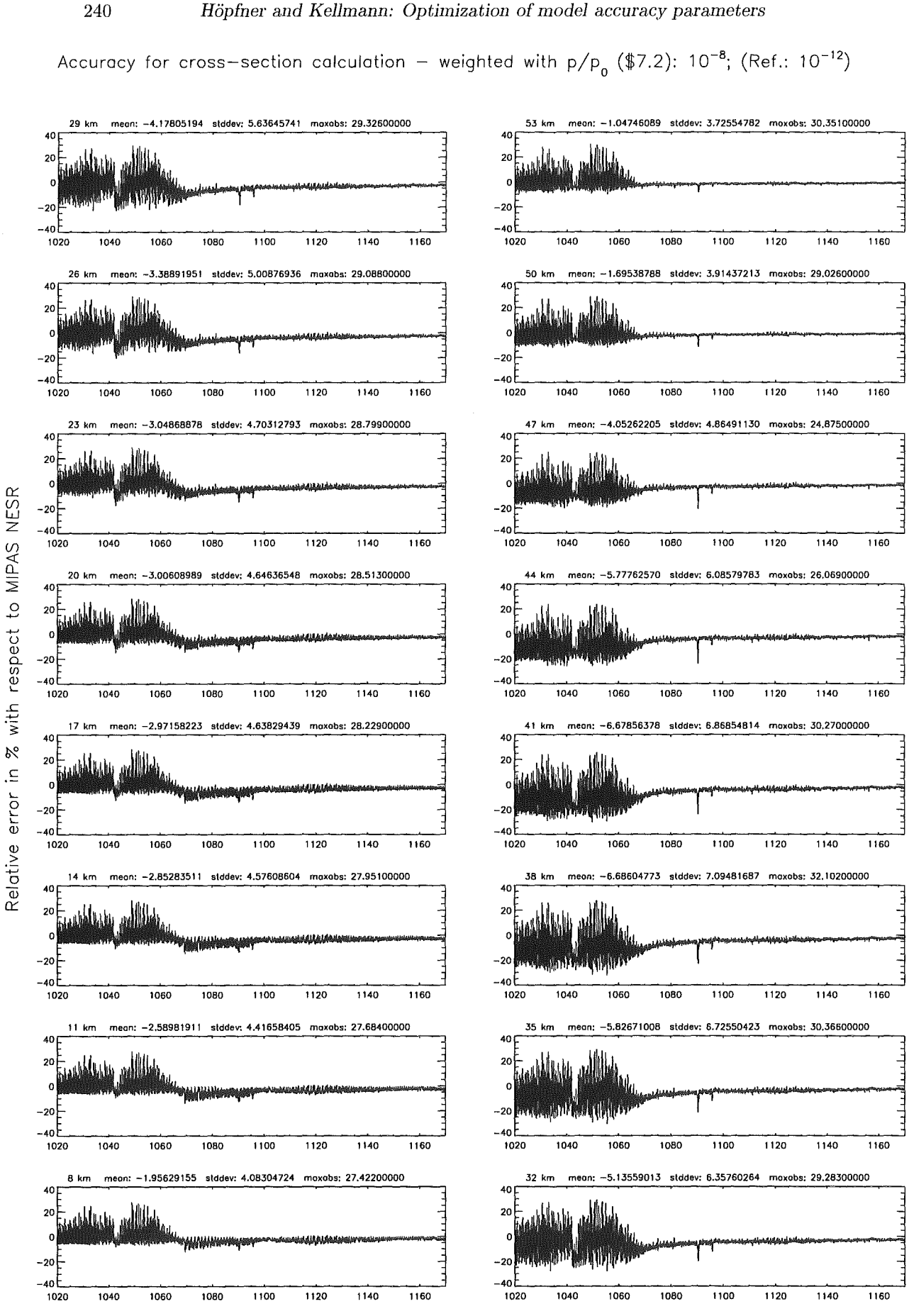
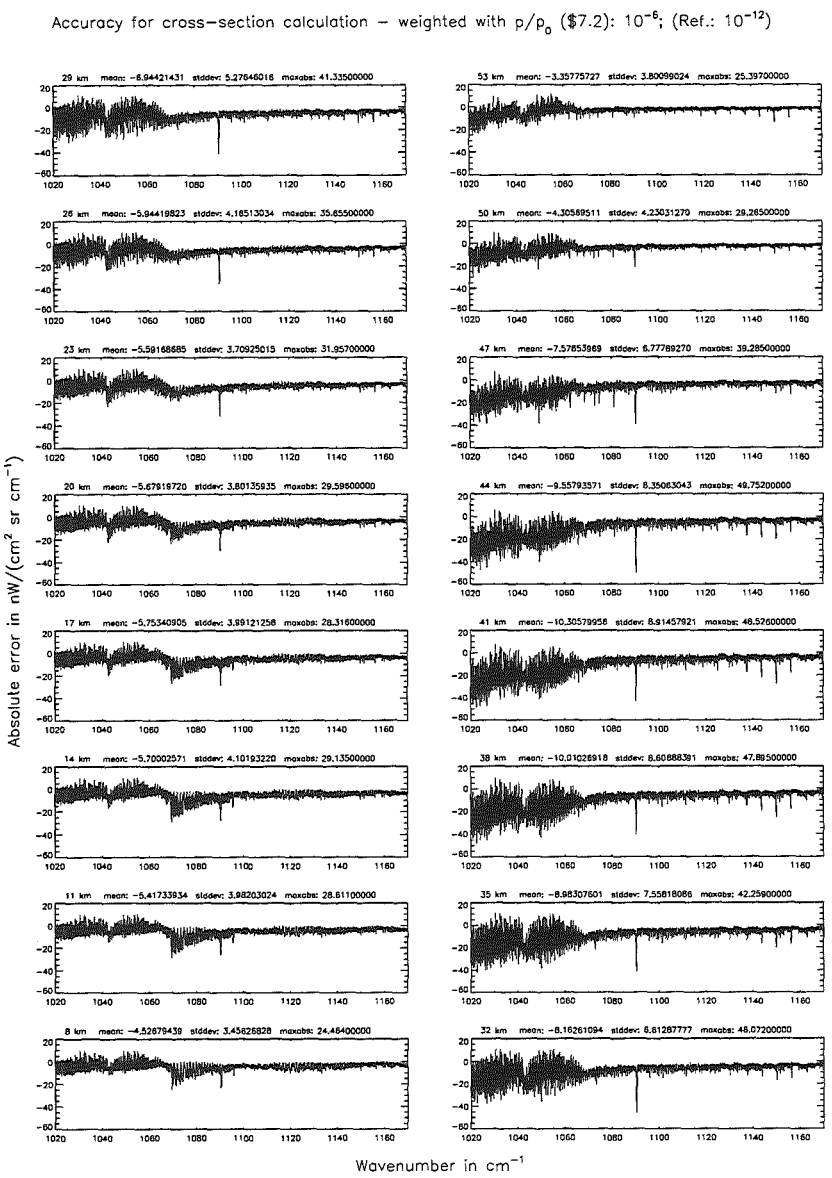


Figure 82: Relative error [%] with respect to MIPAS NESR



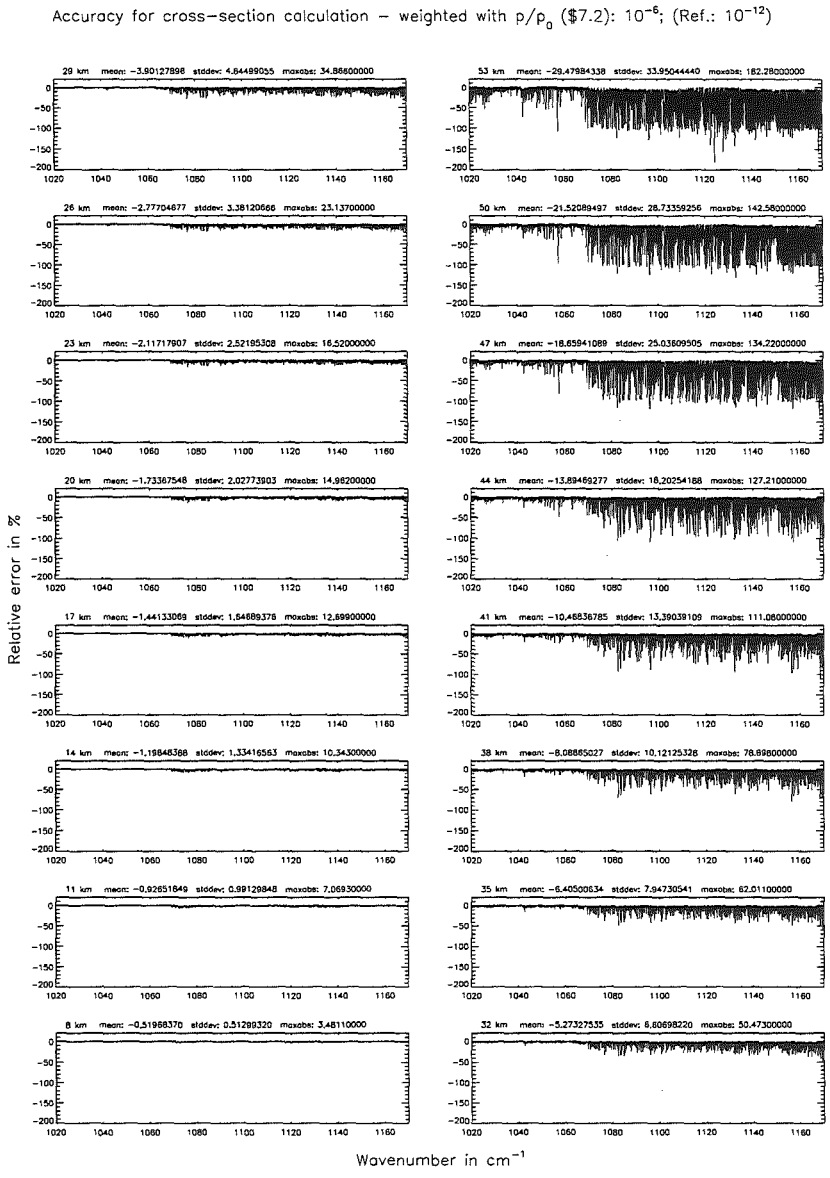


Figure 83: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

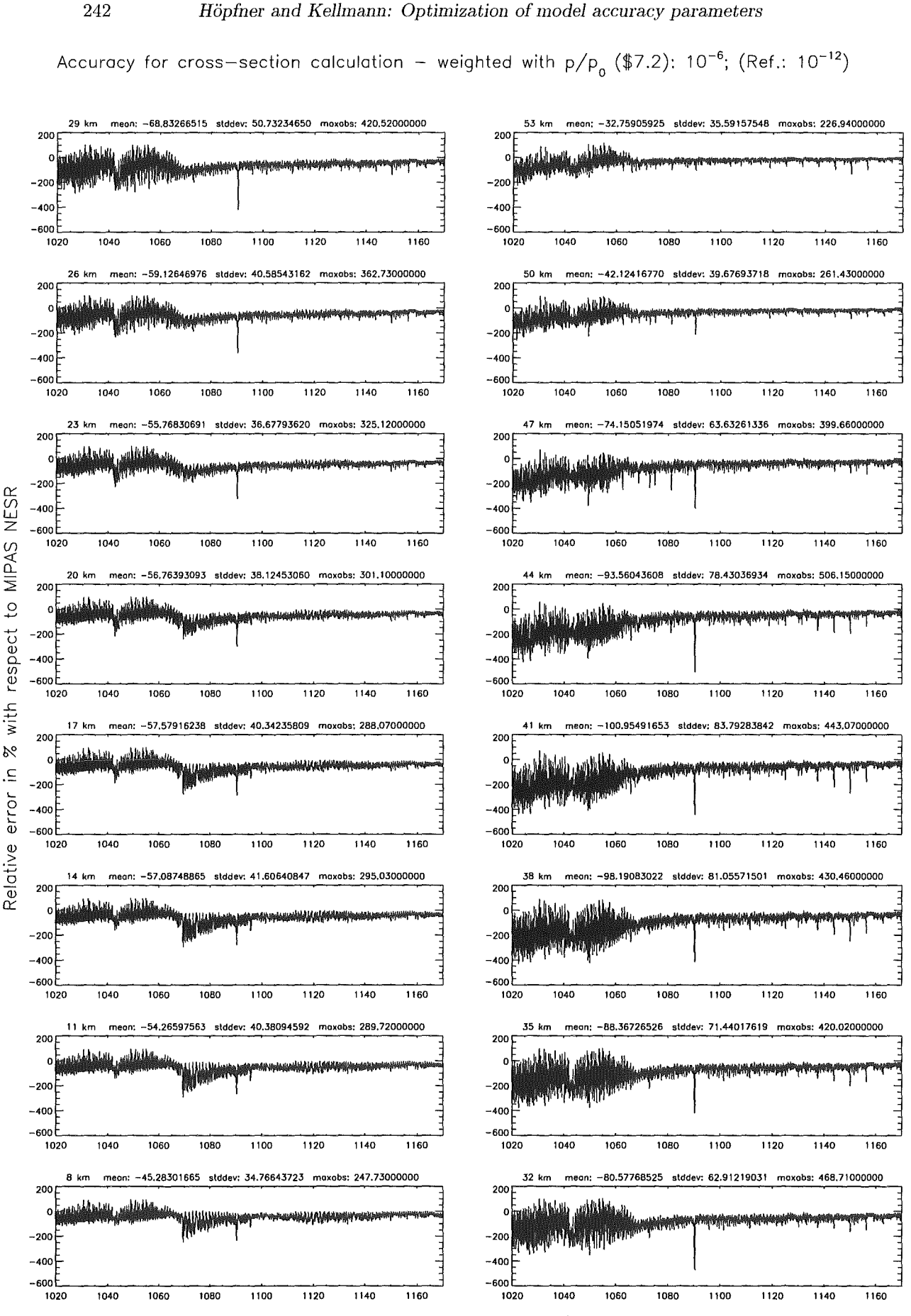
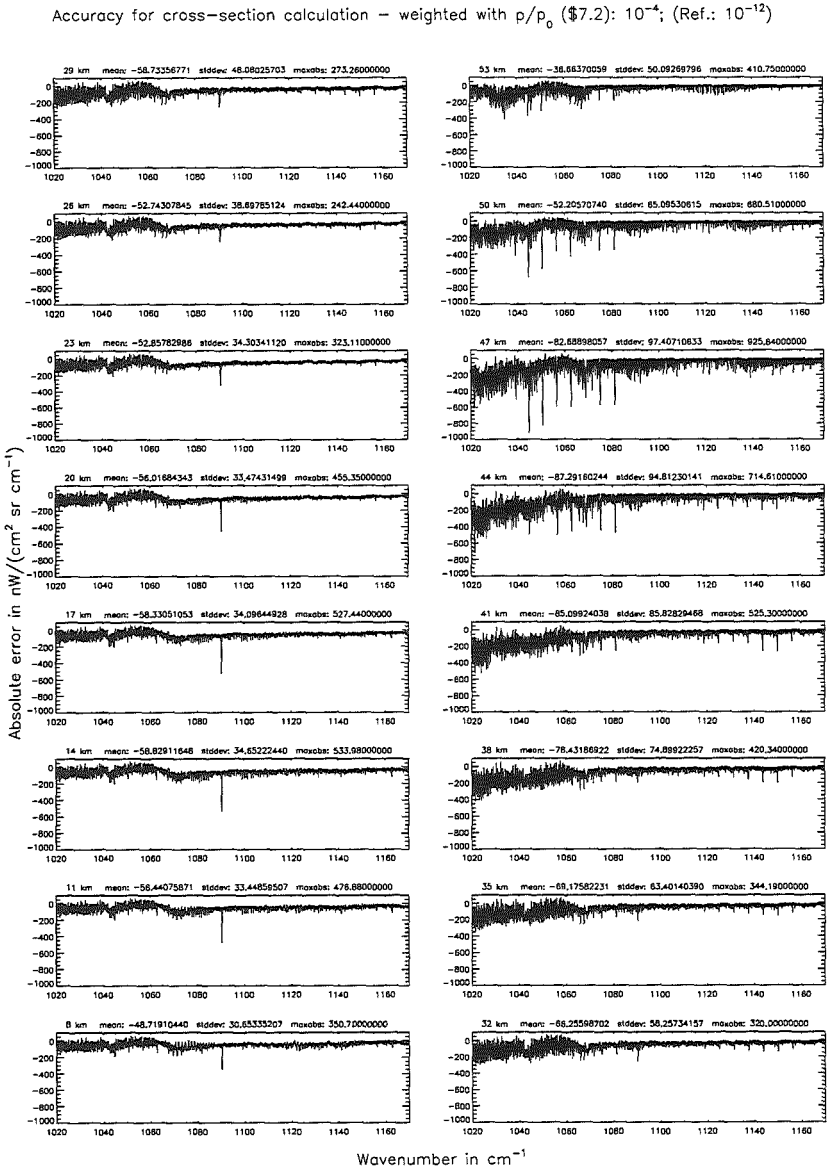


Figure 84: Relative error [%] with respect to MIPAS NESR



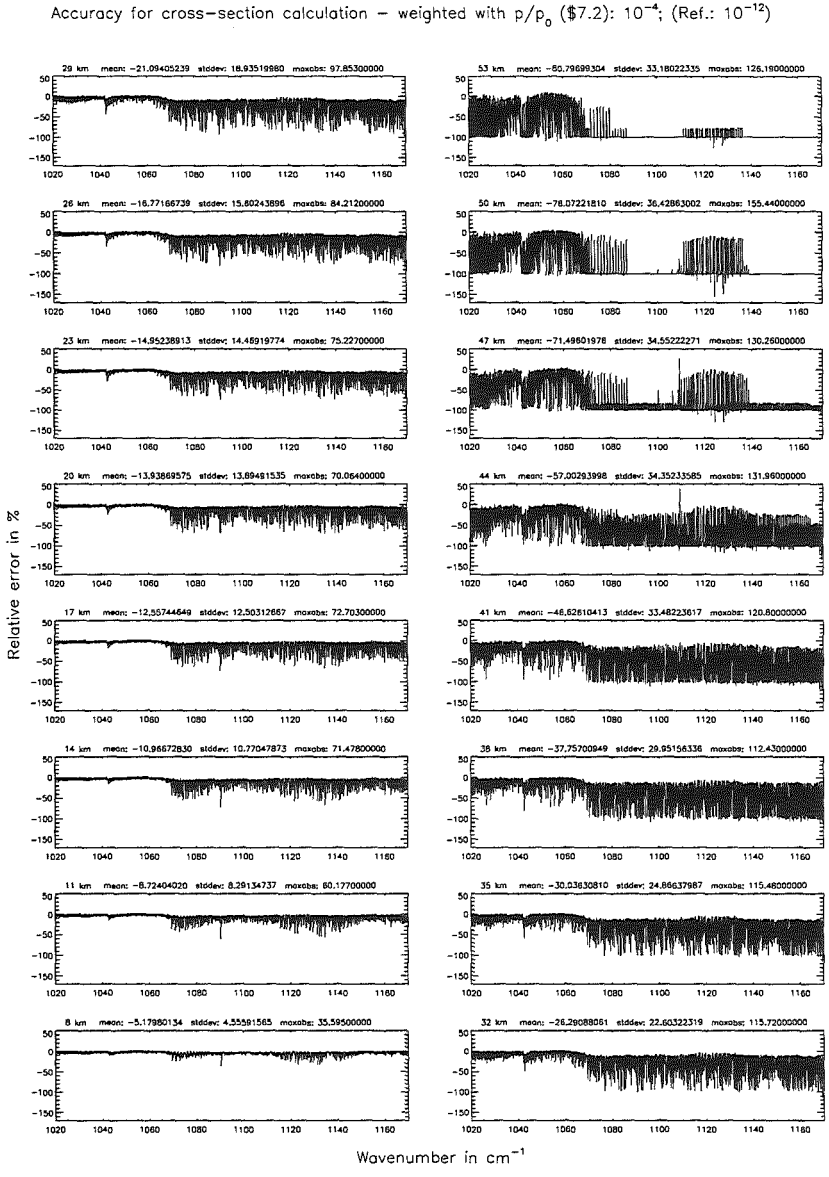


Figure 85: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

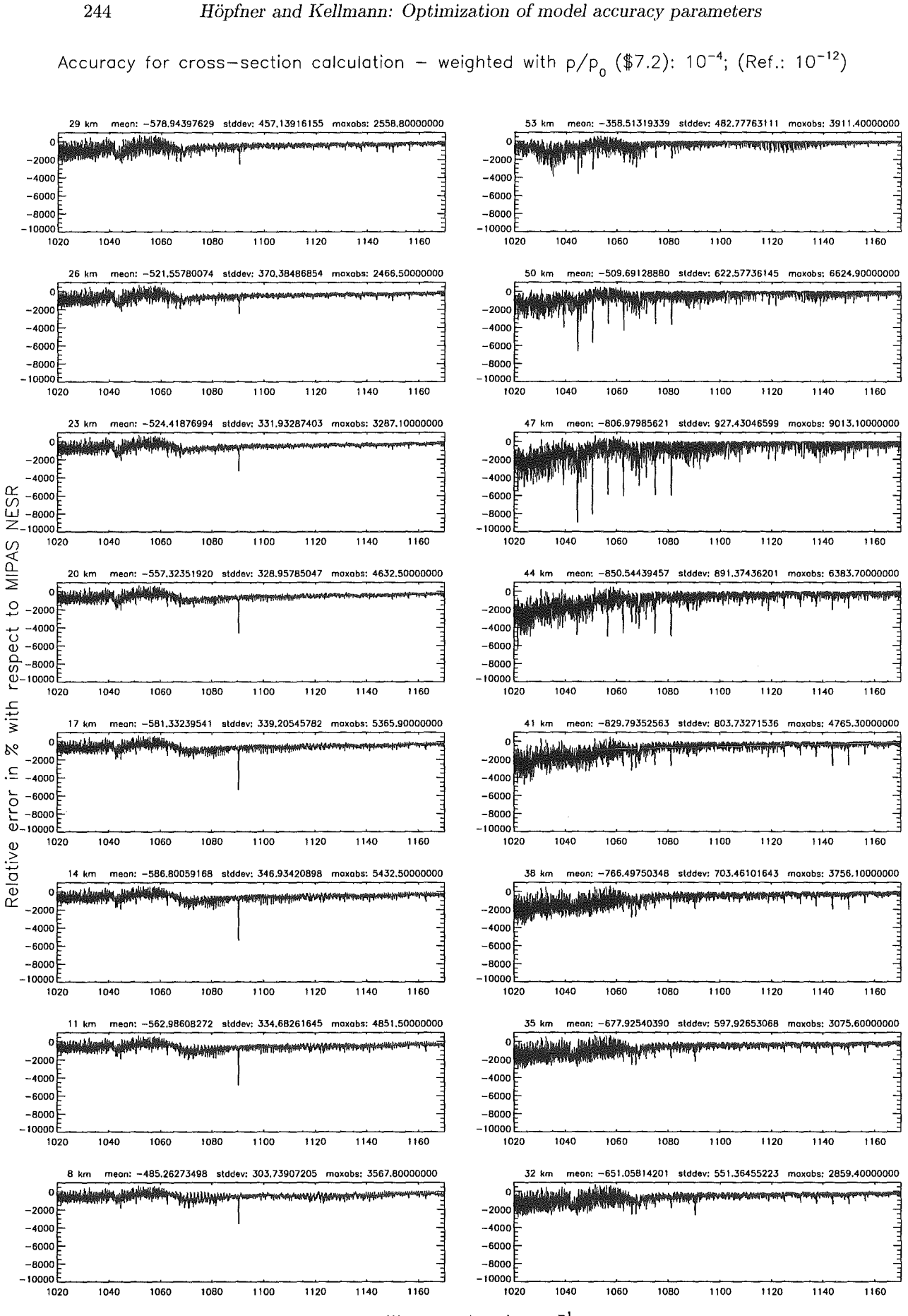


Figure 86: Relative error [%] with respect to MIPAS NESR

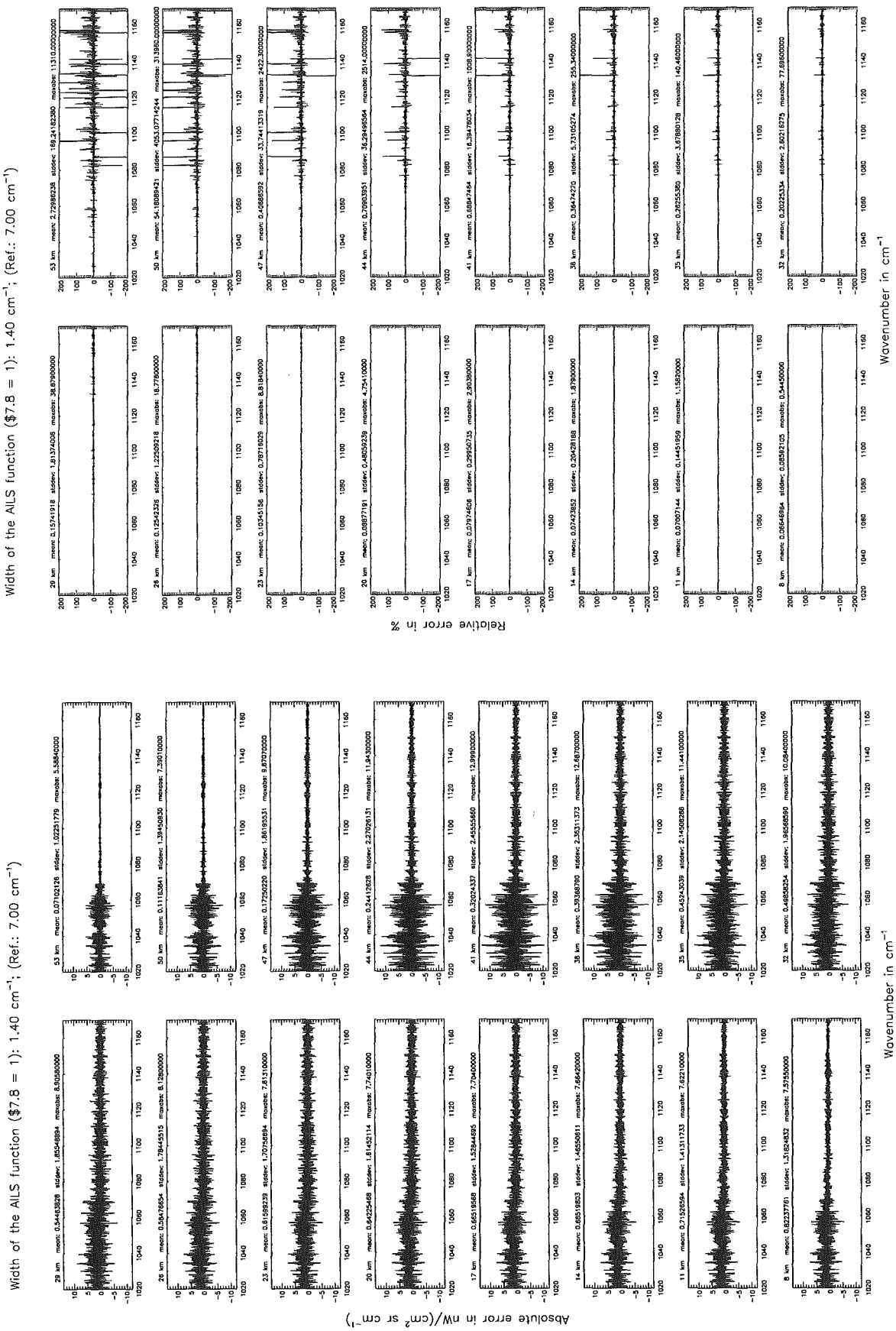


Figure 87: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Width of the AILS function ($7.8 = 1): 1.40 cm^{-1}; (Ref.: 7.00 cm^{-1})

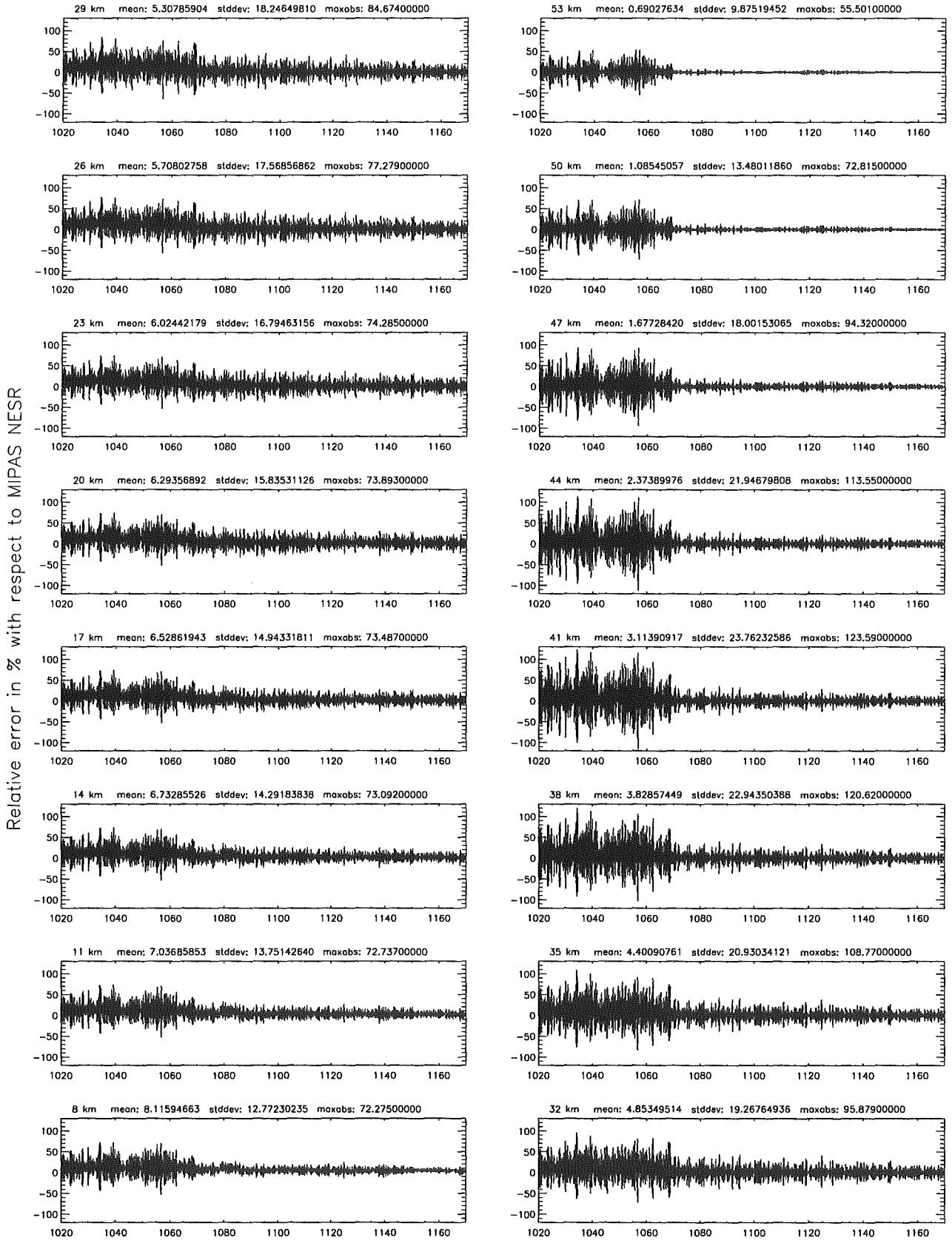


Figure 88: Relative error [%] with respect to MIPAS NESR

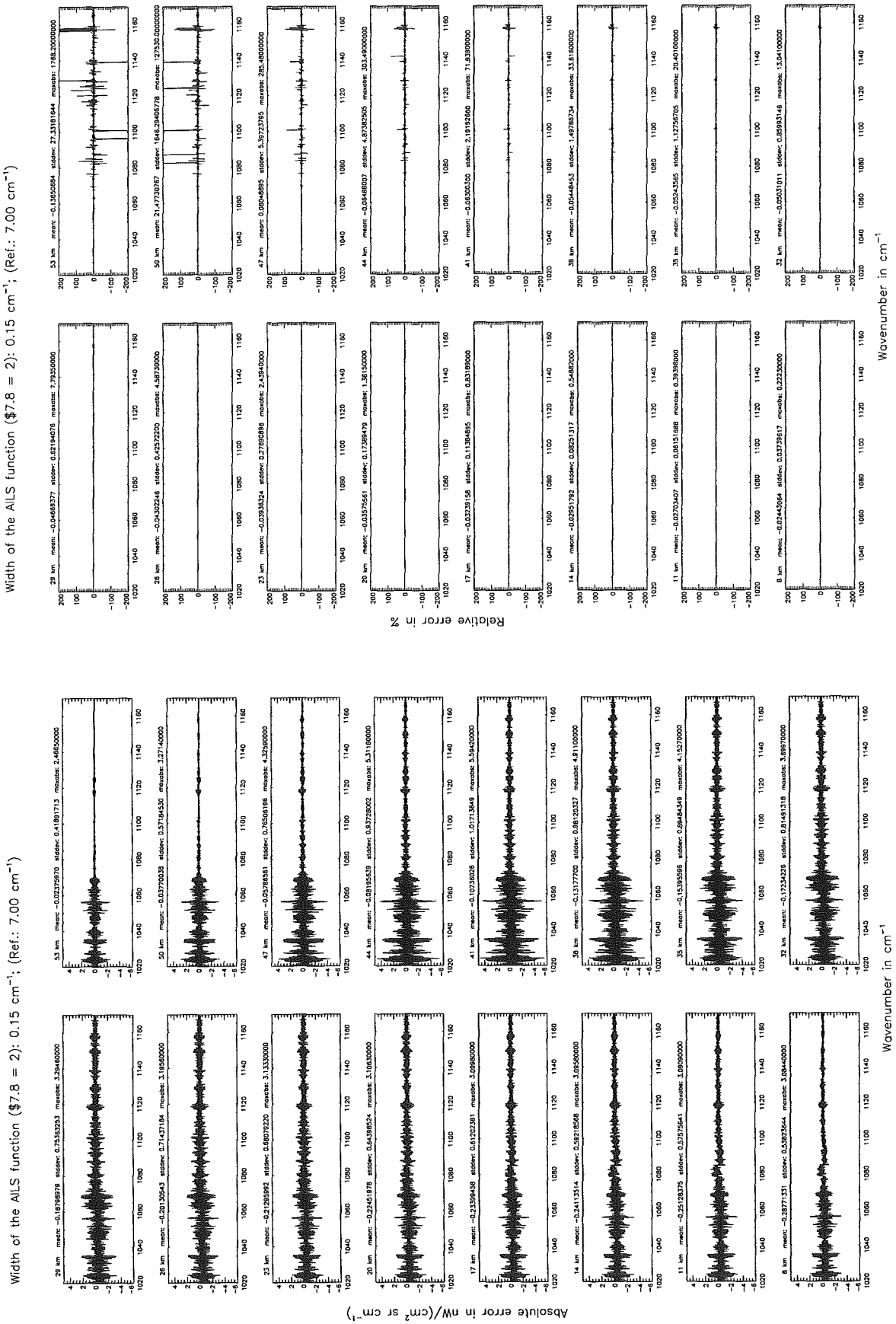


Figure 89: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

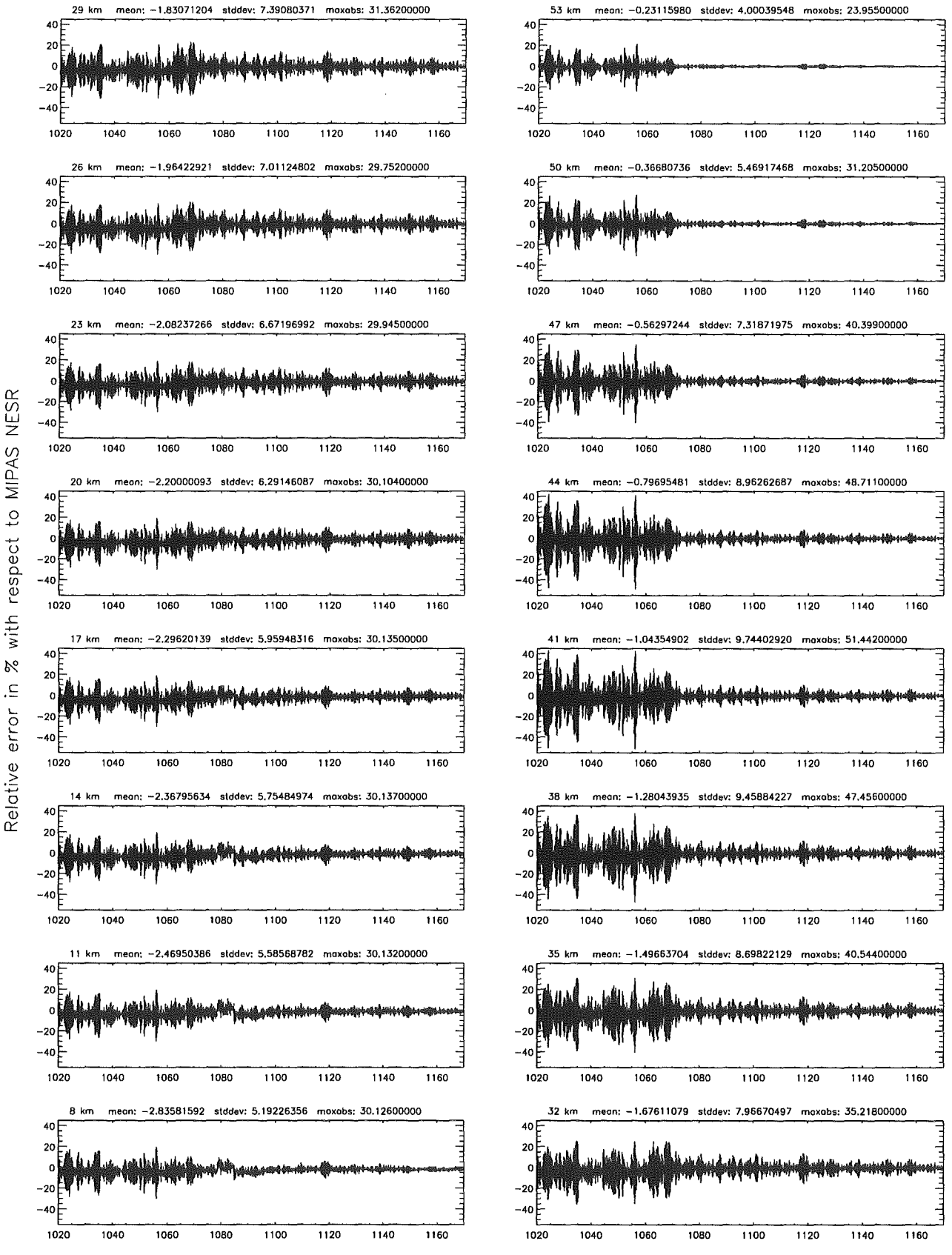


Figure 90: Relative error [%] with respect to MIPAS NESR

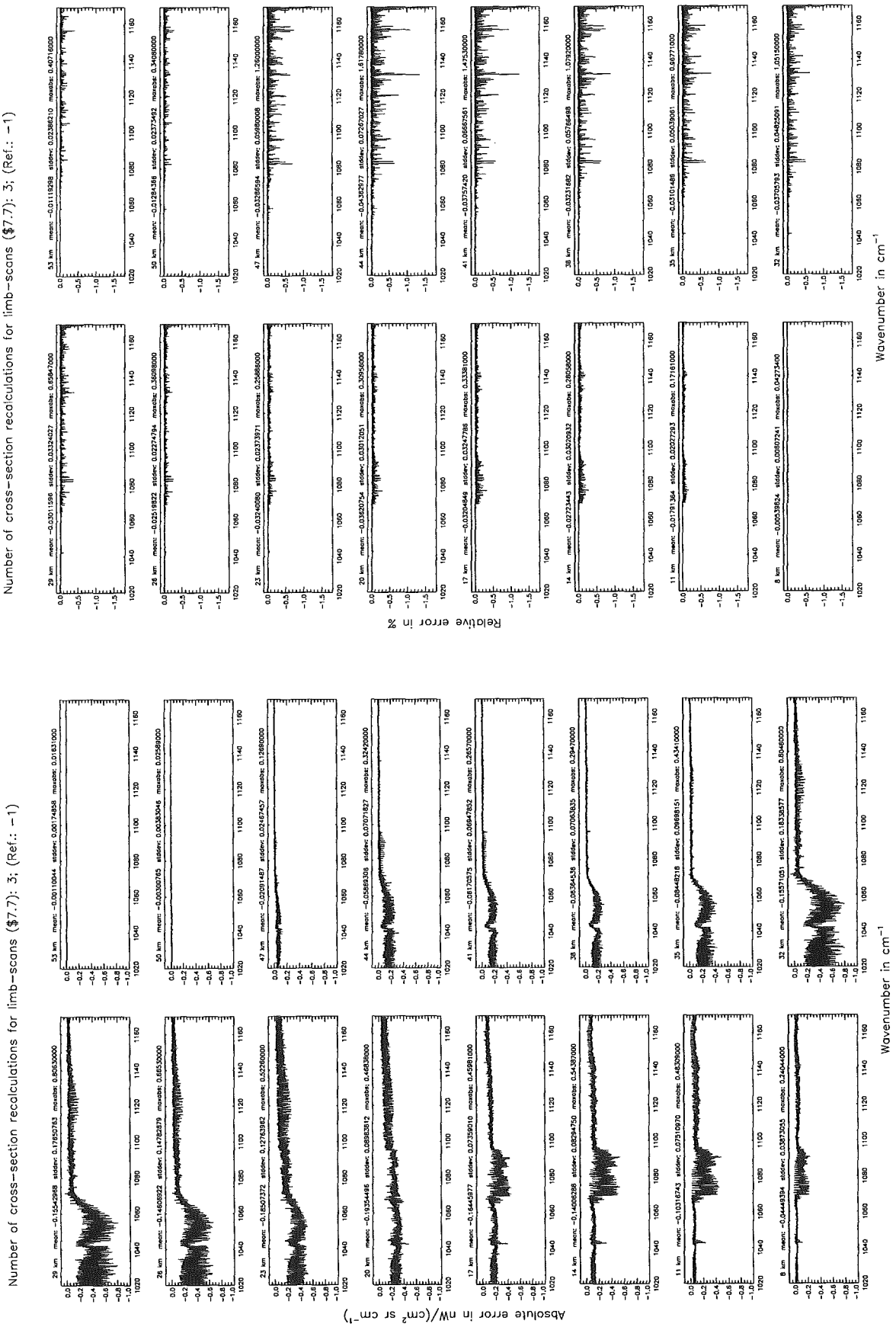


Figure 91: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

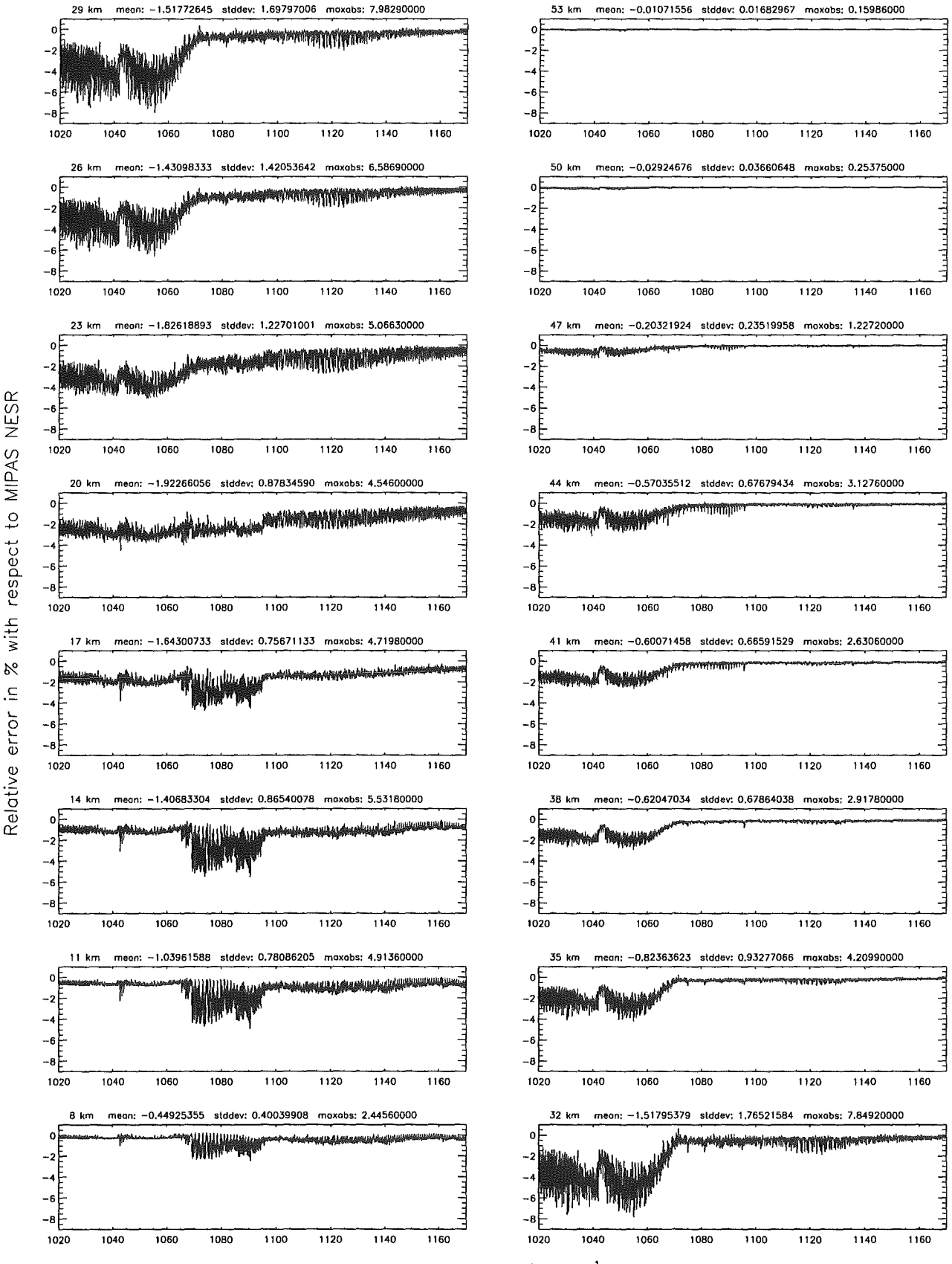


Figure 92: Relative error [%] with respect to MIPAS NESR

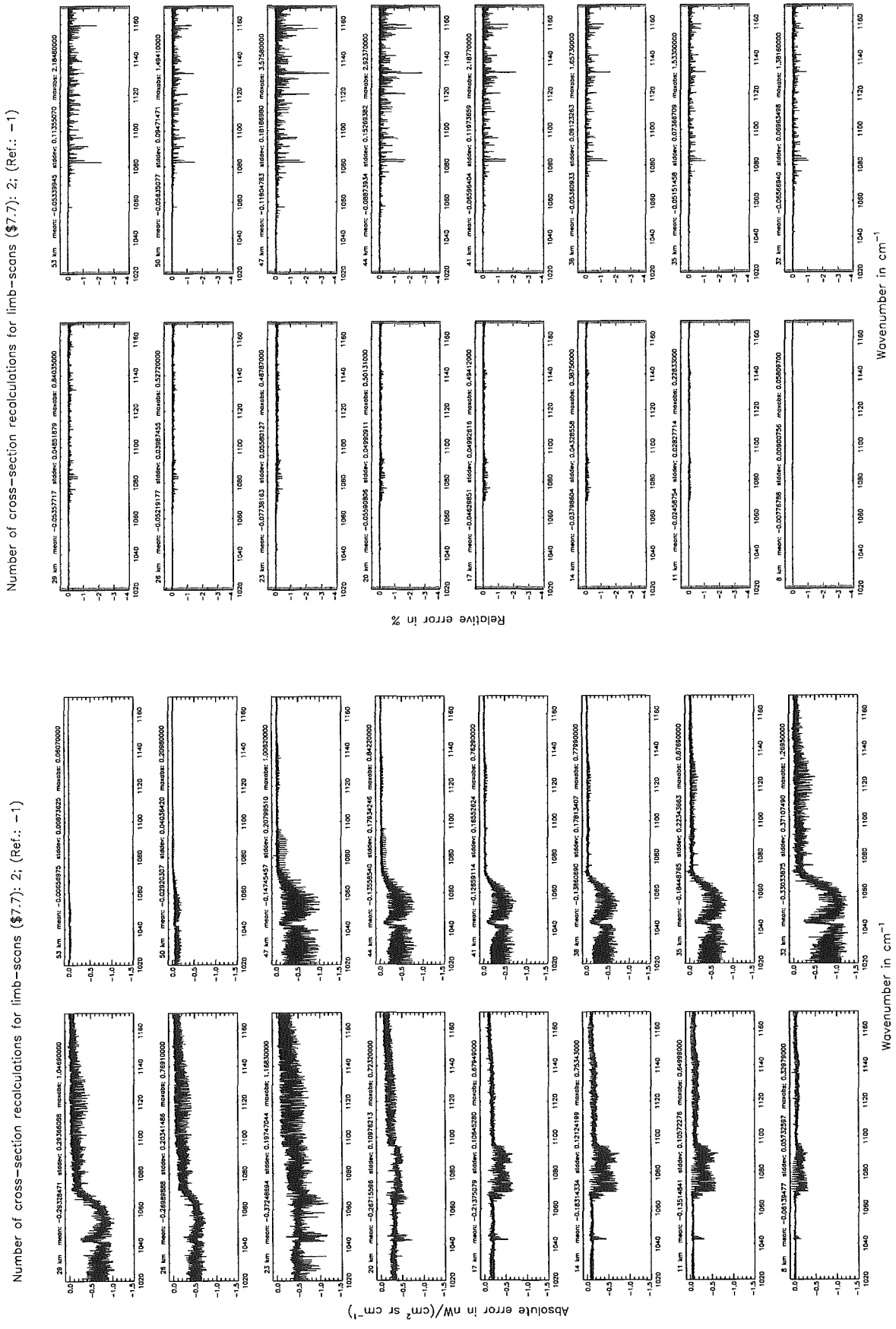


Figure 93: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

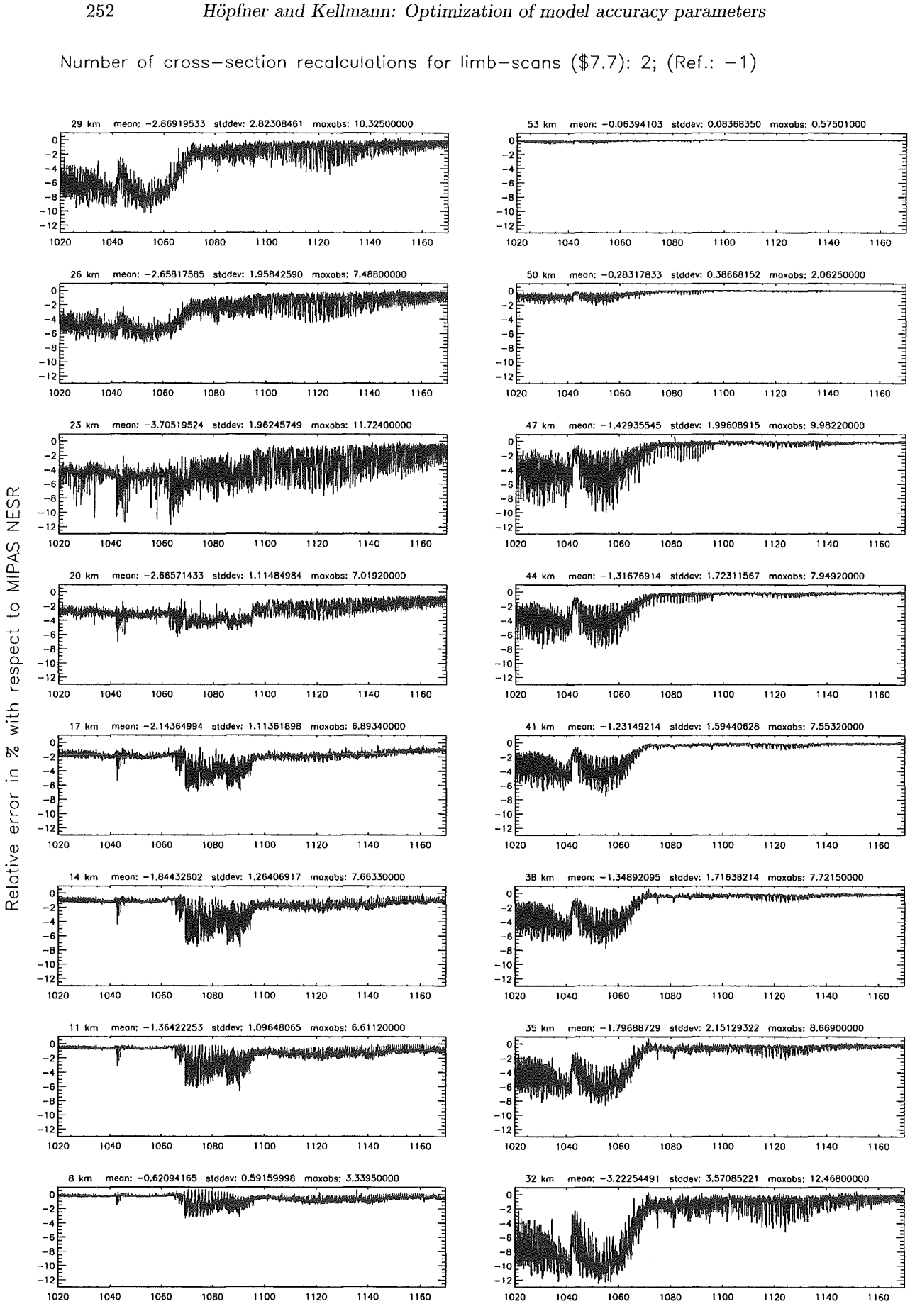
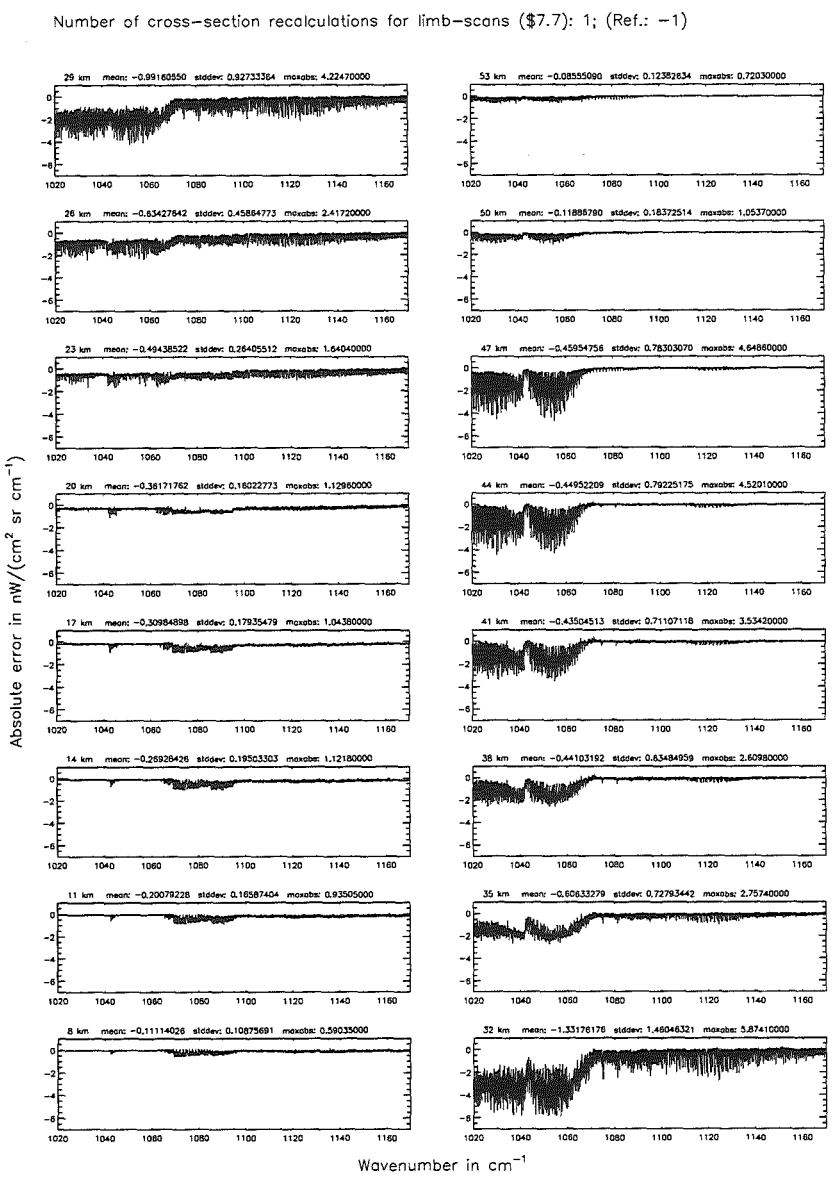


Figure 94: Relative error [%] with respect to MIPAS NESR



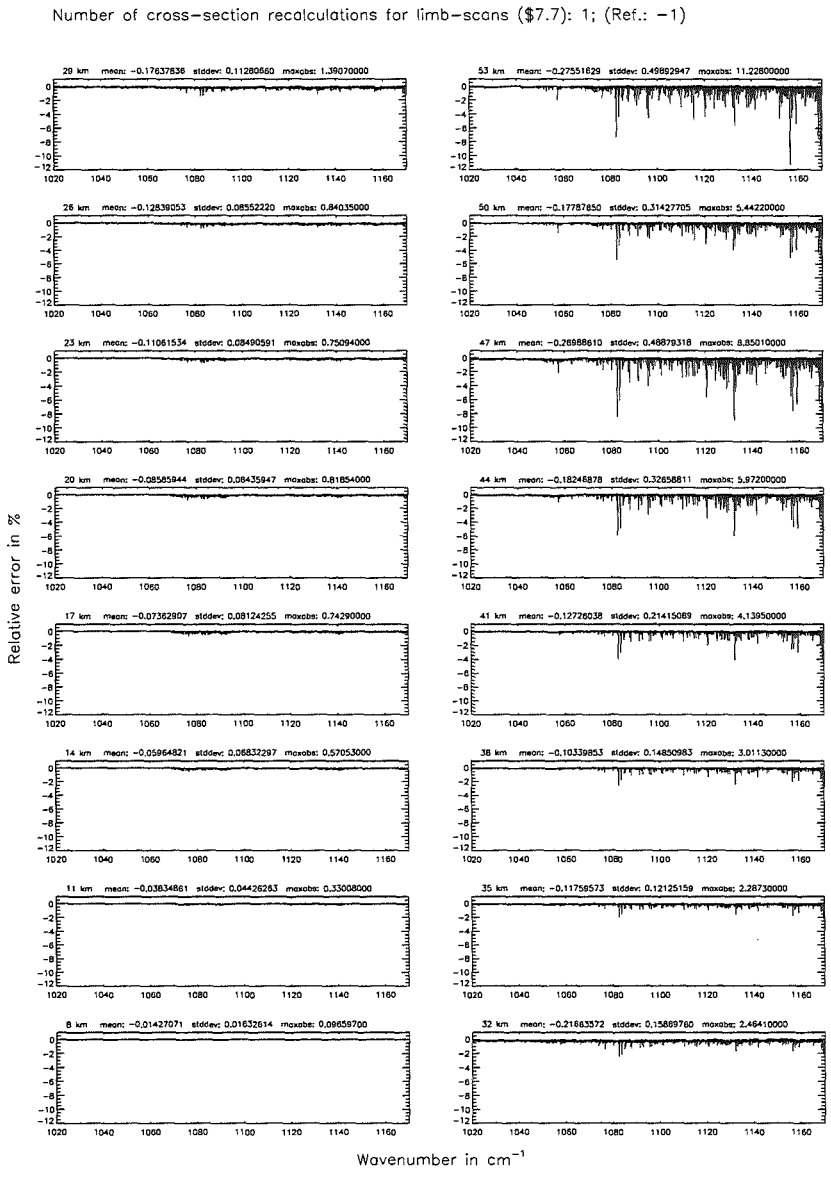


Figure 95: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

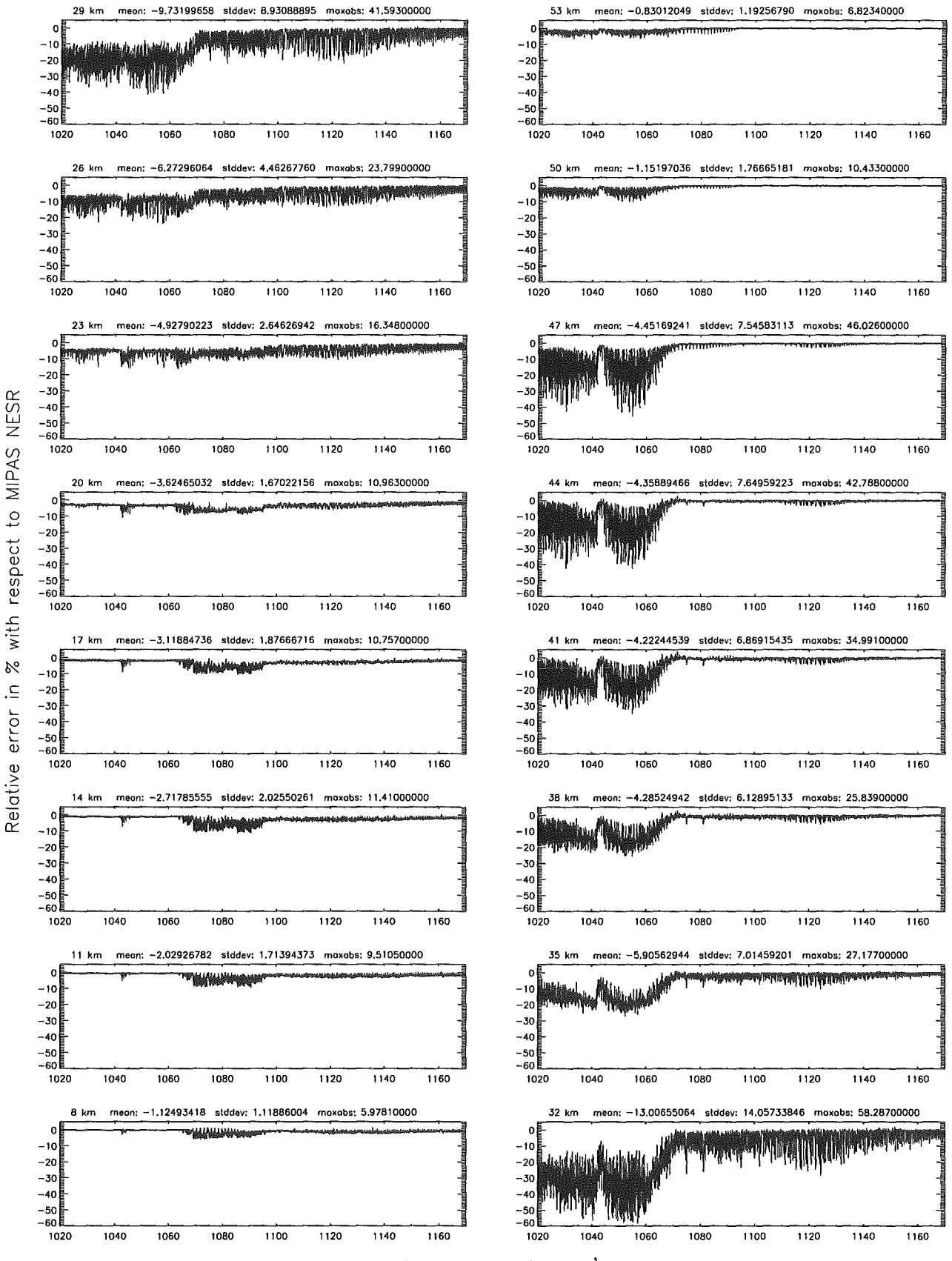


Figure 96: Relative error [%] with respect to MIPAS NESR

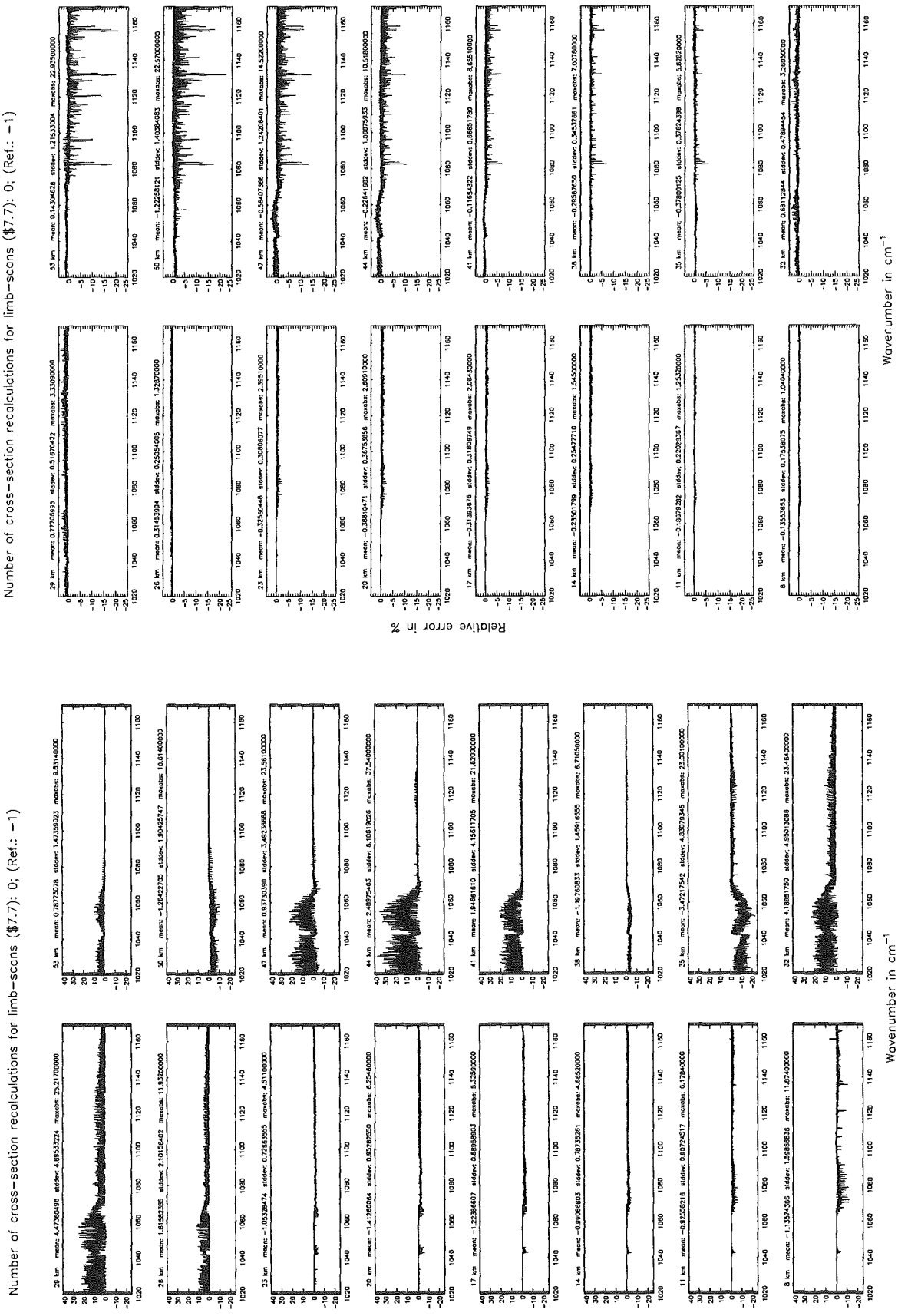


Figure 97: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

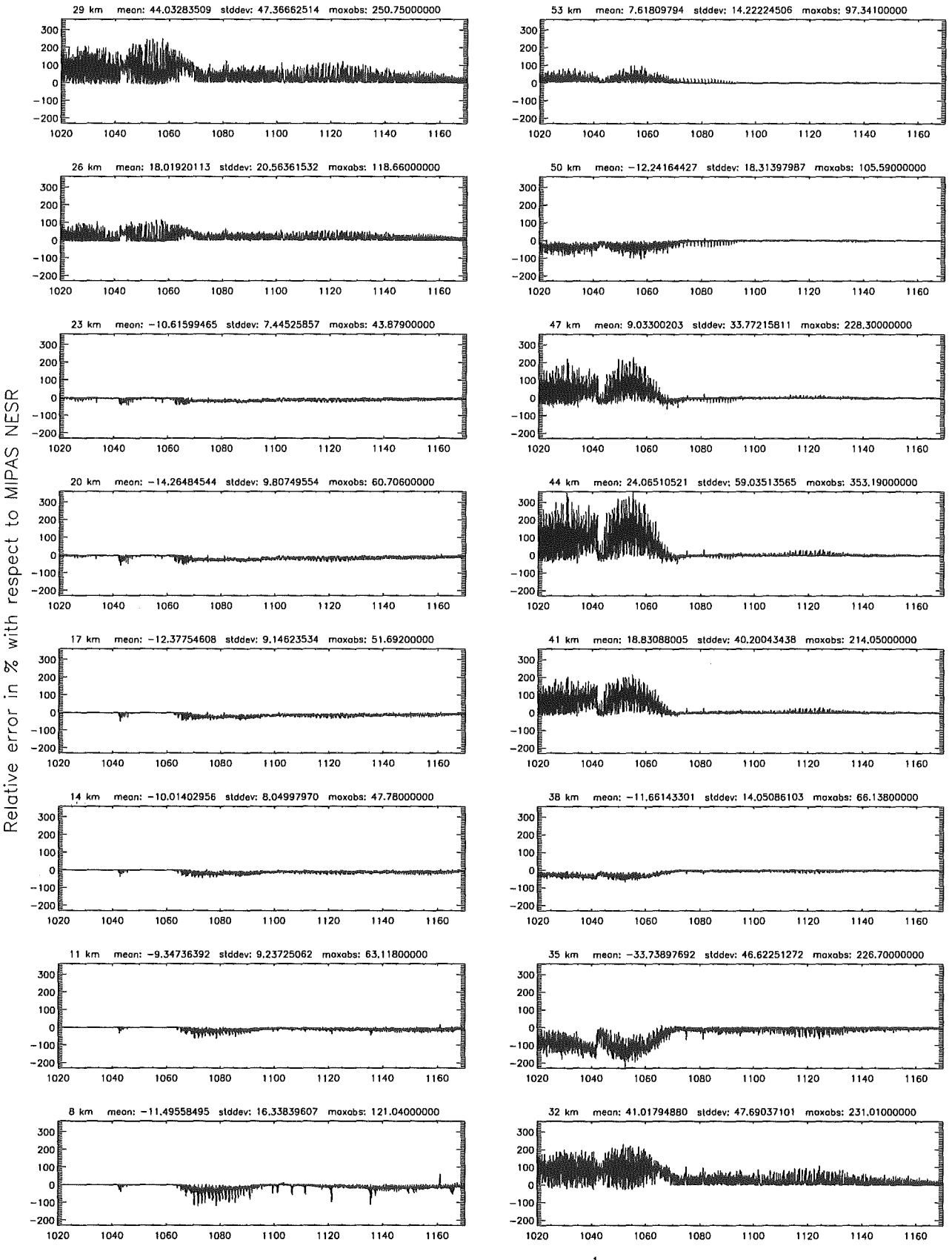


Figure 98: Relative error [%] with respect to MIPAS NESR

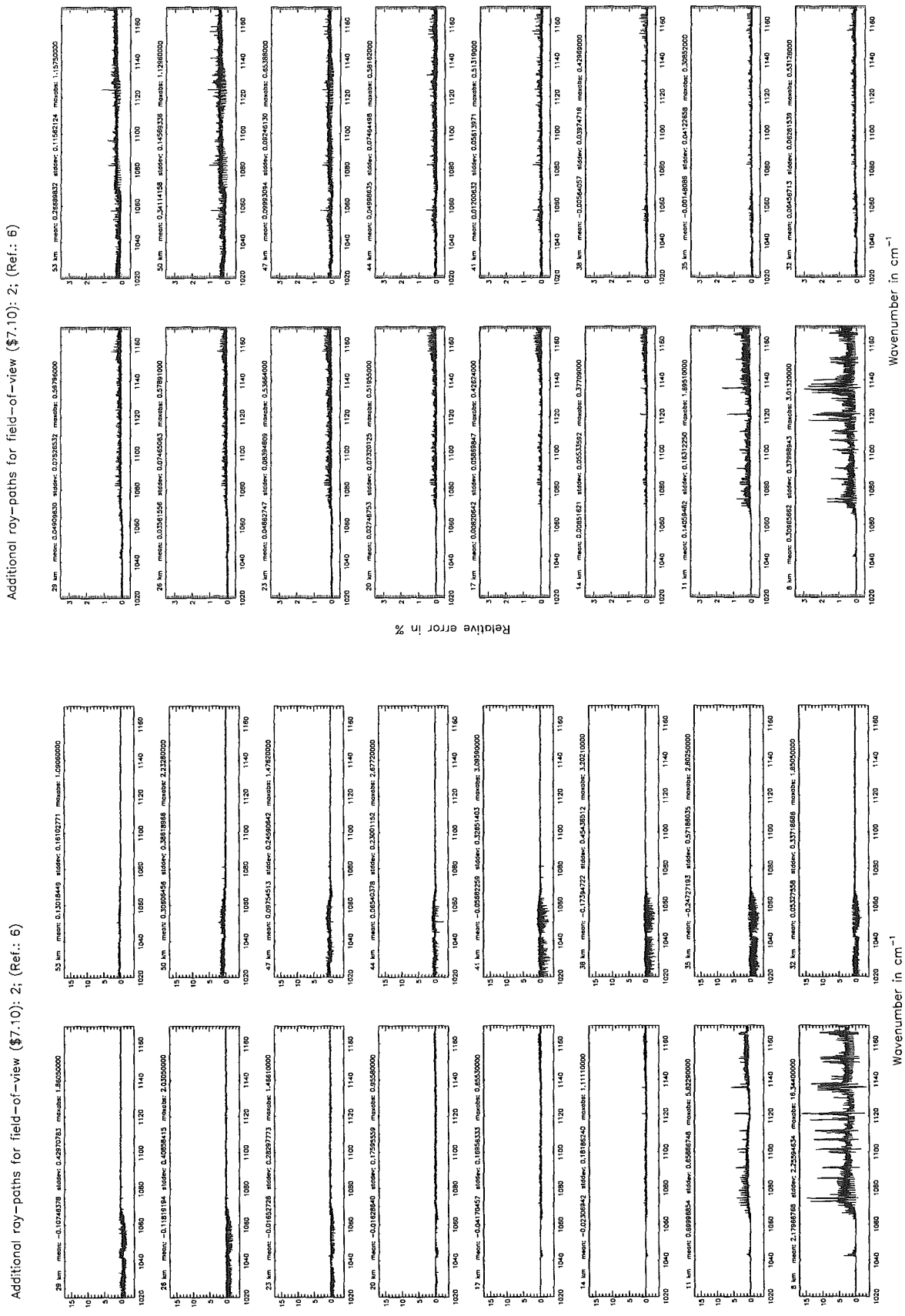


Figure 99: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

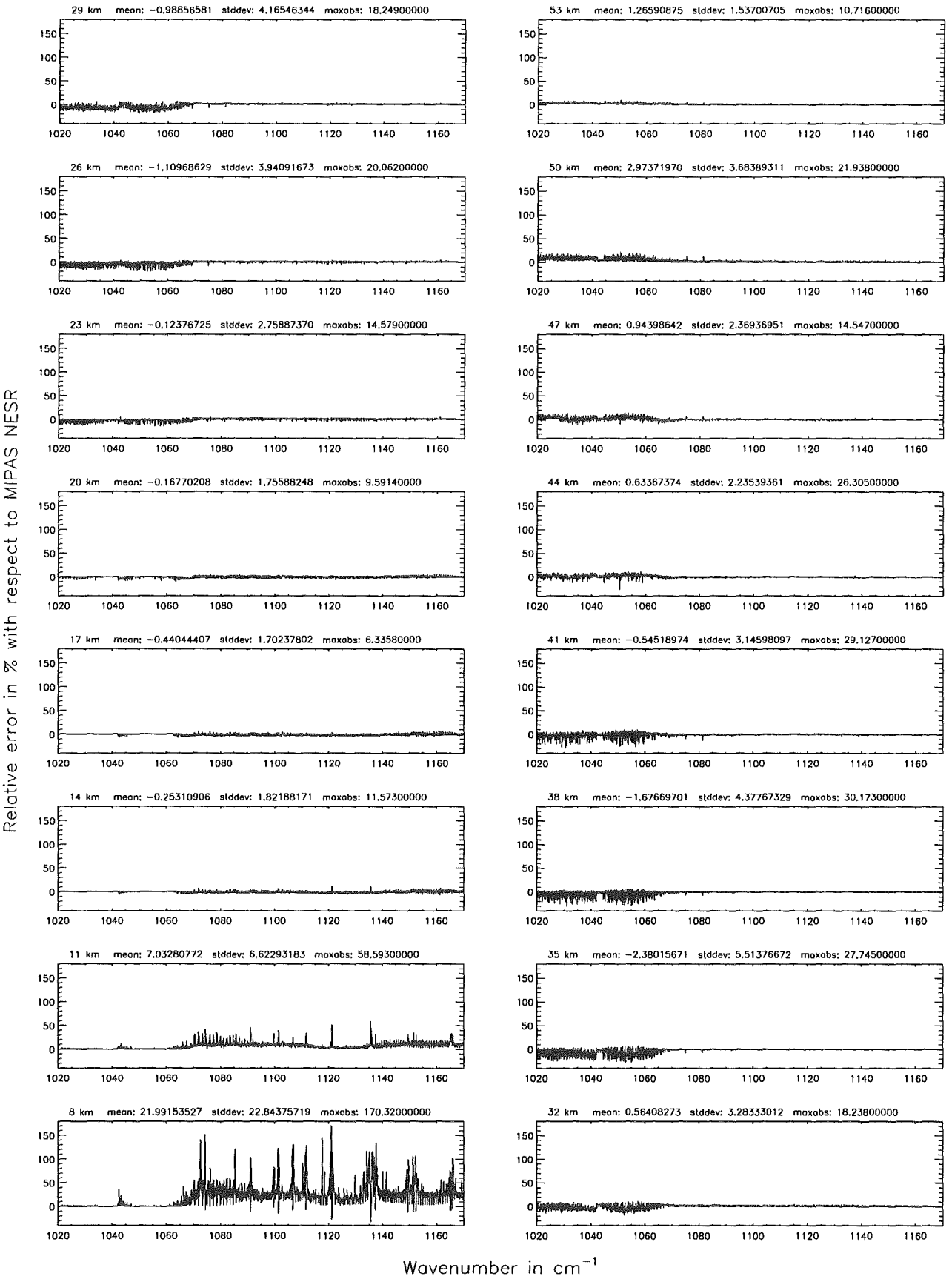


Figure 100: Relative error [%] with respect to MIPAS NESR

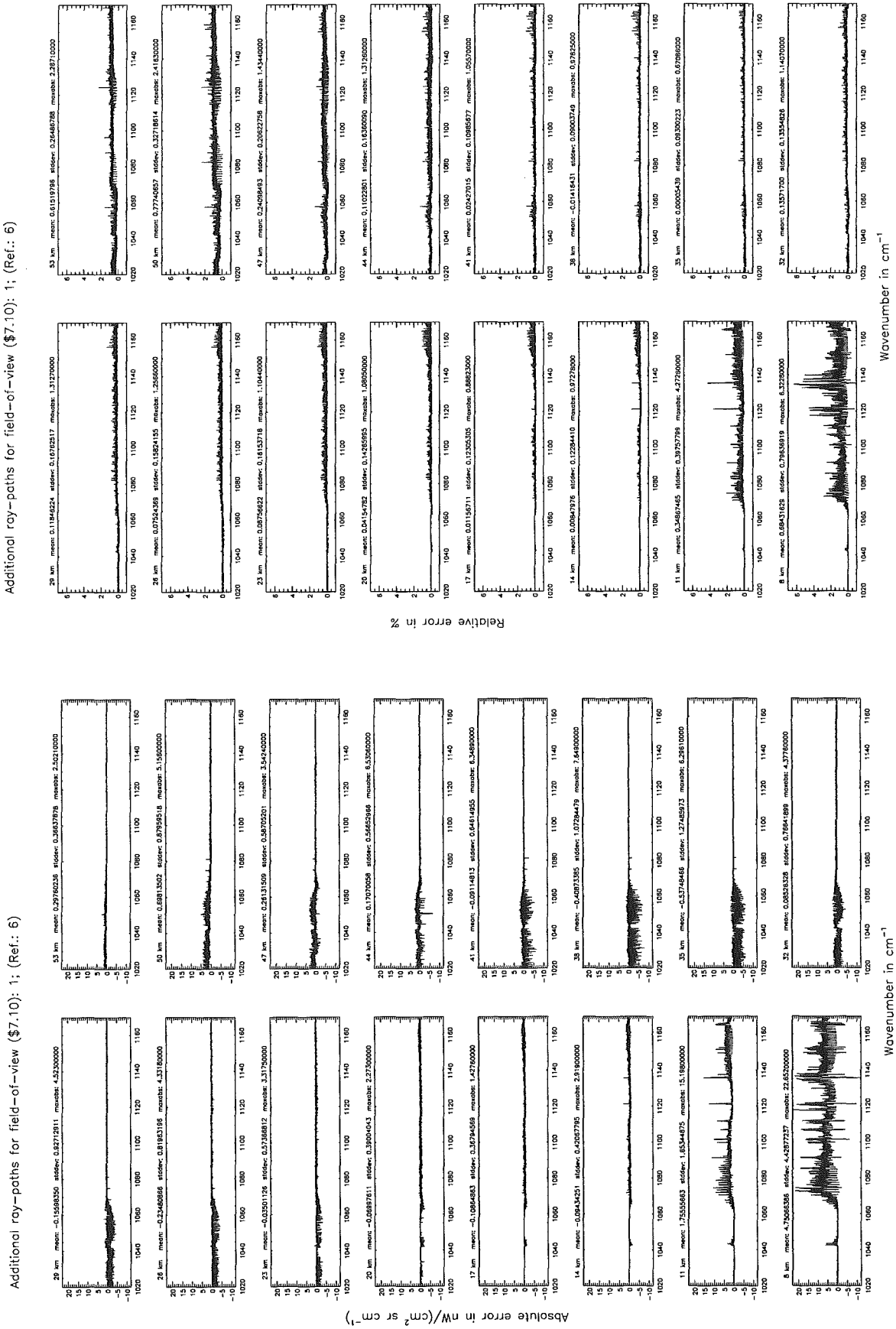


Figure 101: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

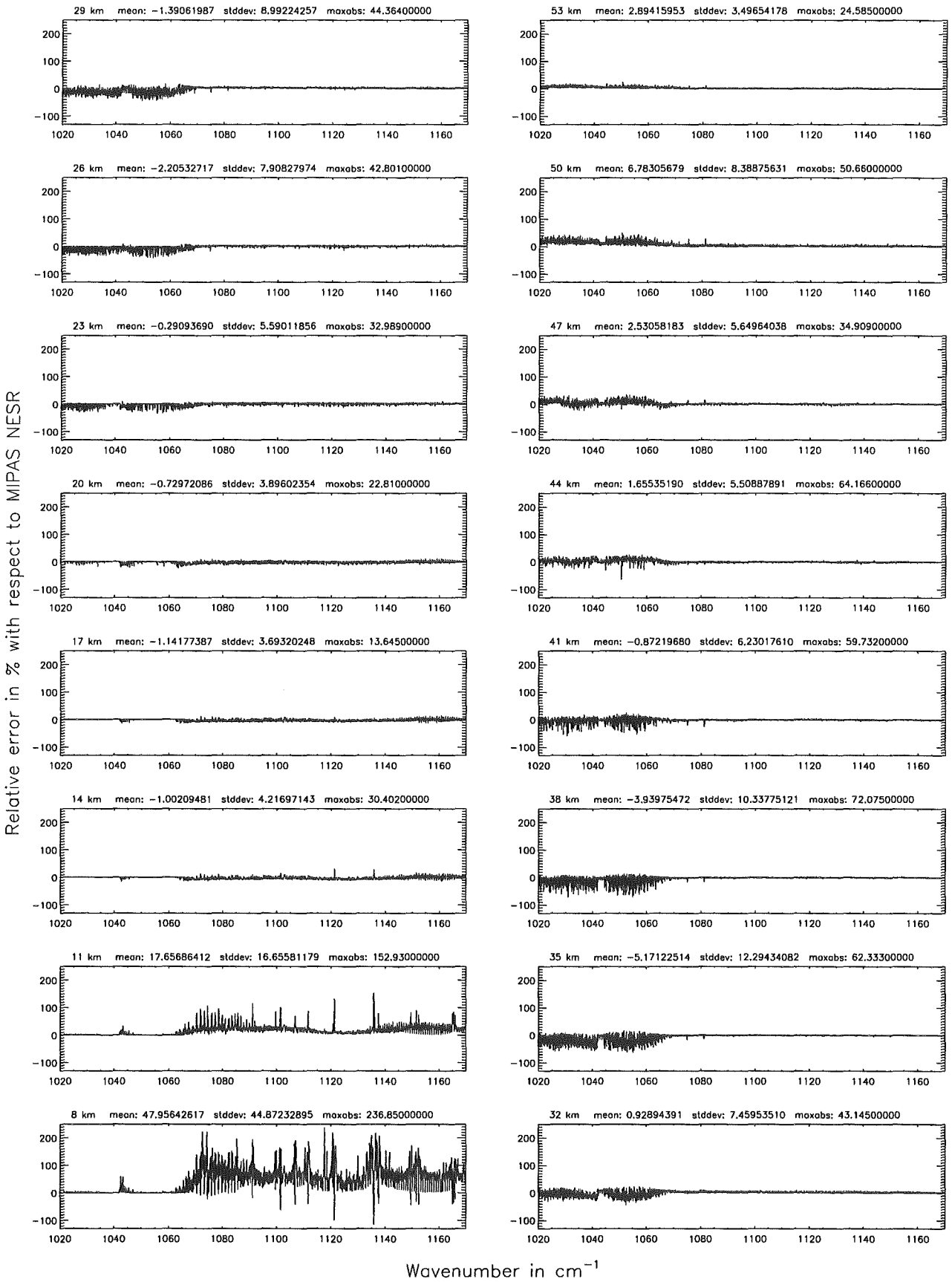


Figure 102: Relative error [%] with respect to MIPAS NESR

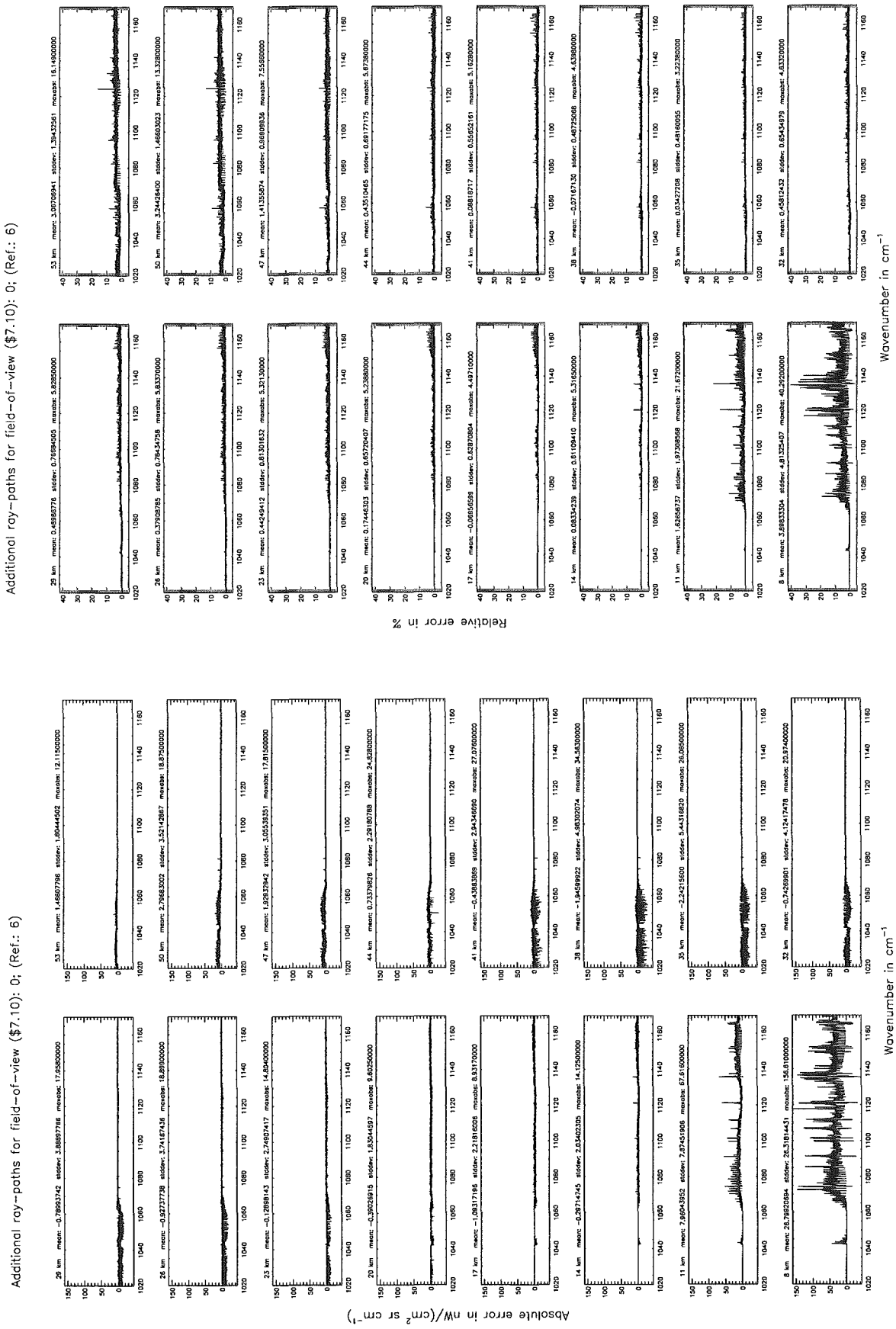


Figure 103: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

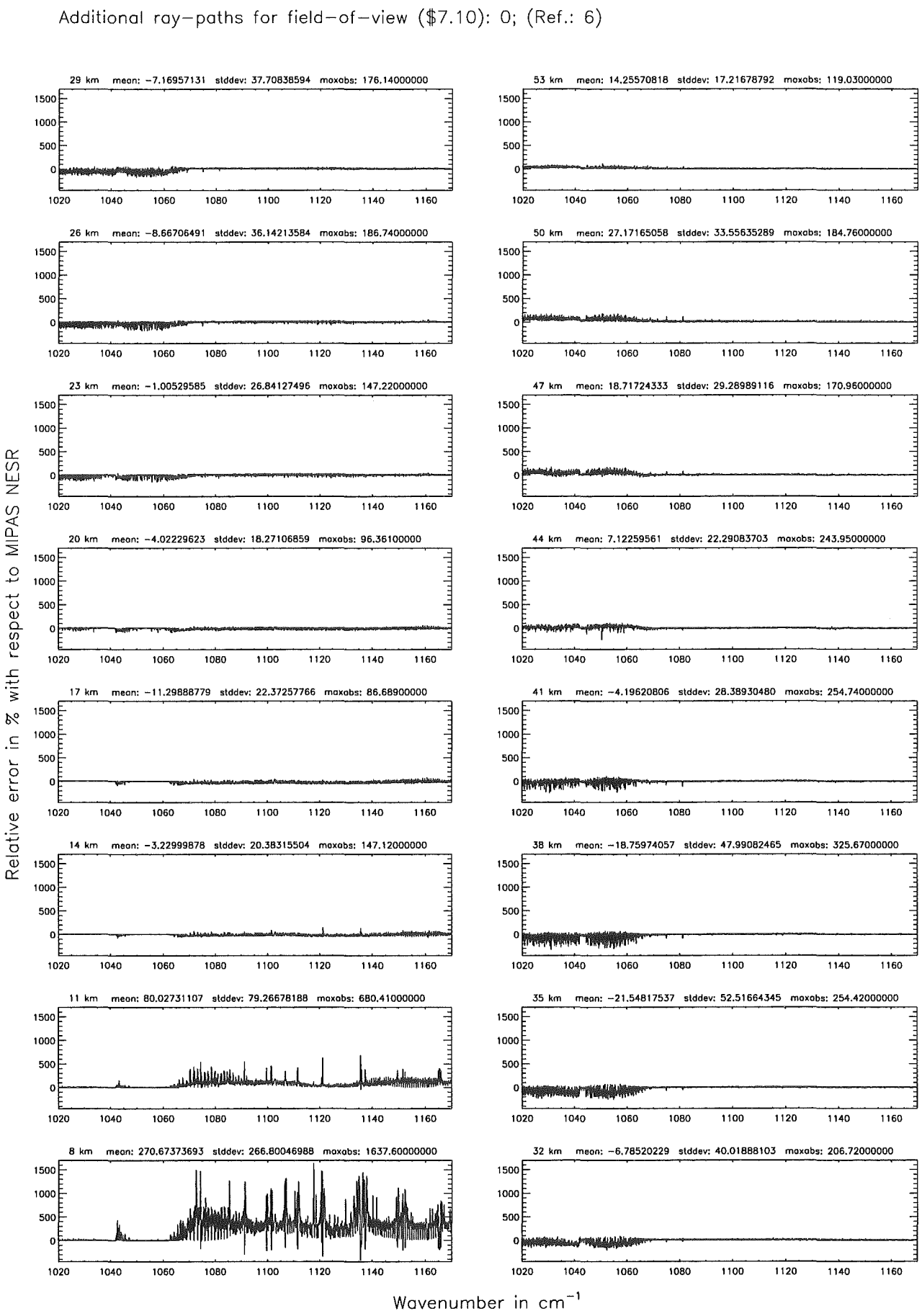


Figure 104: Relative error [%] with respect to MIPAS NESR

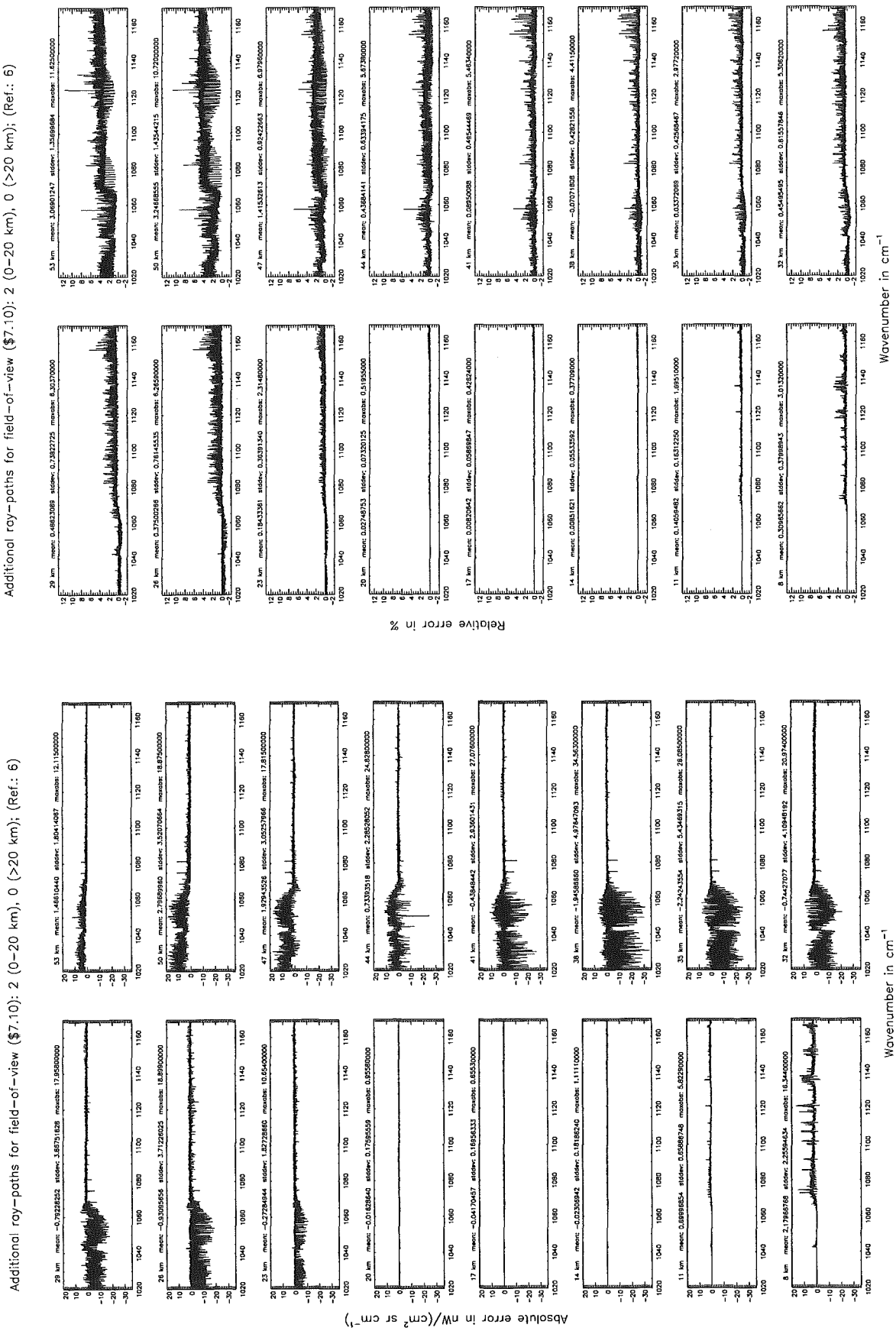


Figure 105: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

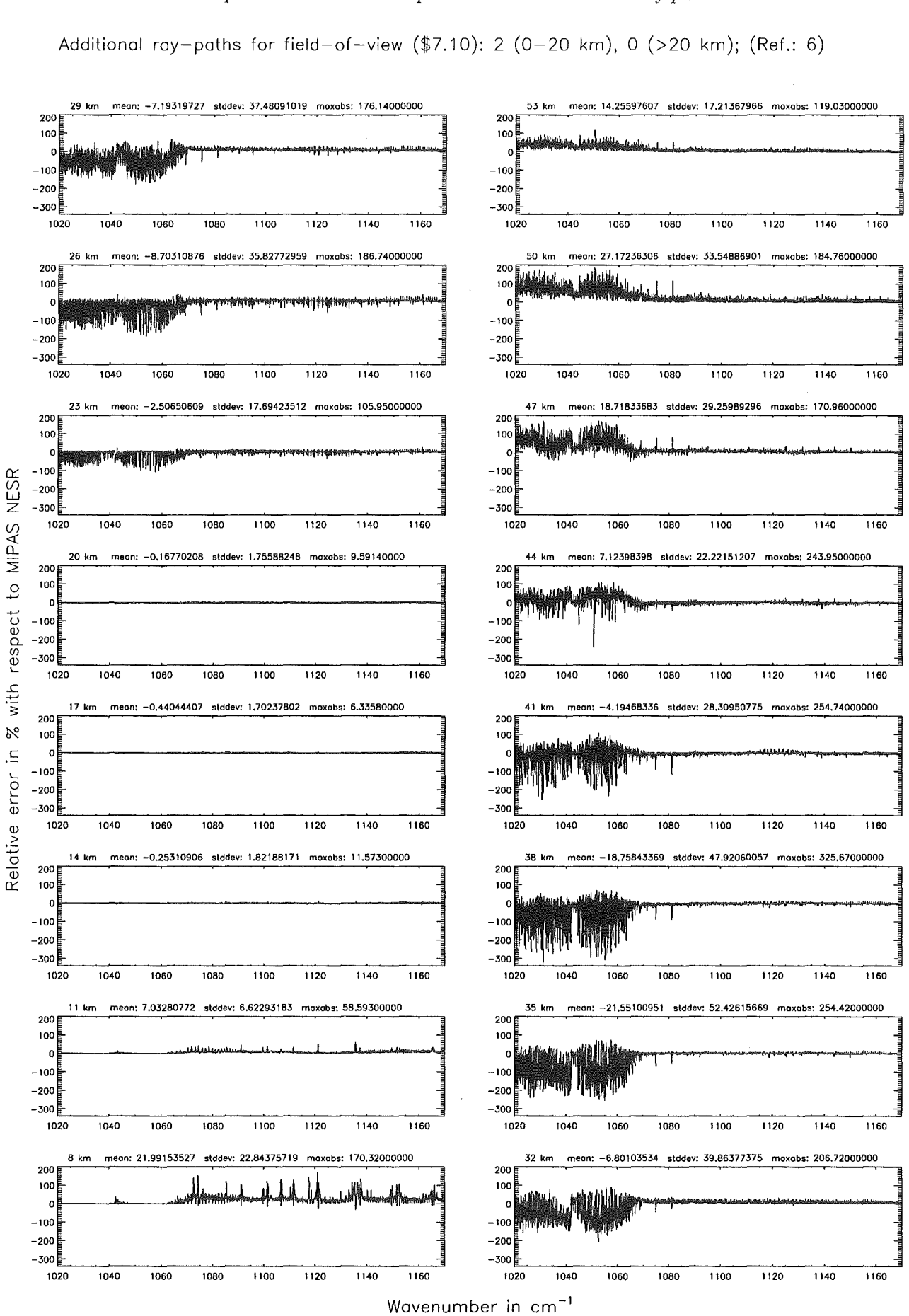
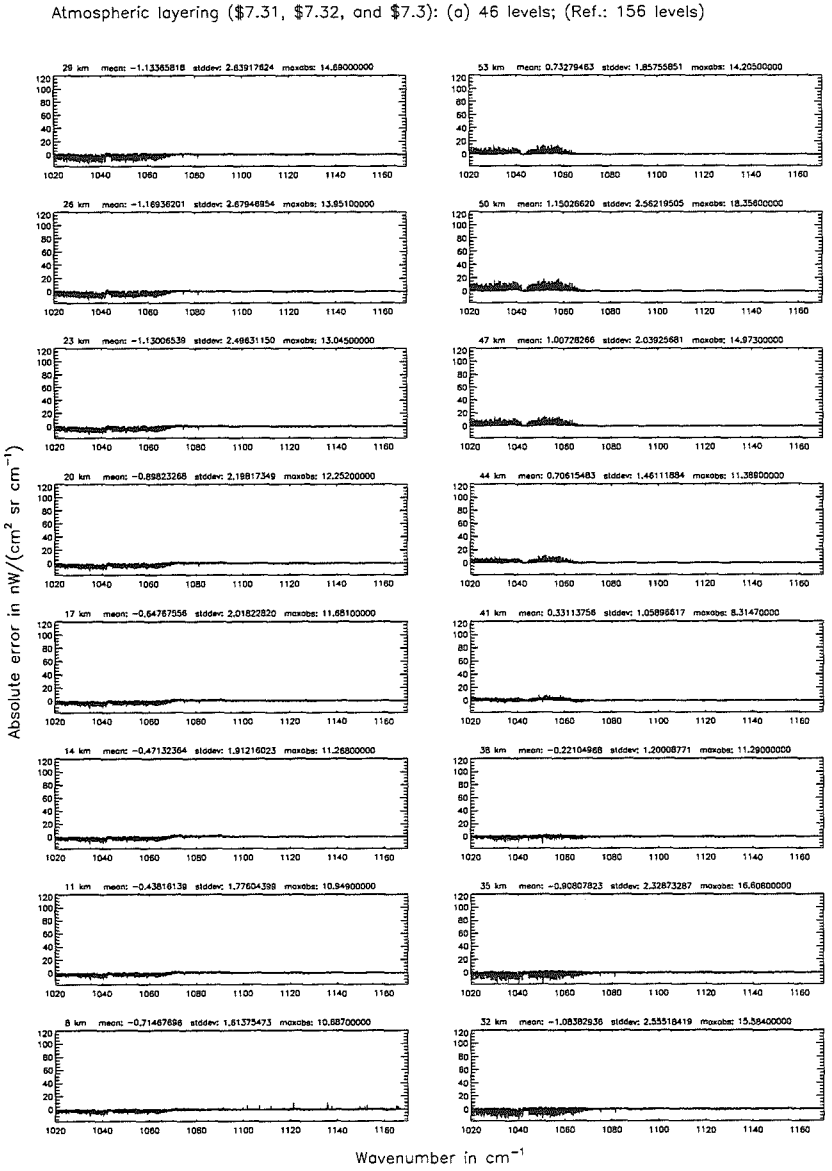


Figure 106: Relative error [%] with respect to MIPAS NESR



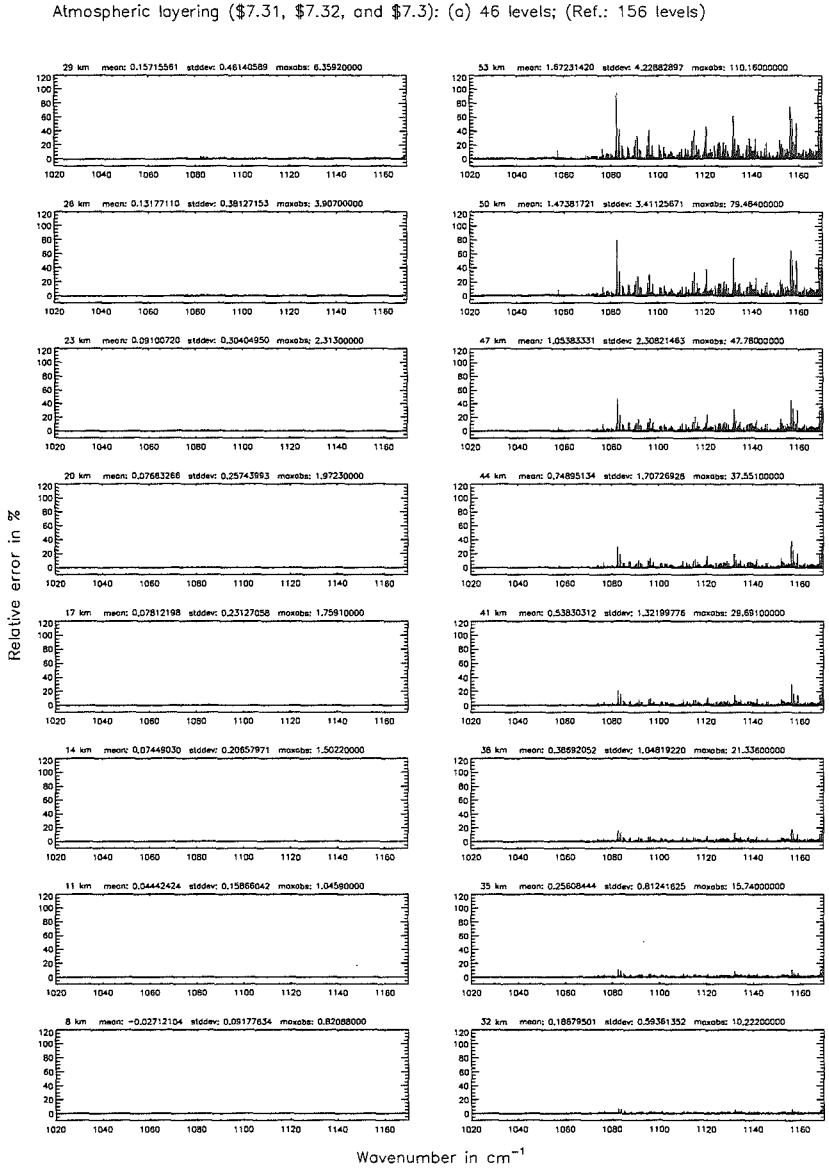


Figure 107: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (a) 46 levels; (Ref.: 156 levels)

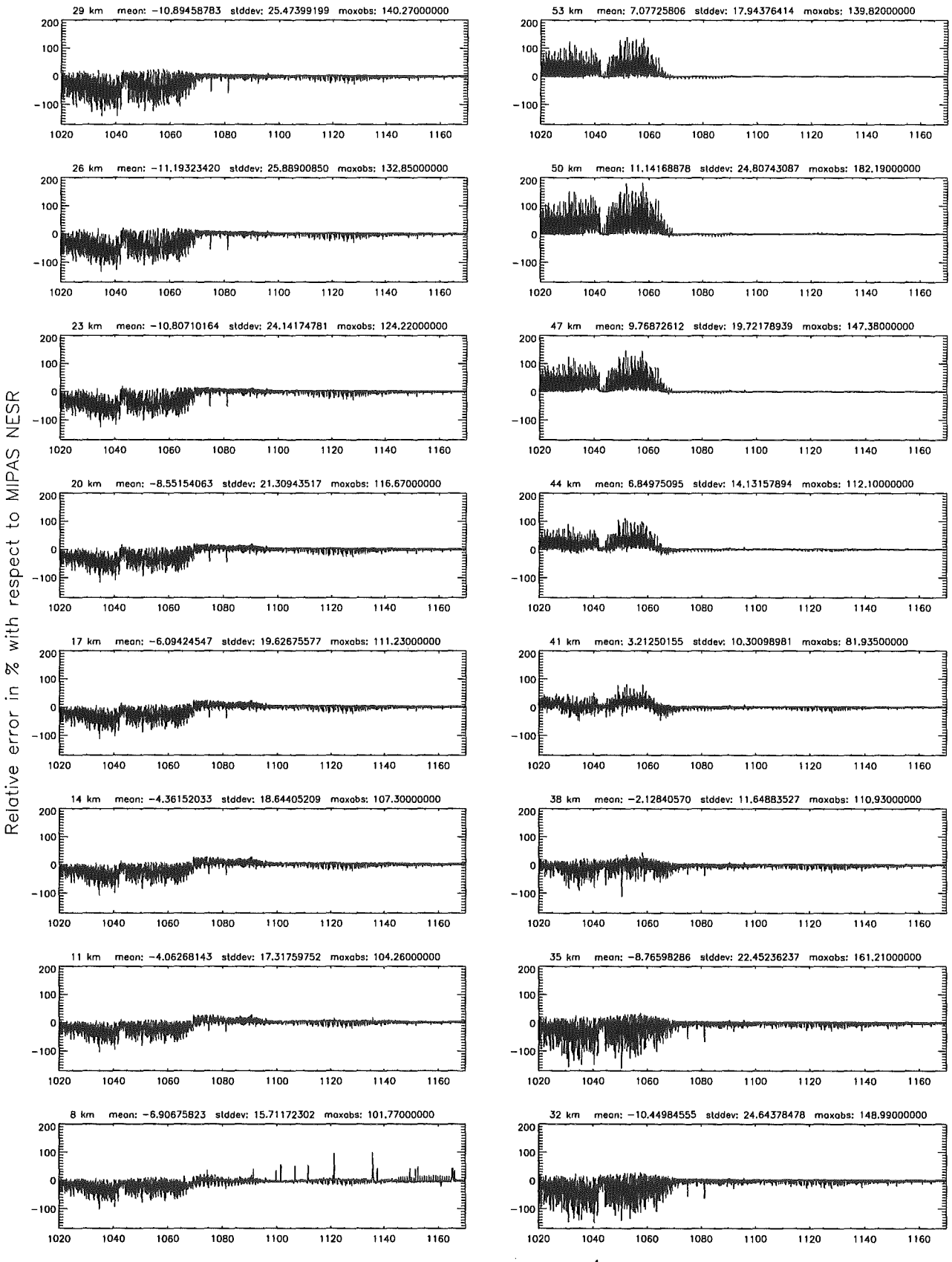


Figure 108: Relative error [%] with respect to MIPAS NESR

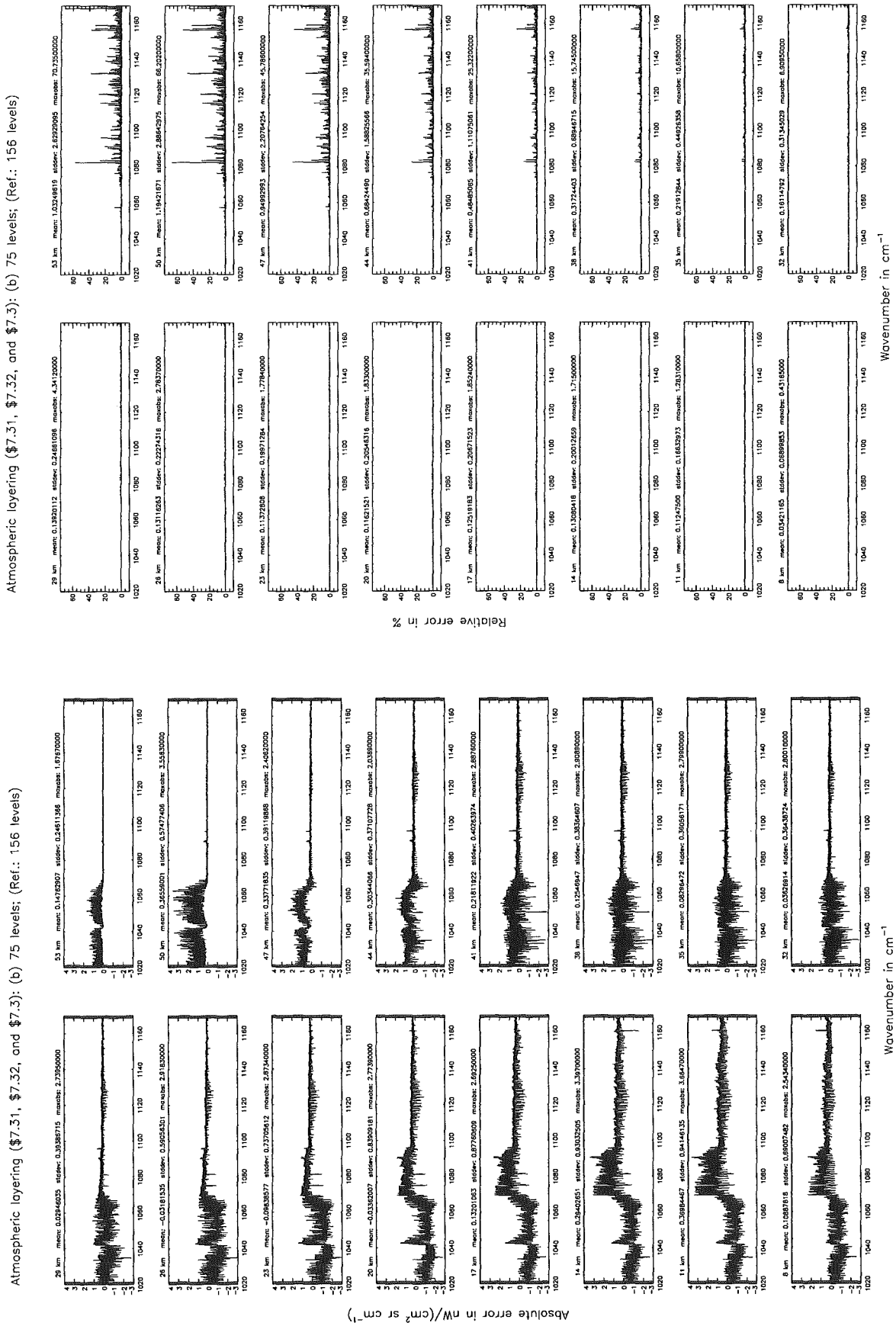


Figure 109: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

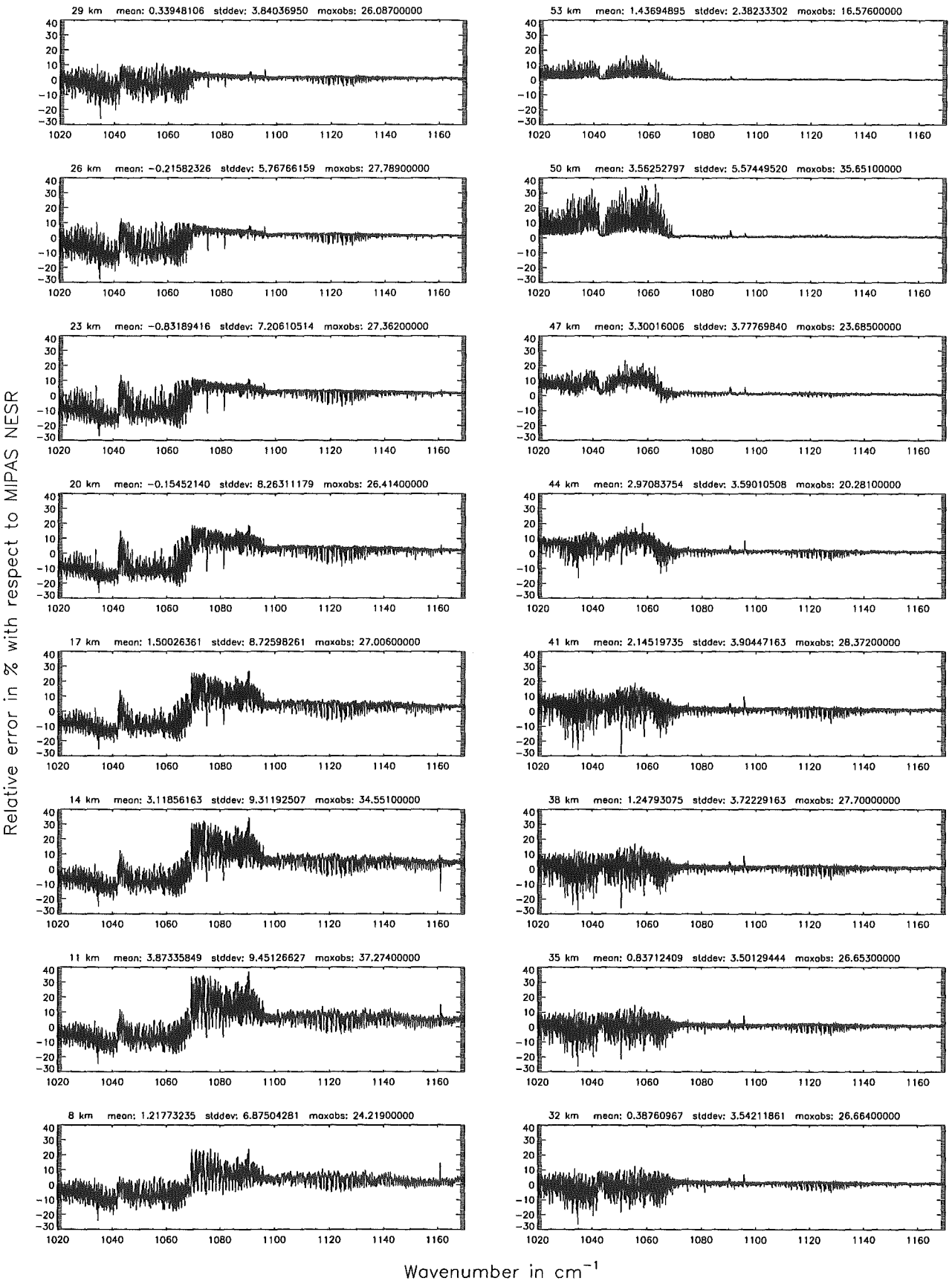


Figure 110: Relative error [%] with respect to MIPAS NESR

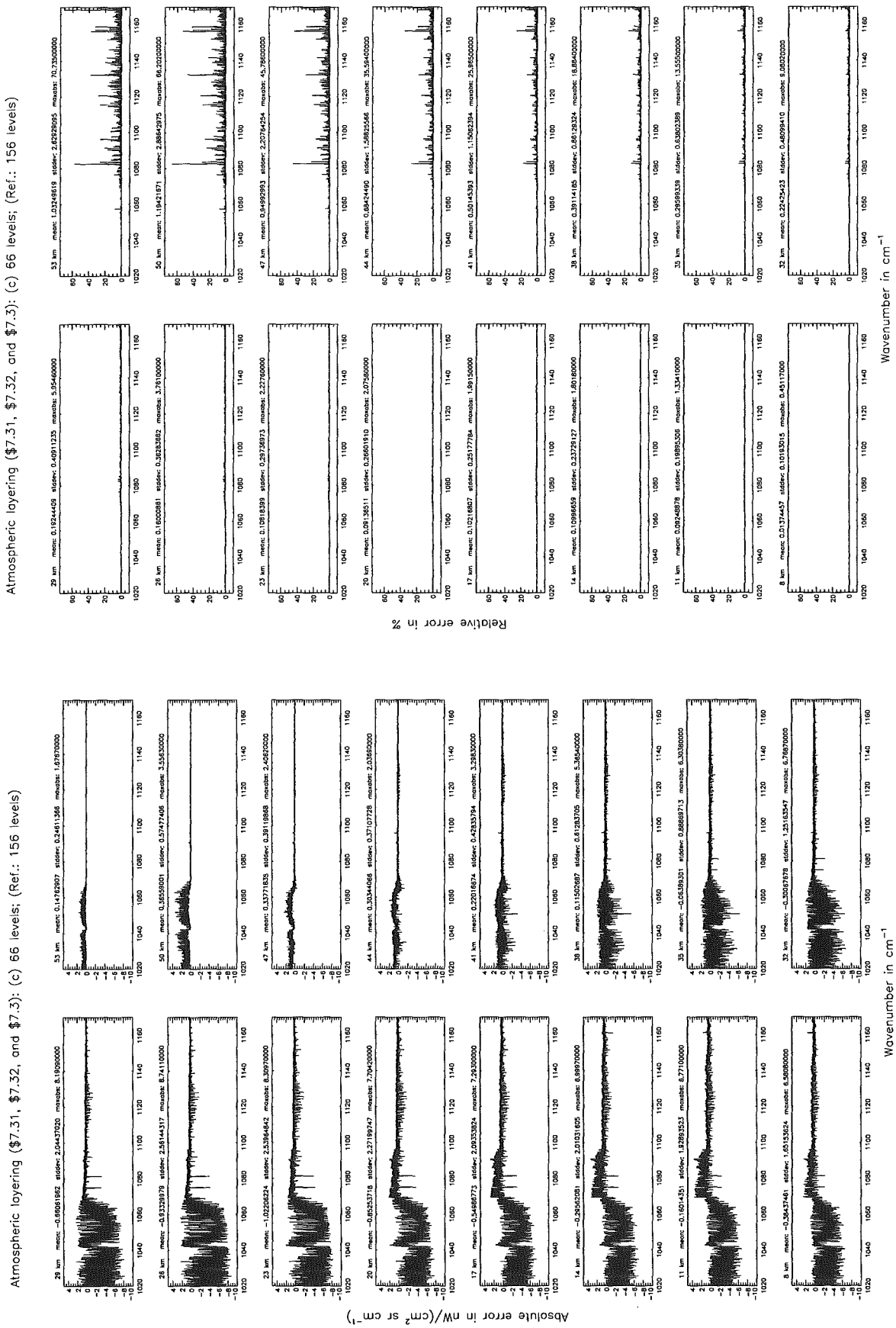


Figure 111: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (c) 66 levels; (Ref.: 156 levels)

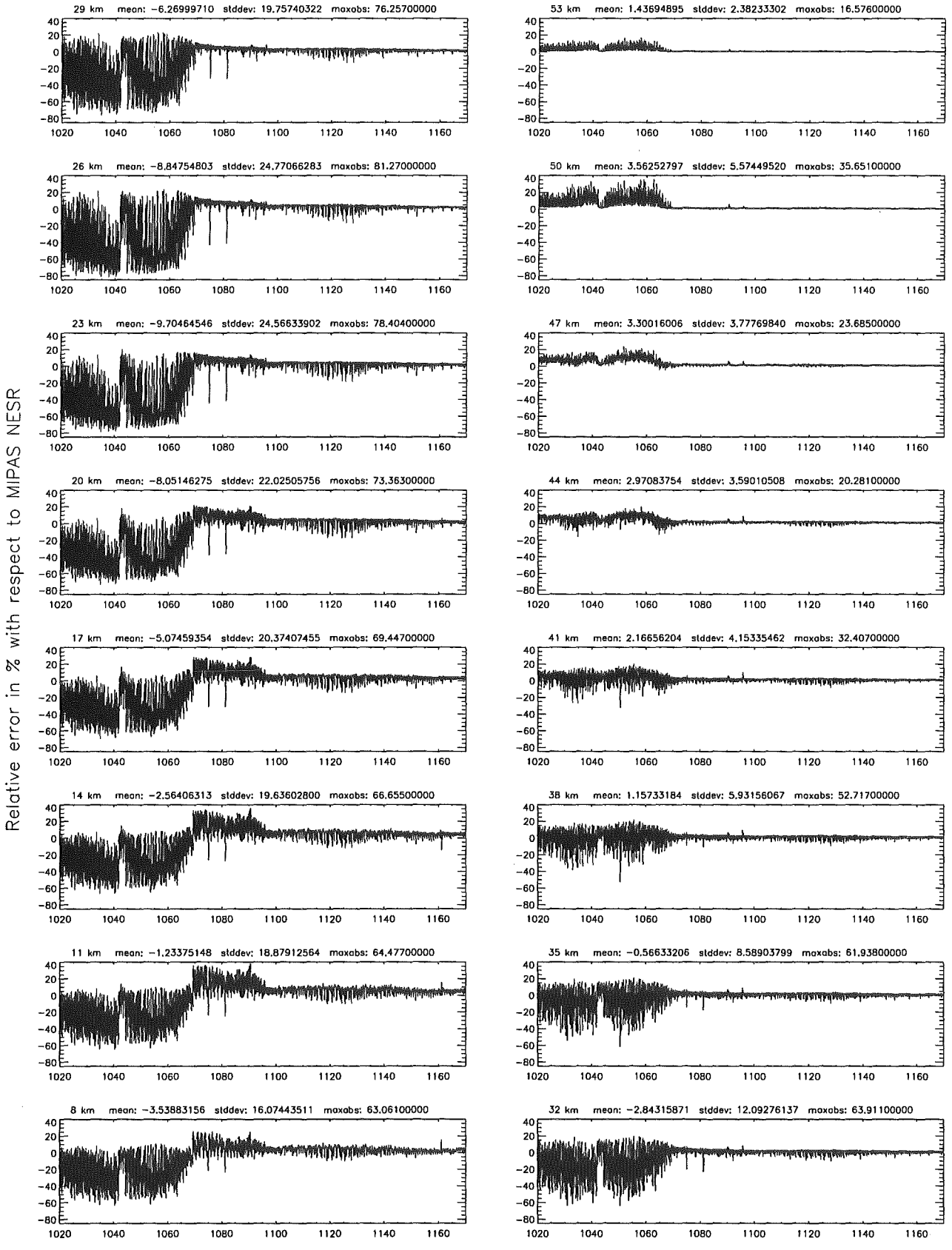
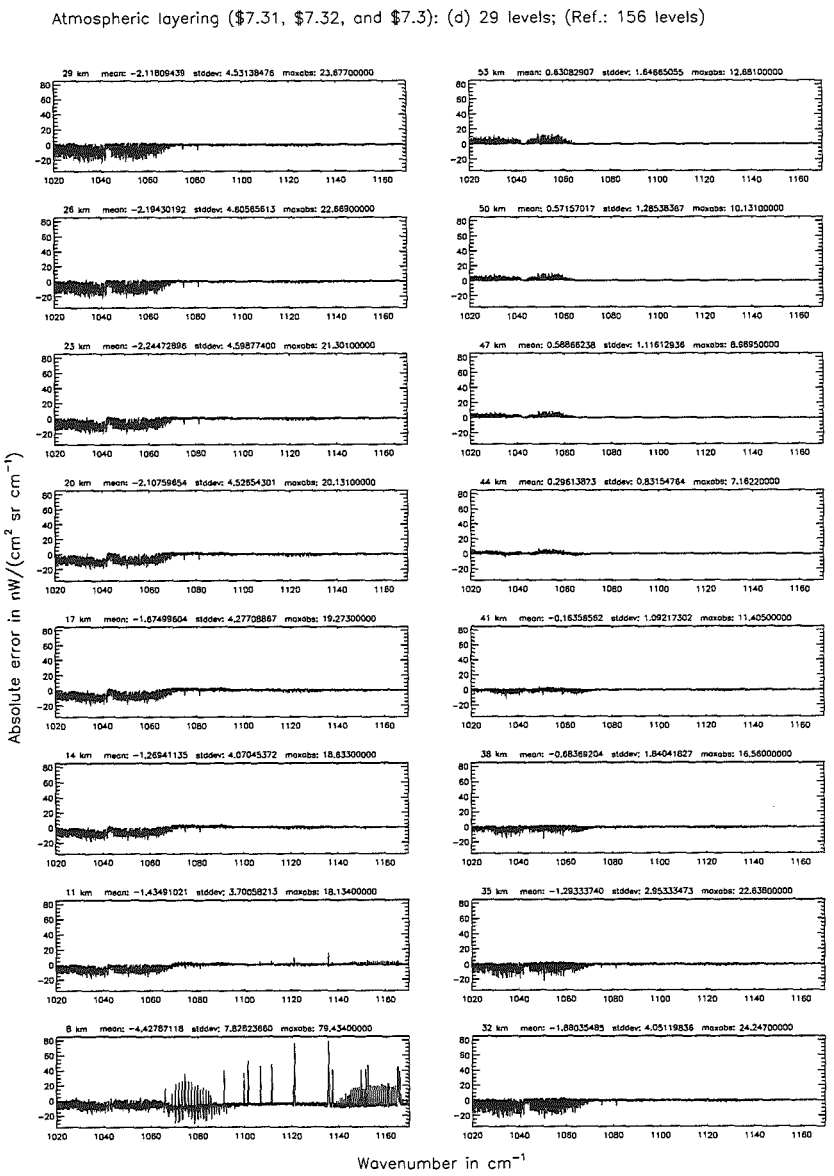


Figure 112: Relative error [%] with respect to MIPAS NESR



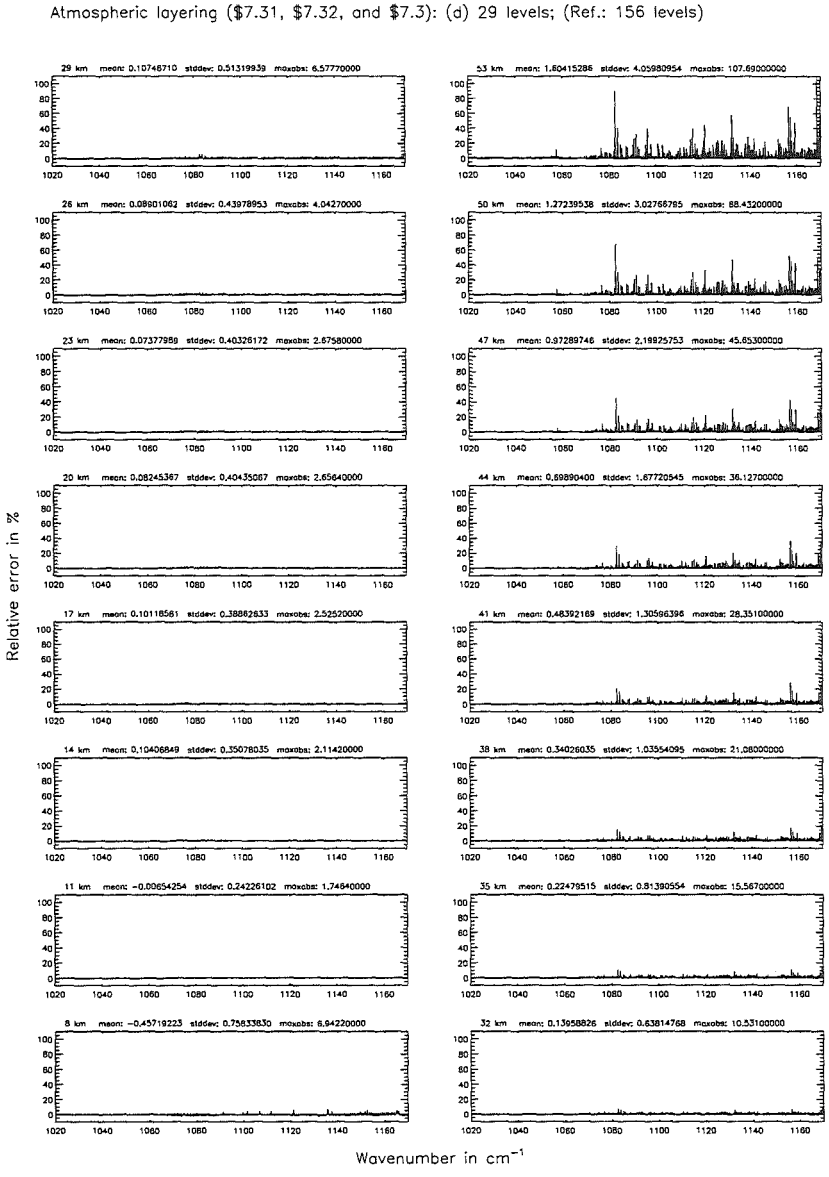


Figure 113: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (d) 29 levels; (Ref.: 156 levels)

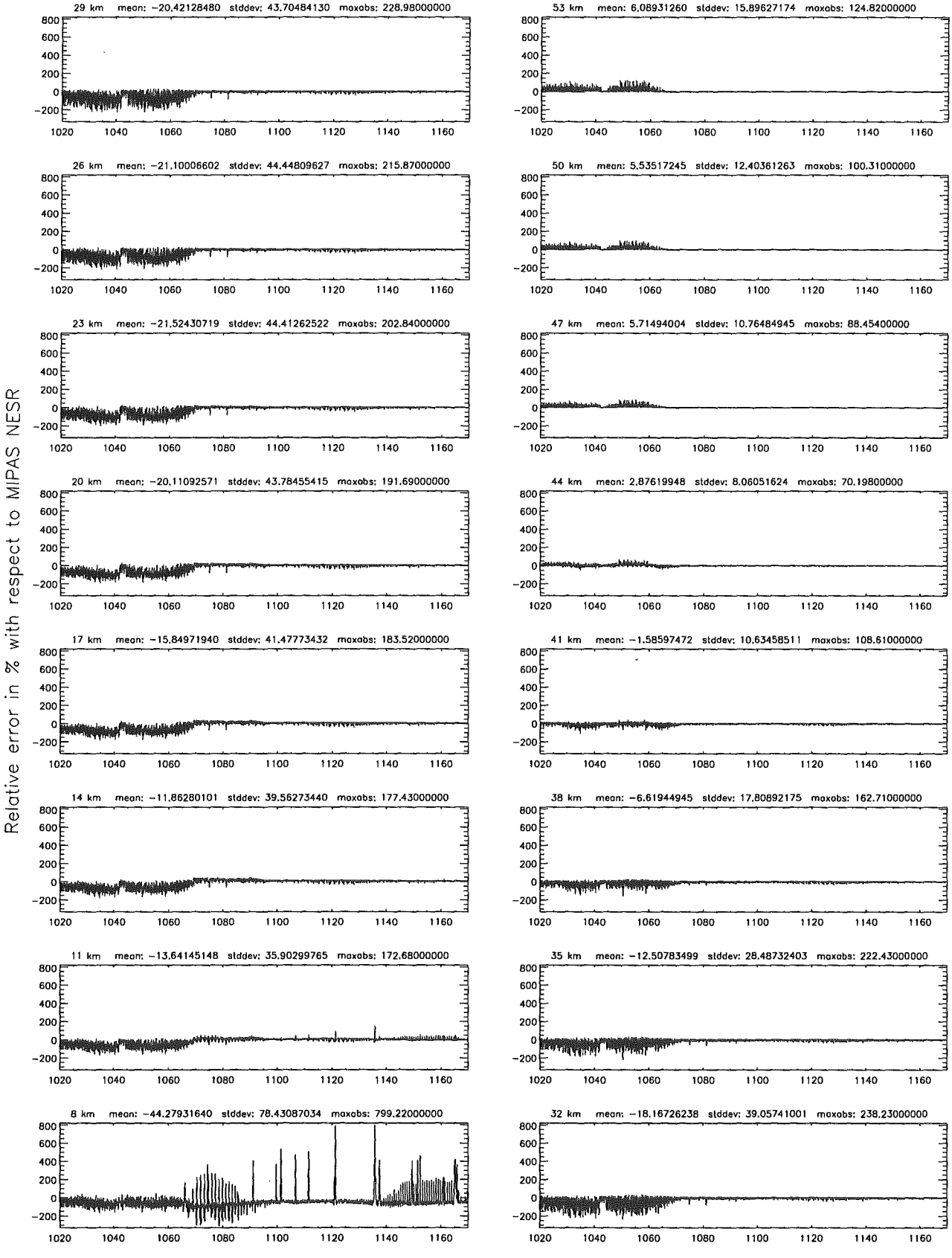
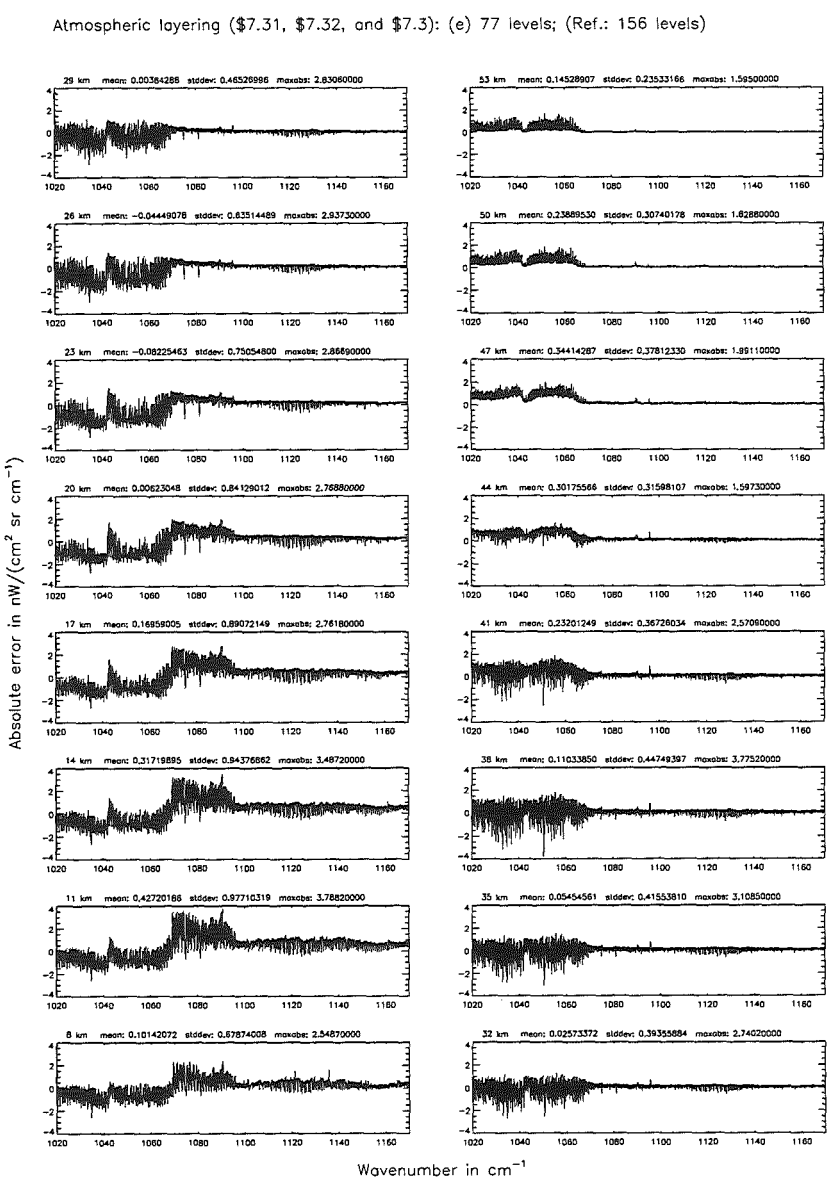


Figure 114: Relative error [%] with respect to MIPAS NESR



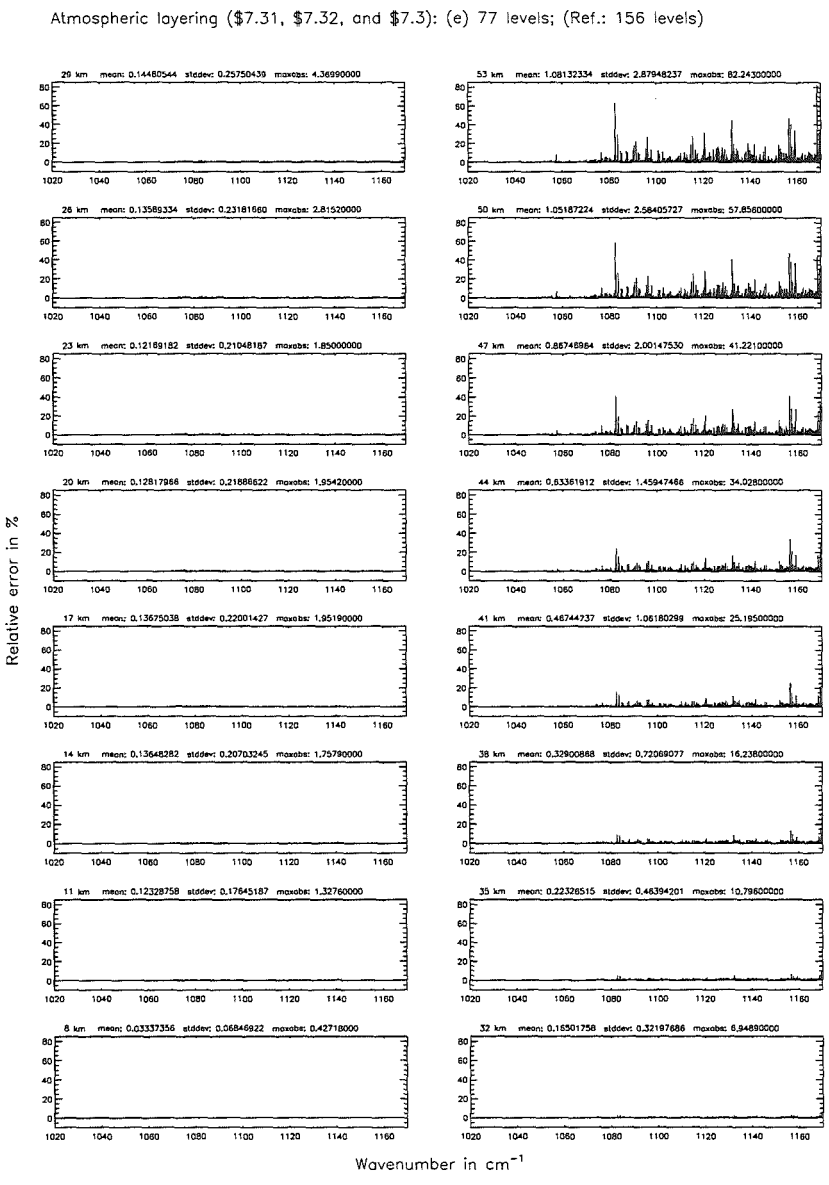


Figure 115: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (e) 77 levels; (Ref.: 156 levels)

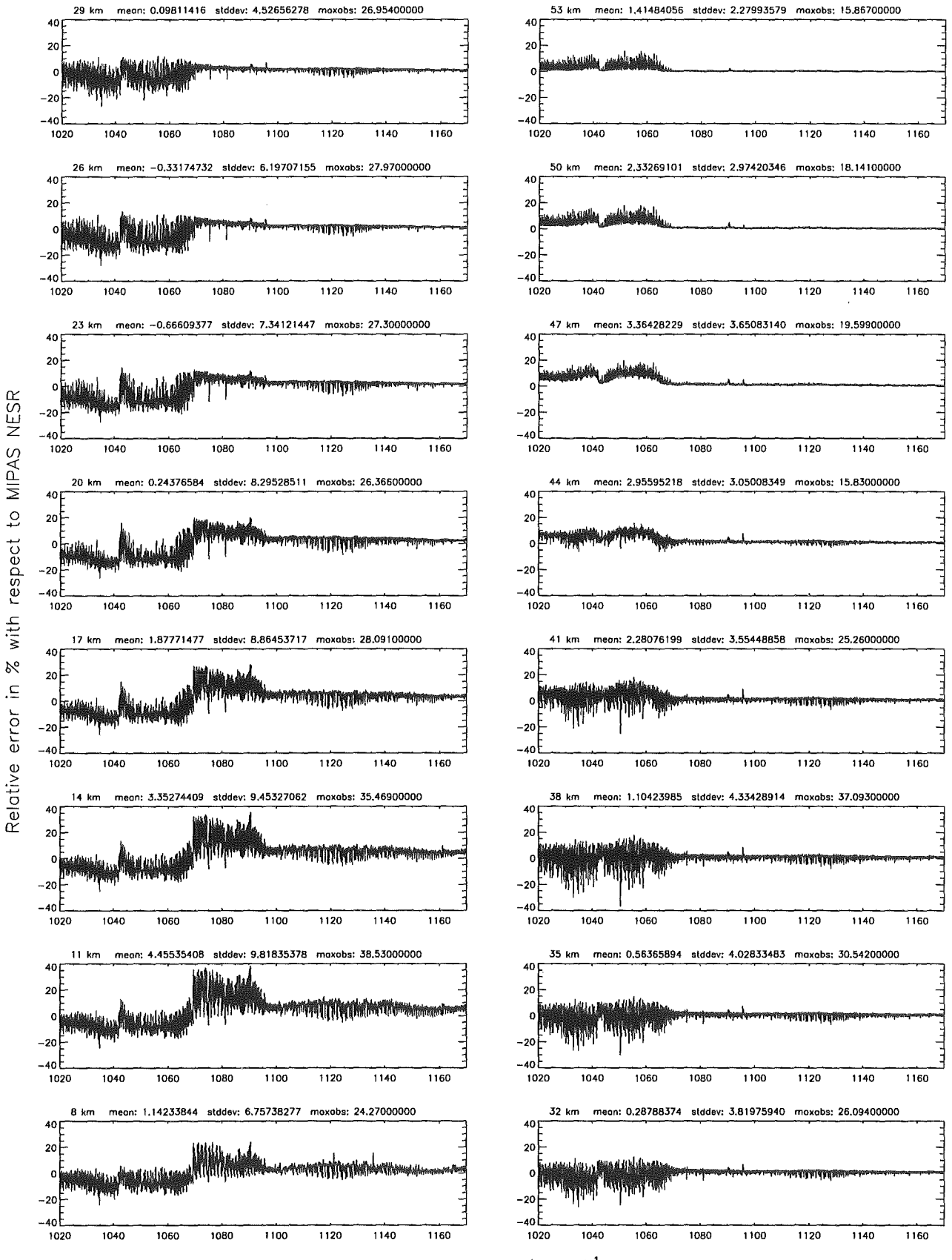


Figure 116: Relative error [%] with respect to MIPAS NESR

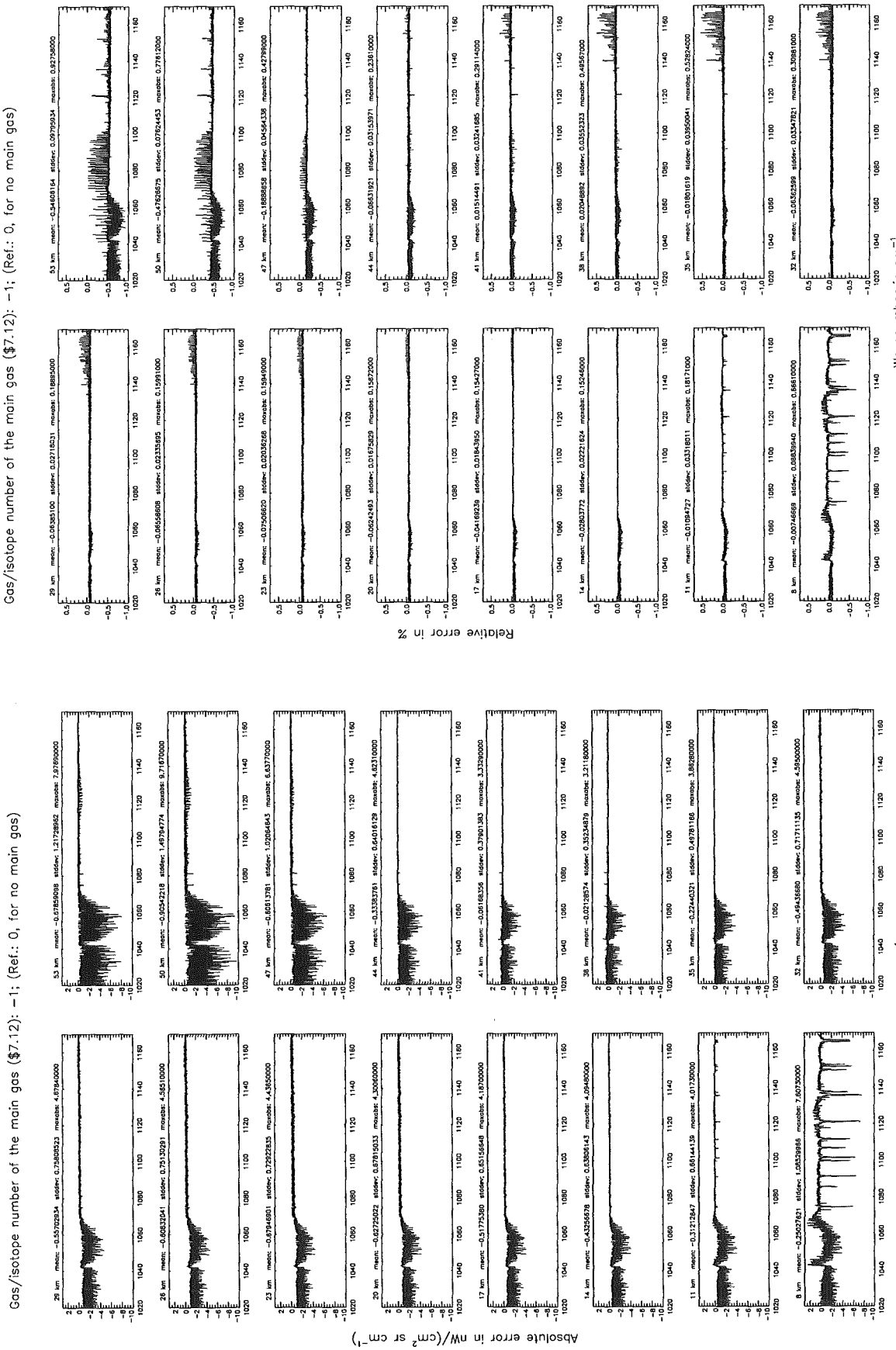


Figure 117: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Gas/isotope number of the main gas ($7.12): −1; (Ref.: 0, for no main gas)

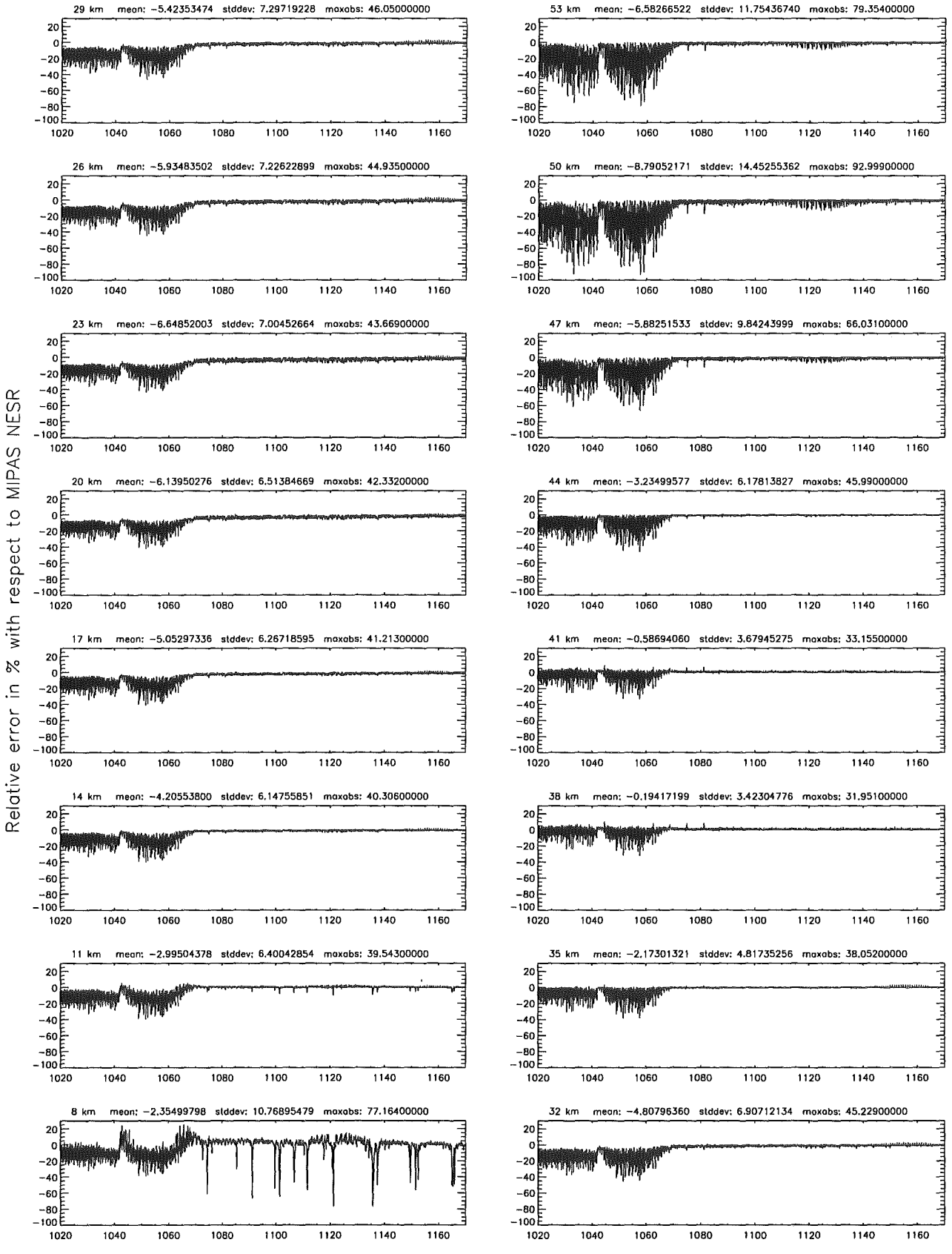


Figure 118: Relative error [%] with respect to MIPAS NESR

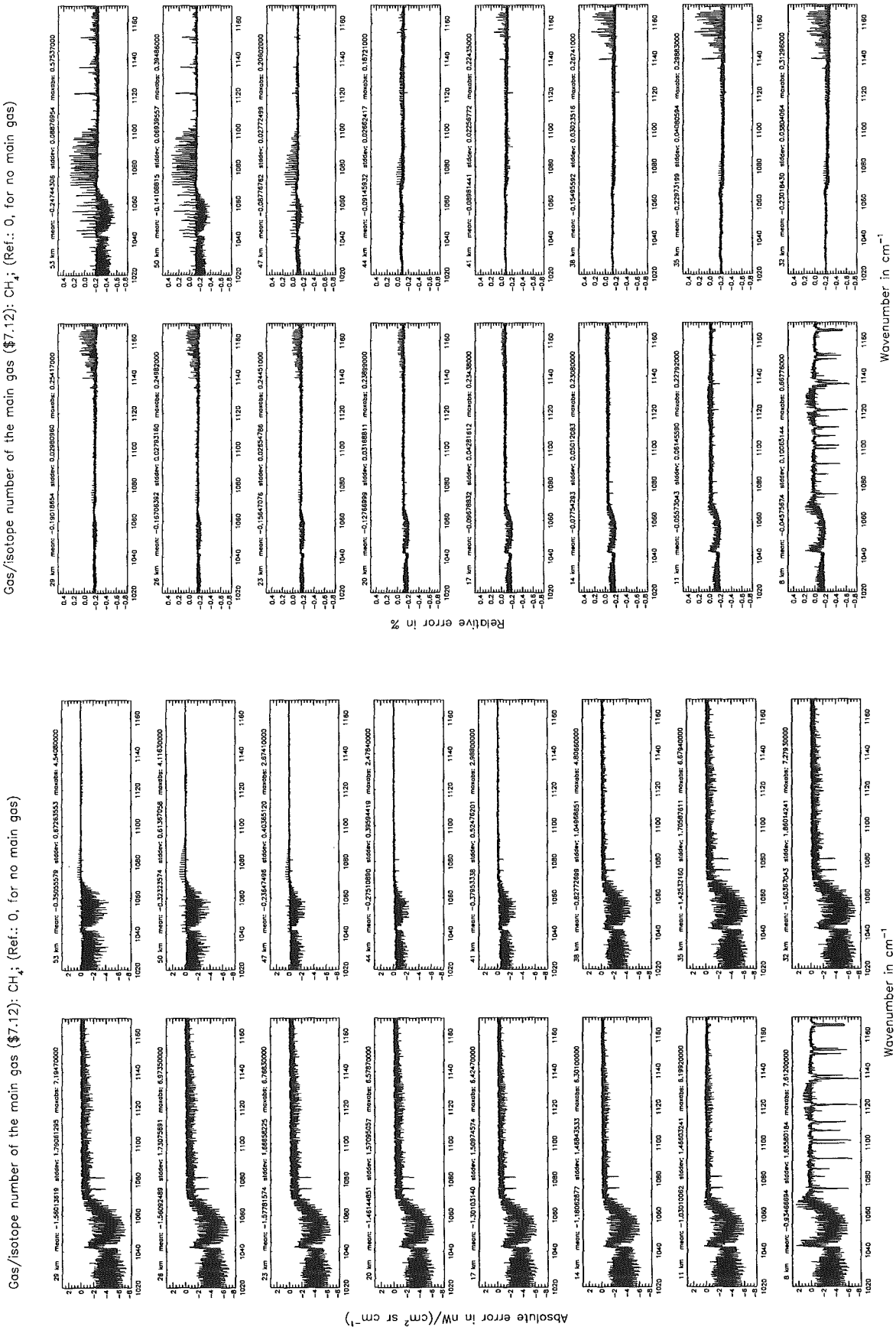


Figure 119: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

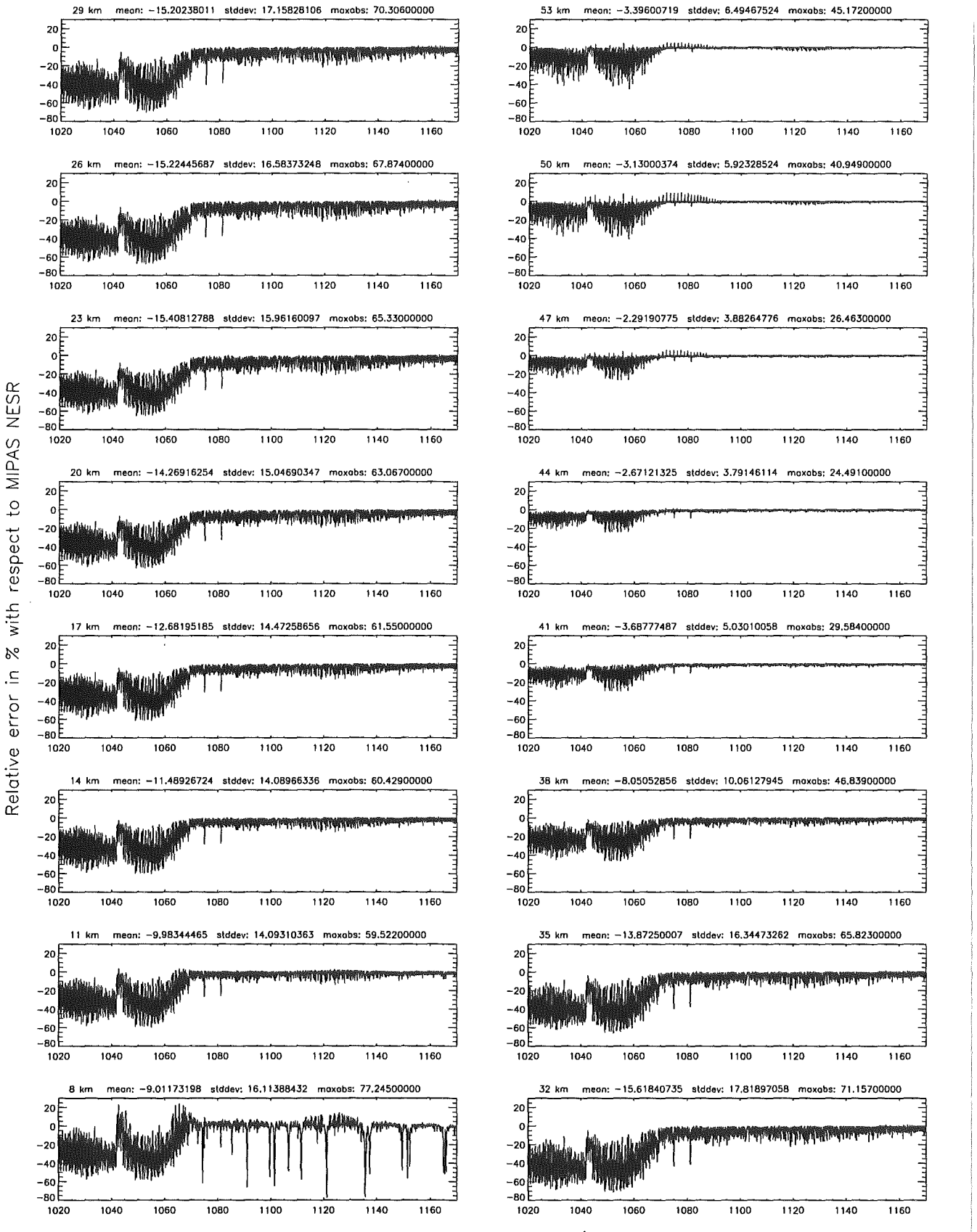


Figure 120: Relative error [%] with respect to MIPAS NESR

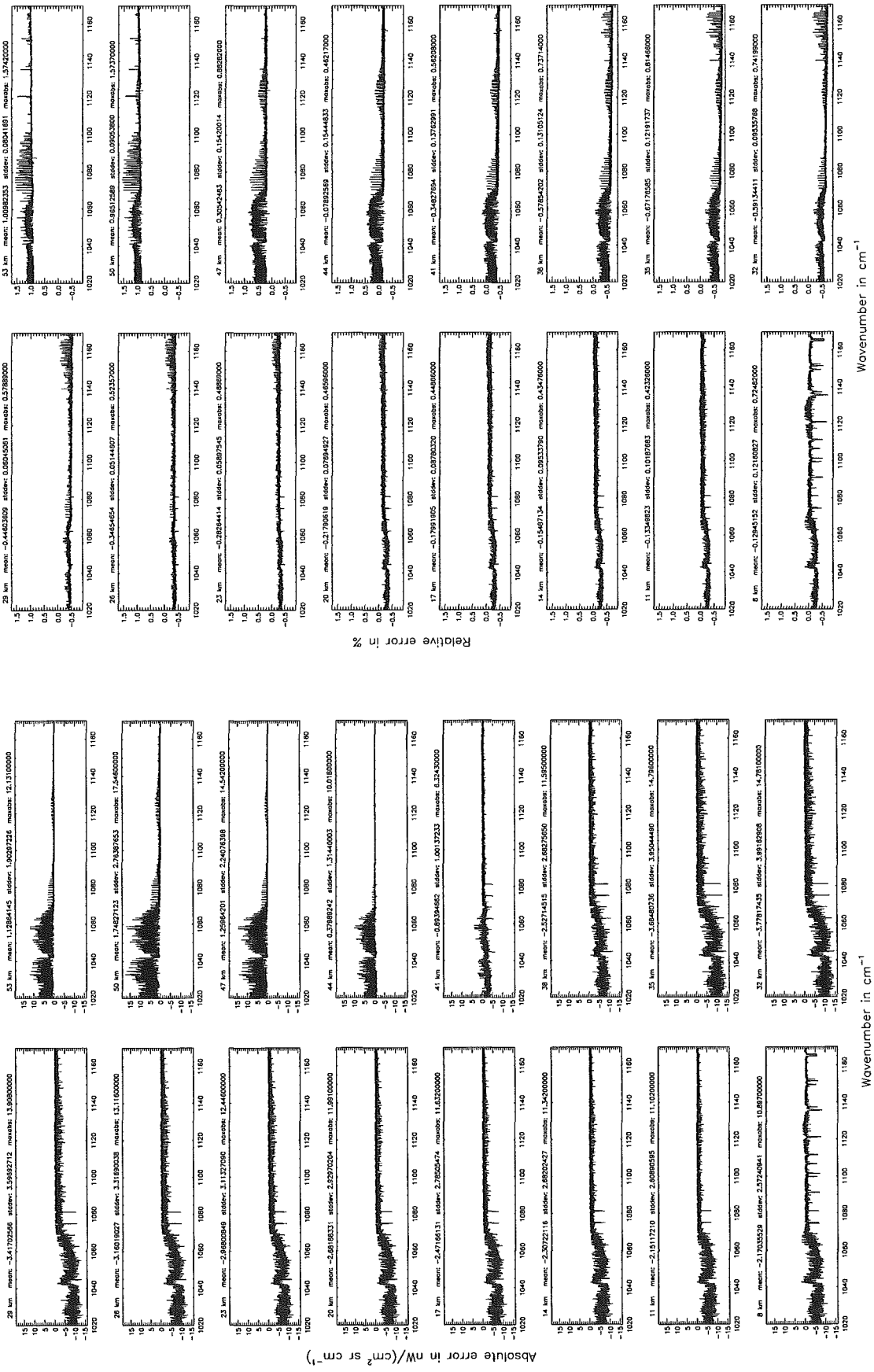


Figure 121: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

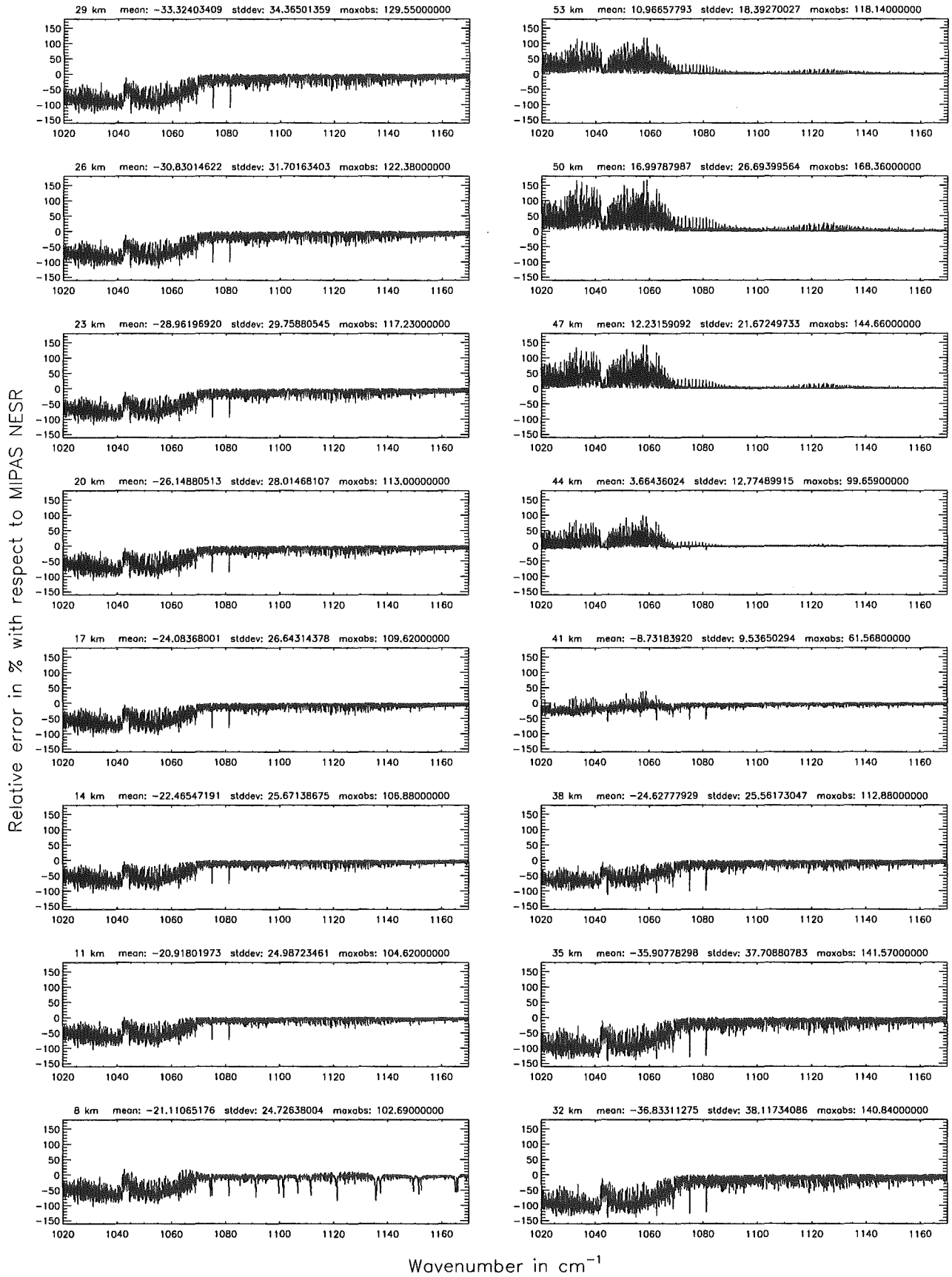


Figure 122: Relative error [%] with respect to MIPAS NESR

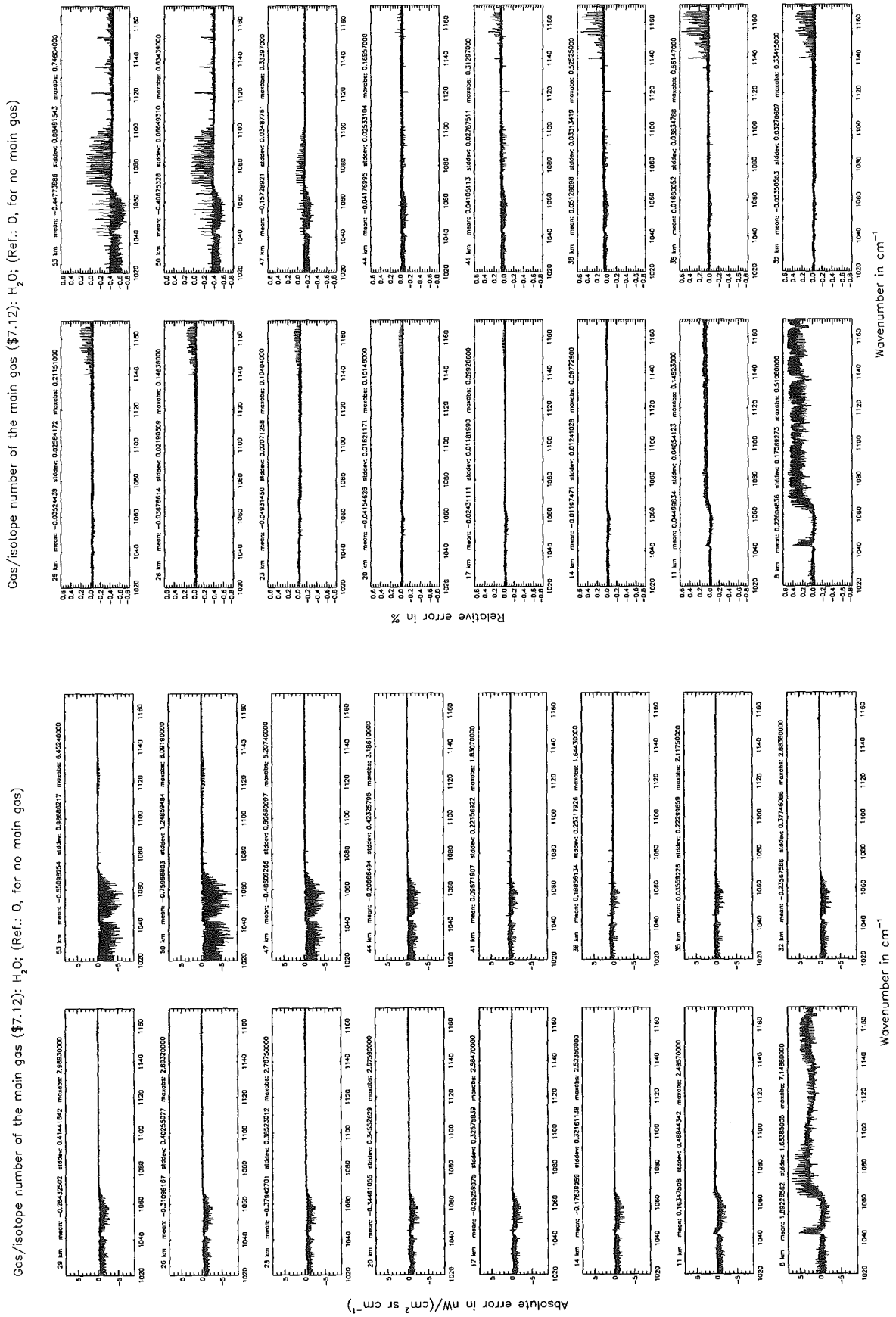


Figure 123: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

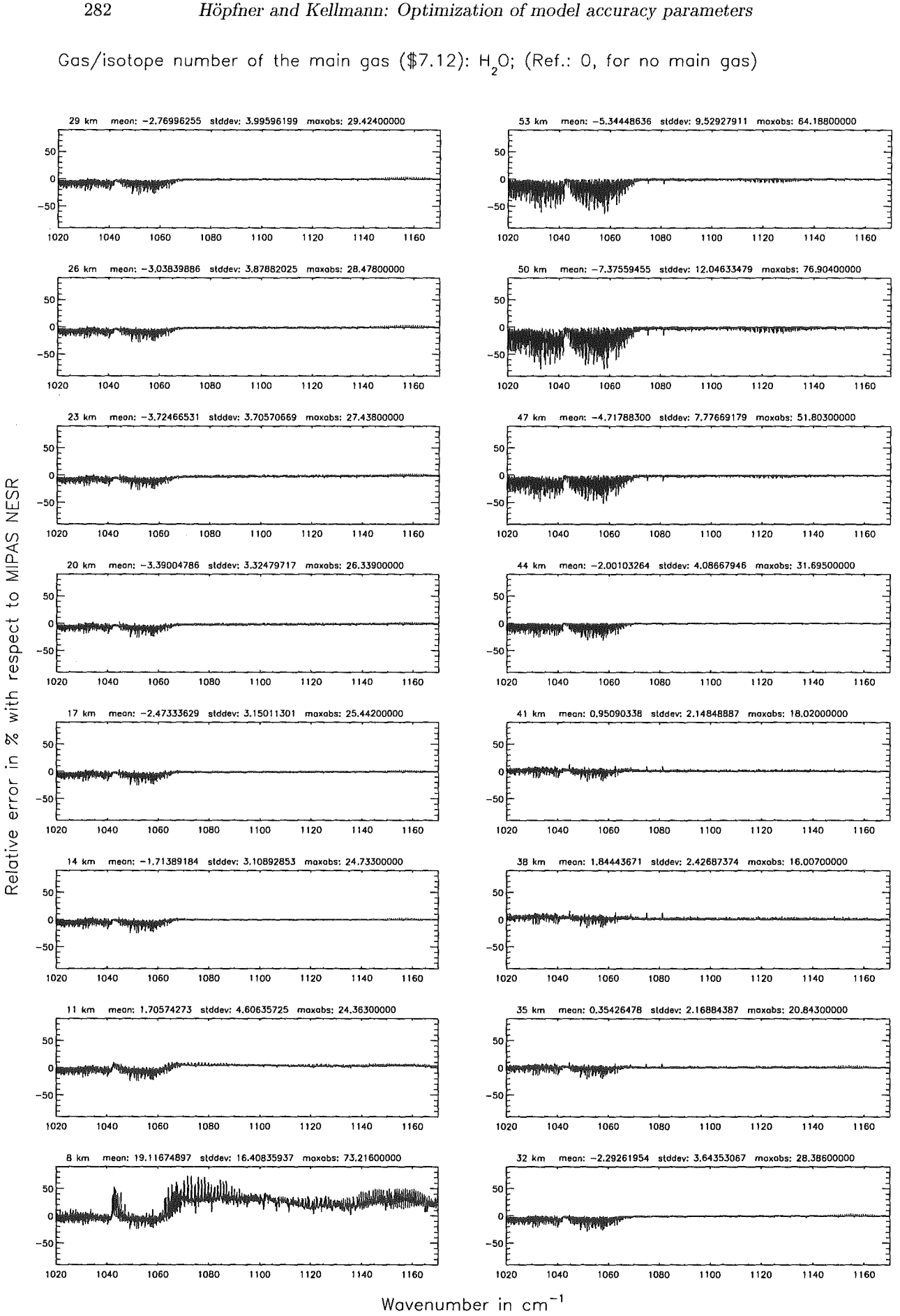


Figure 124: Relative error [%] with respect to MIPAS NESR

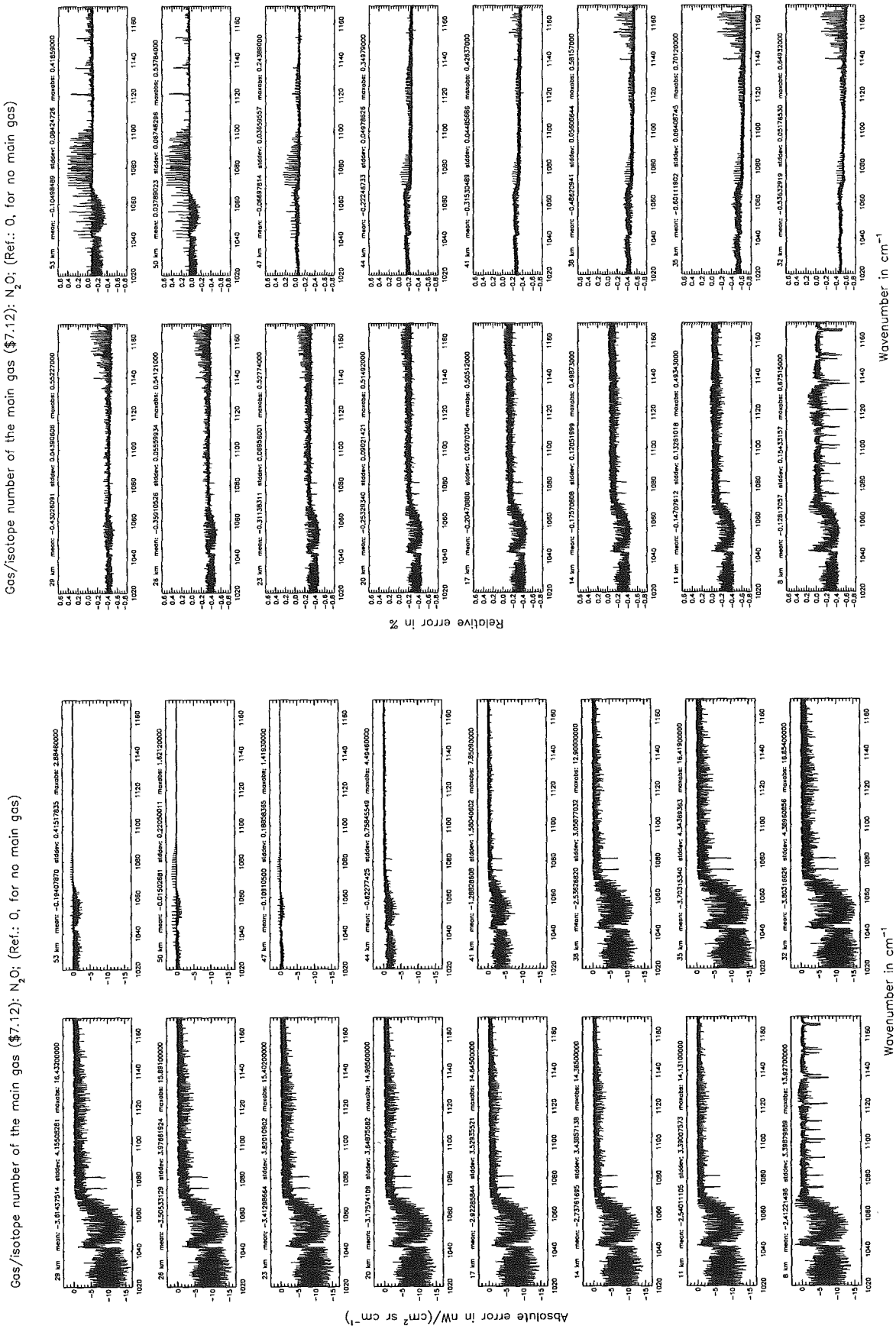


Figure 125: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

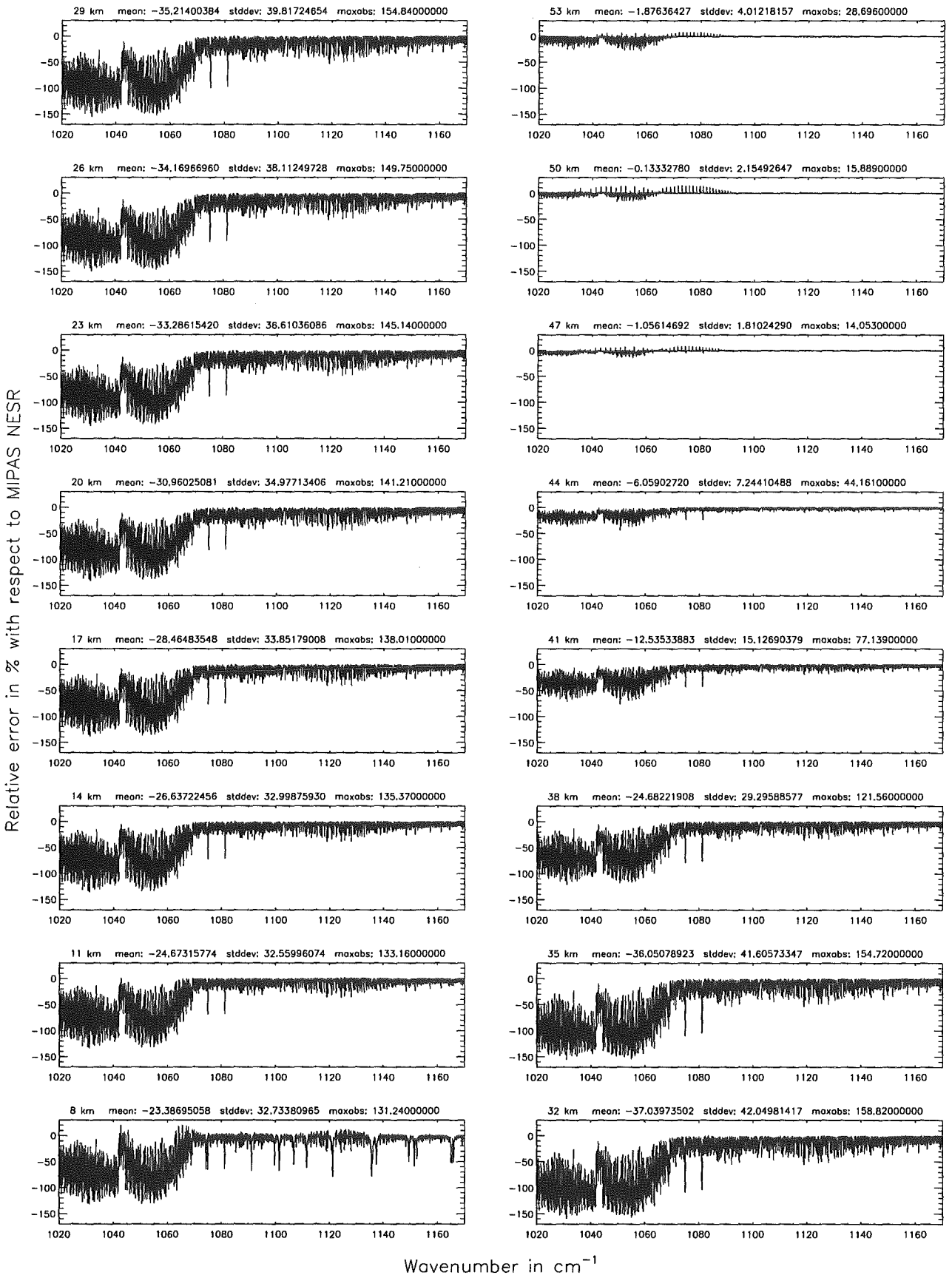


Figure 126: Relative error [%] with respect to MIPAS NESR

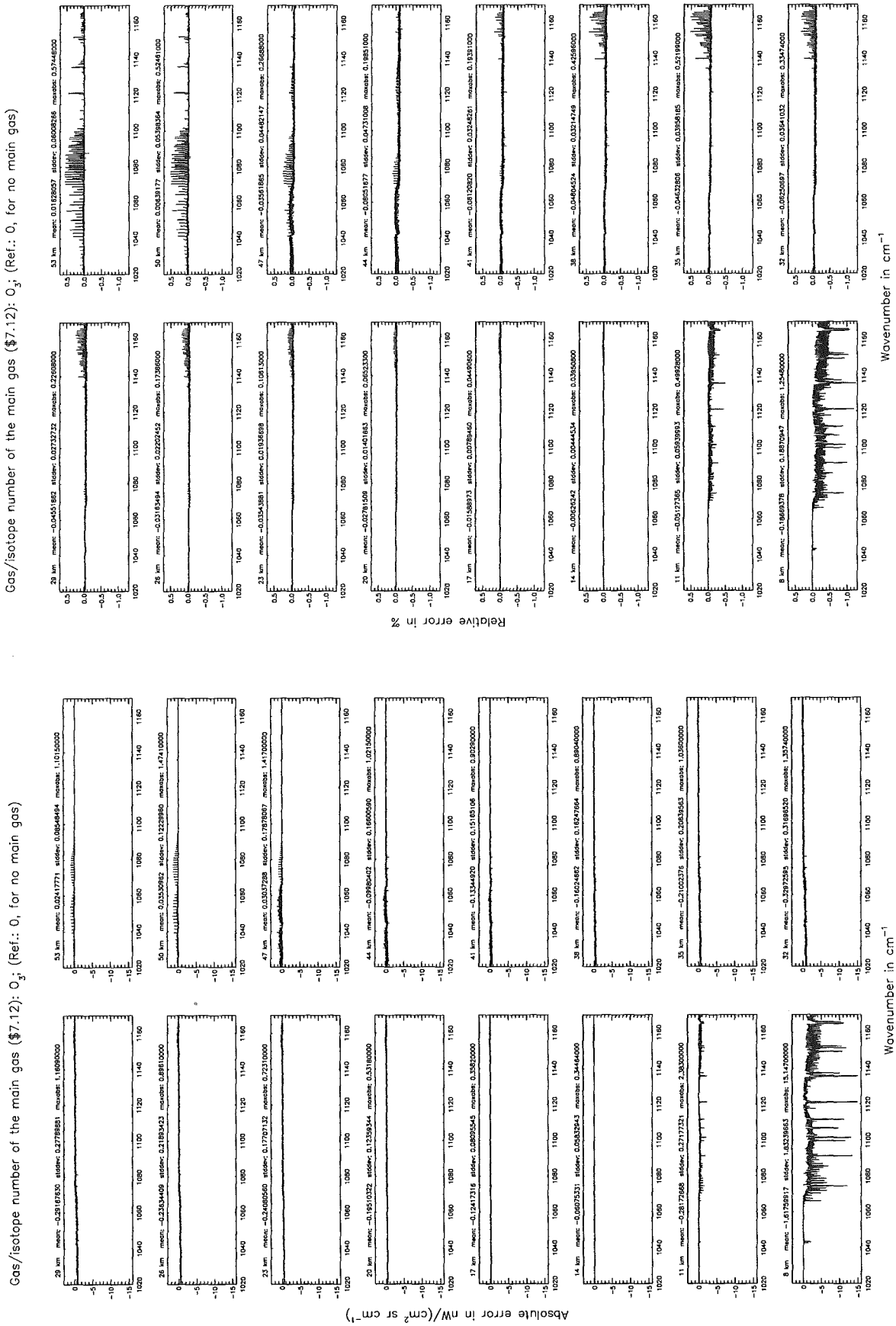


Figure 127: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

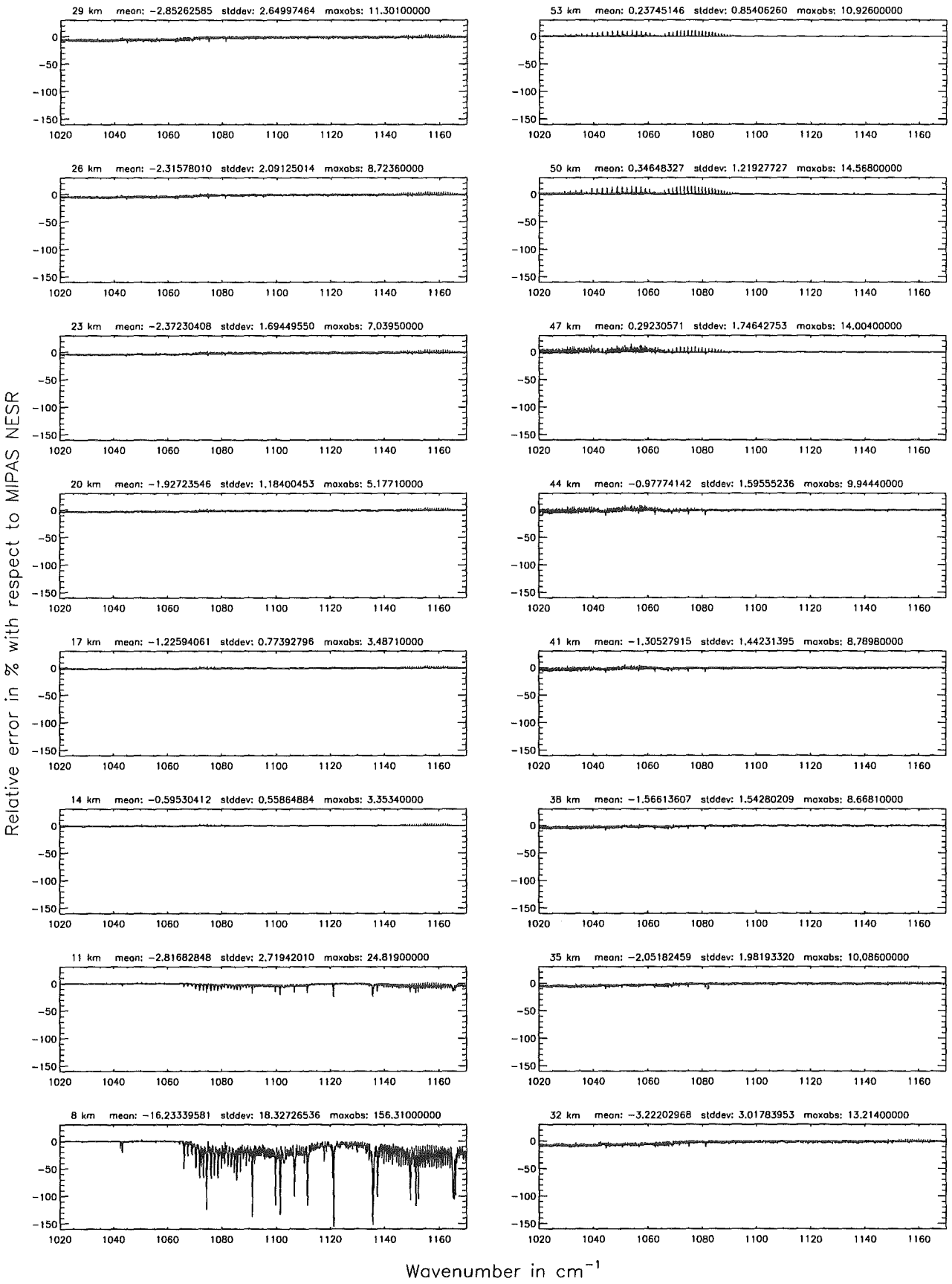


Figure 128: Relative error [%] with respect to MIPAS NESR

Band B: 1215 – 1500 cm^{-1}

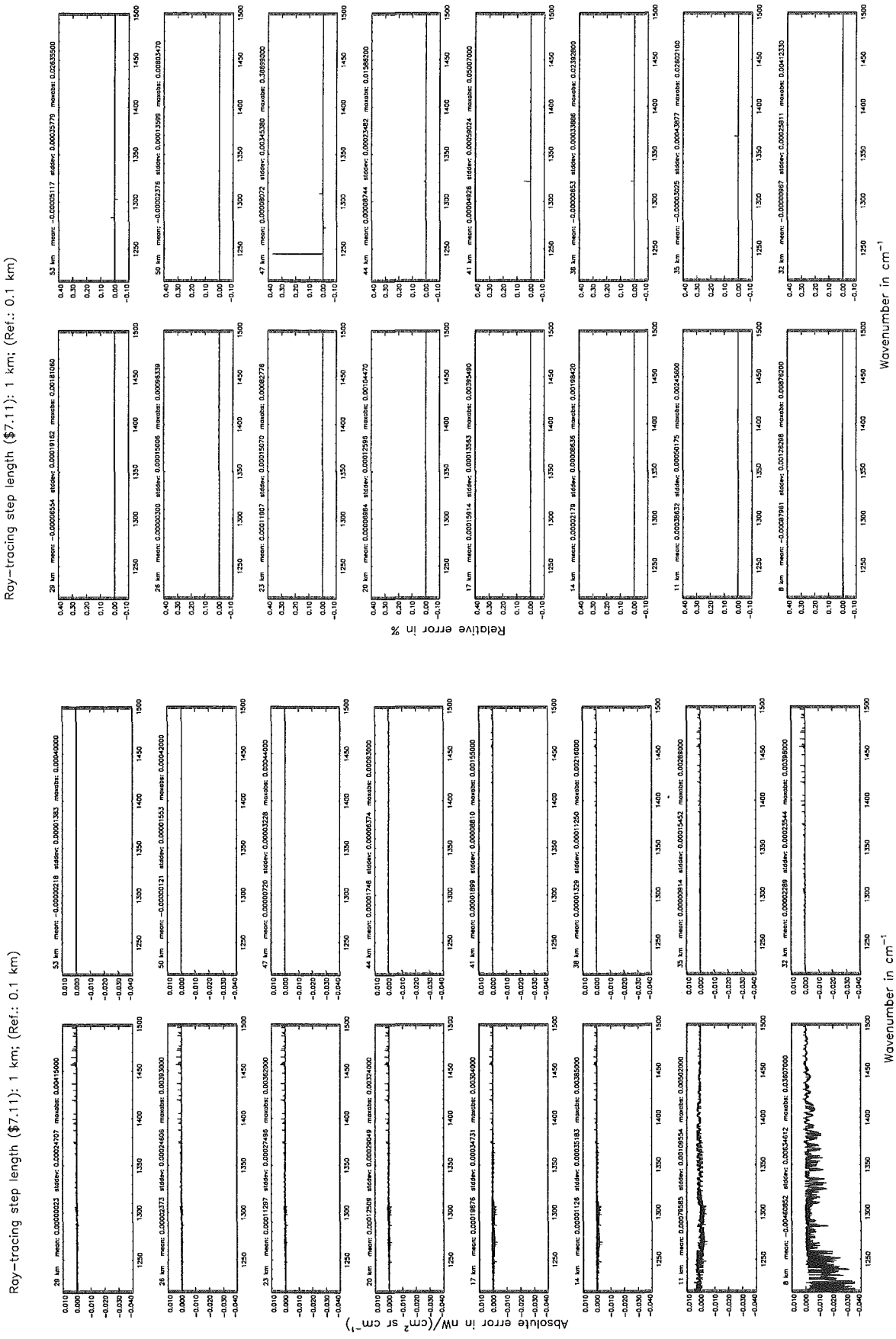


Figure 129: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

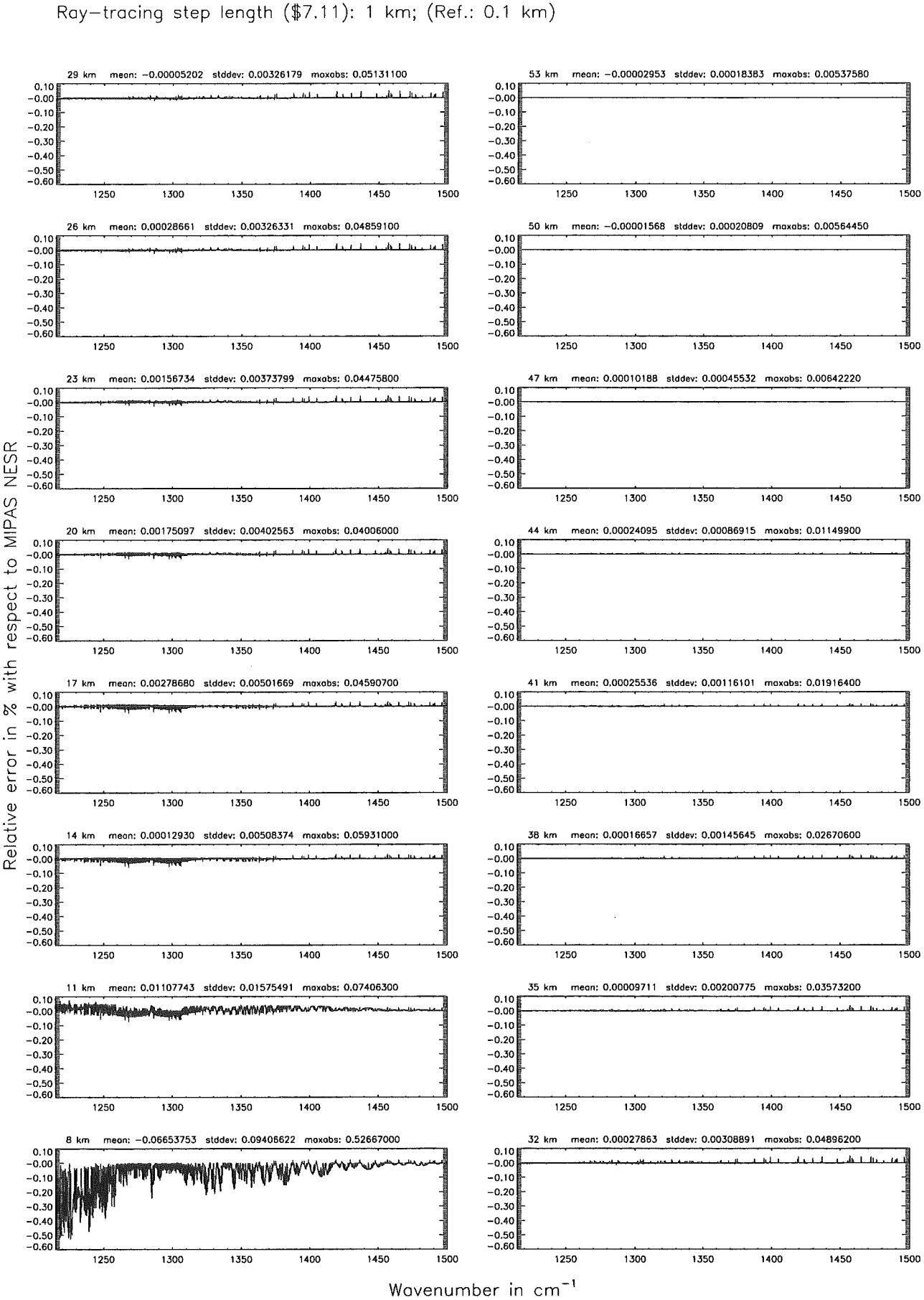
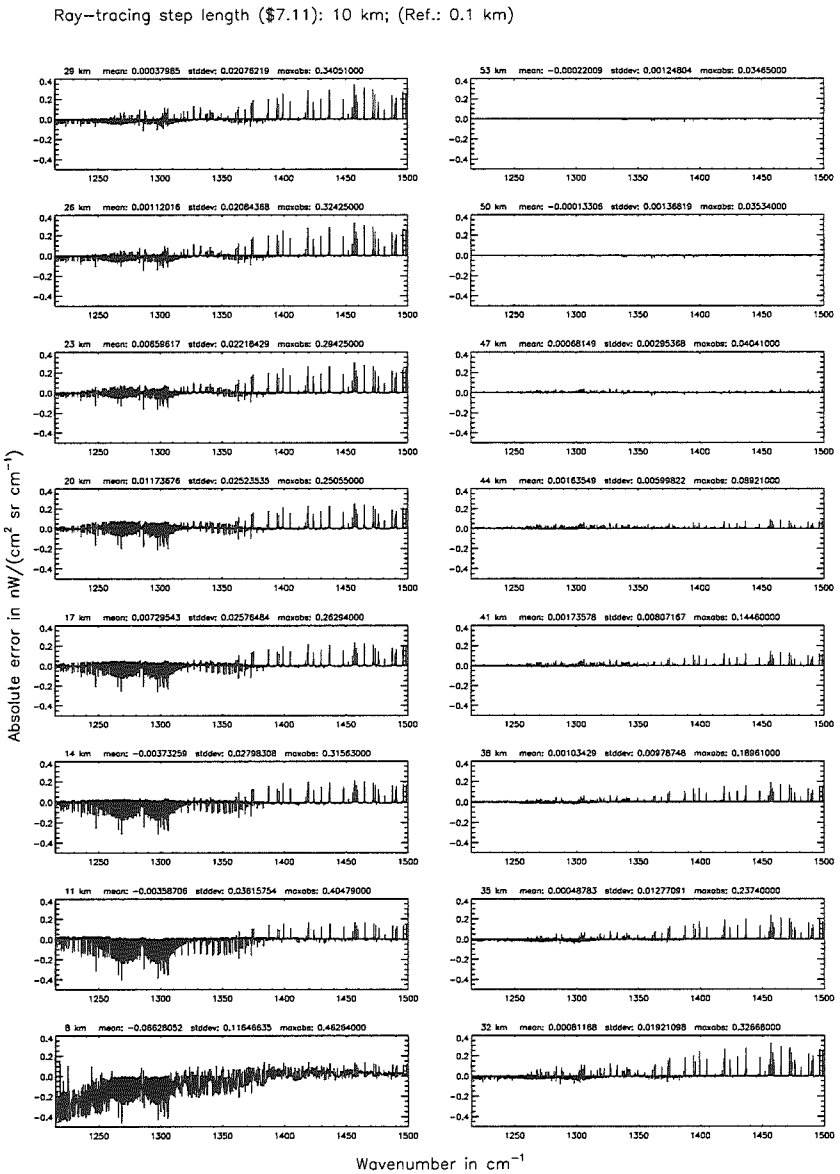


Figure 130: Relative error [%] with respect to MIPAS NESR



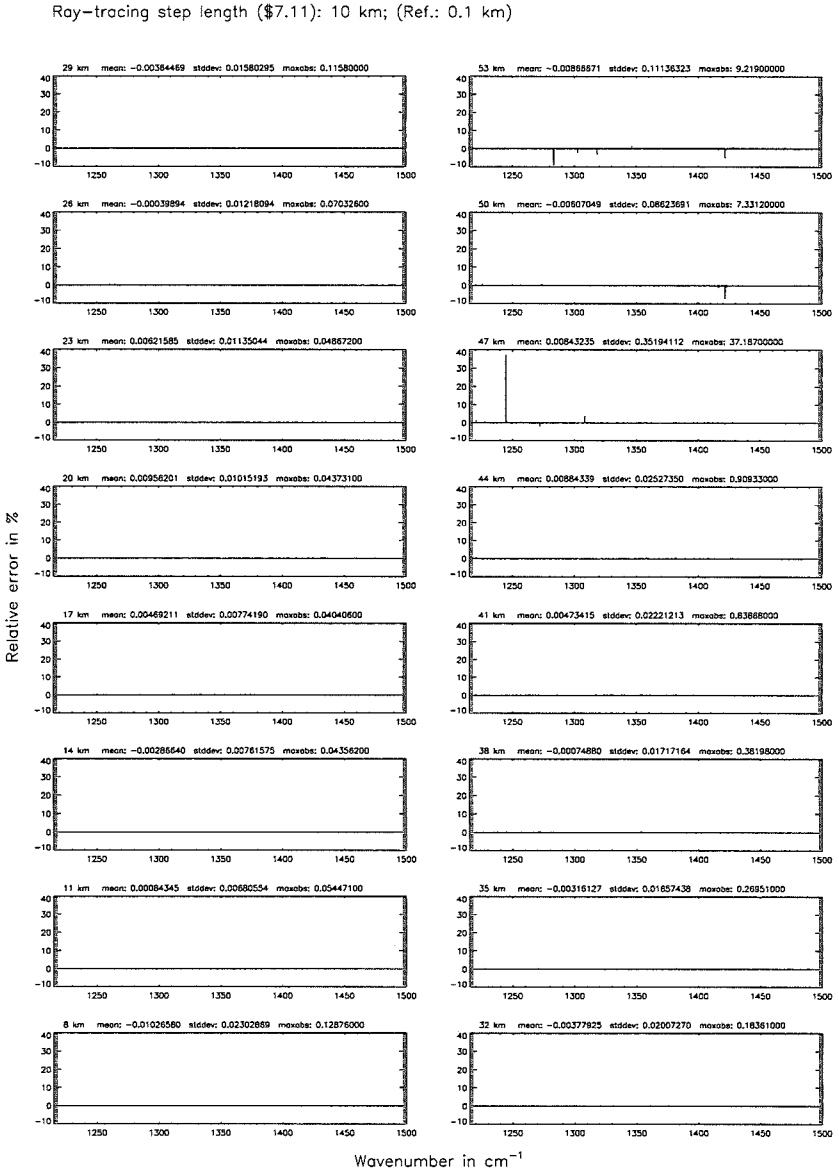


Figure 131: Absolute error [nW/(cm² sr cm⁻¹)] and relative error [%]

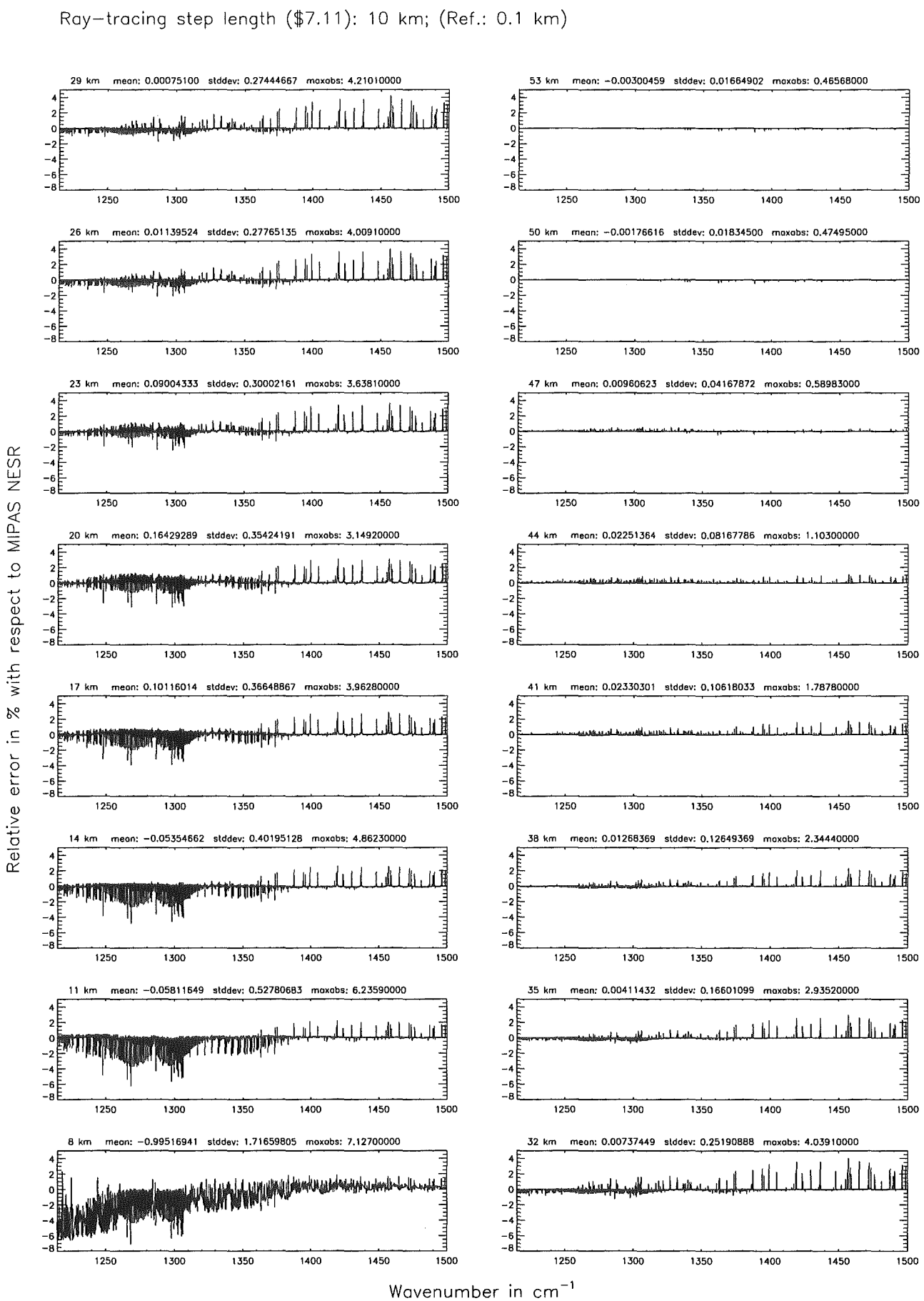
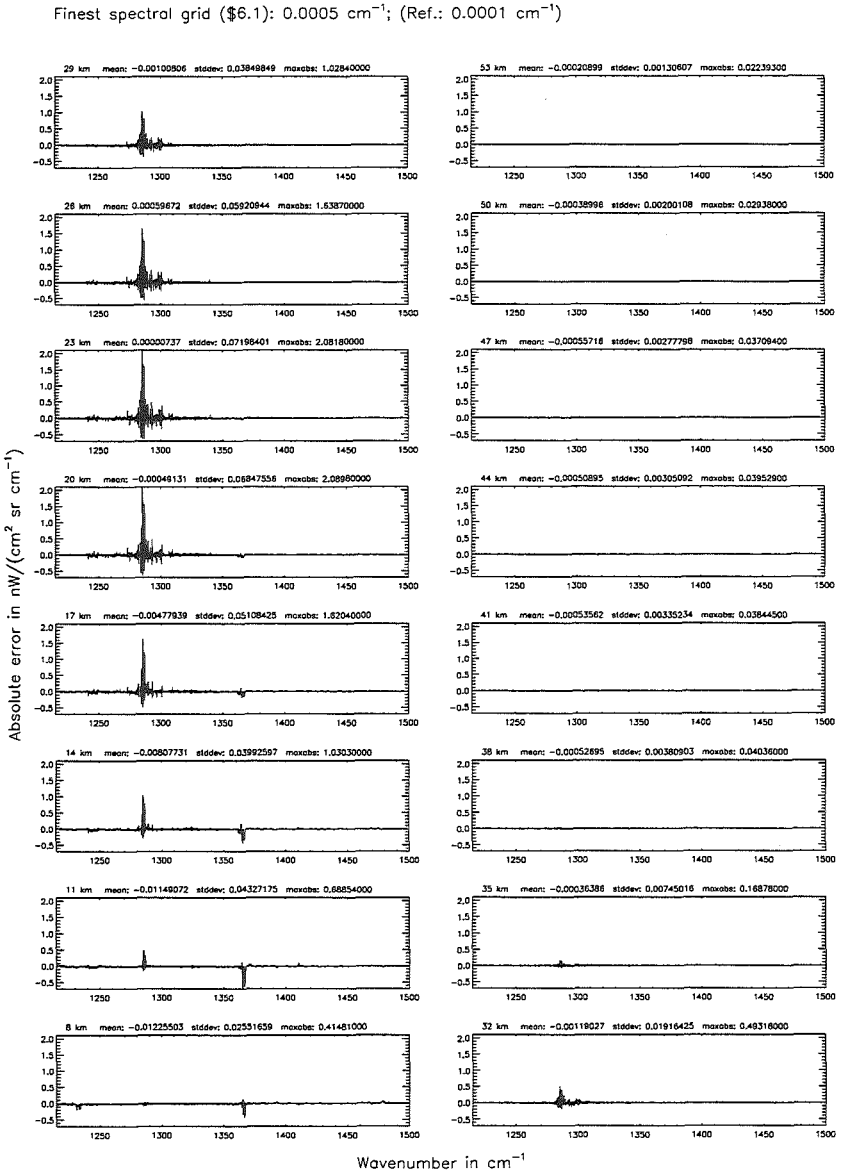


Figure 132: Relative error [%] with respect to MIPAS NESR



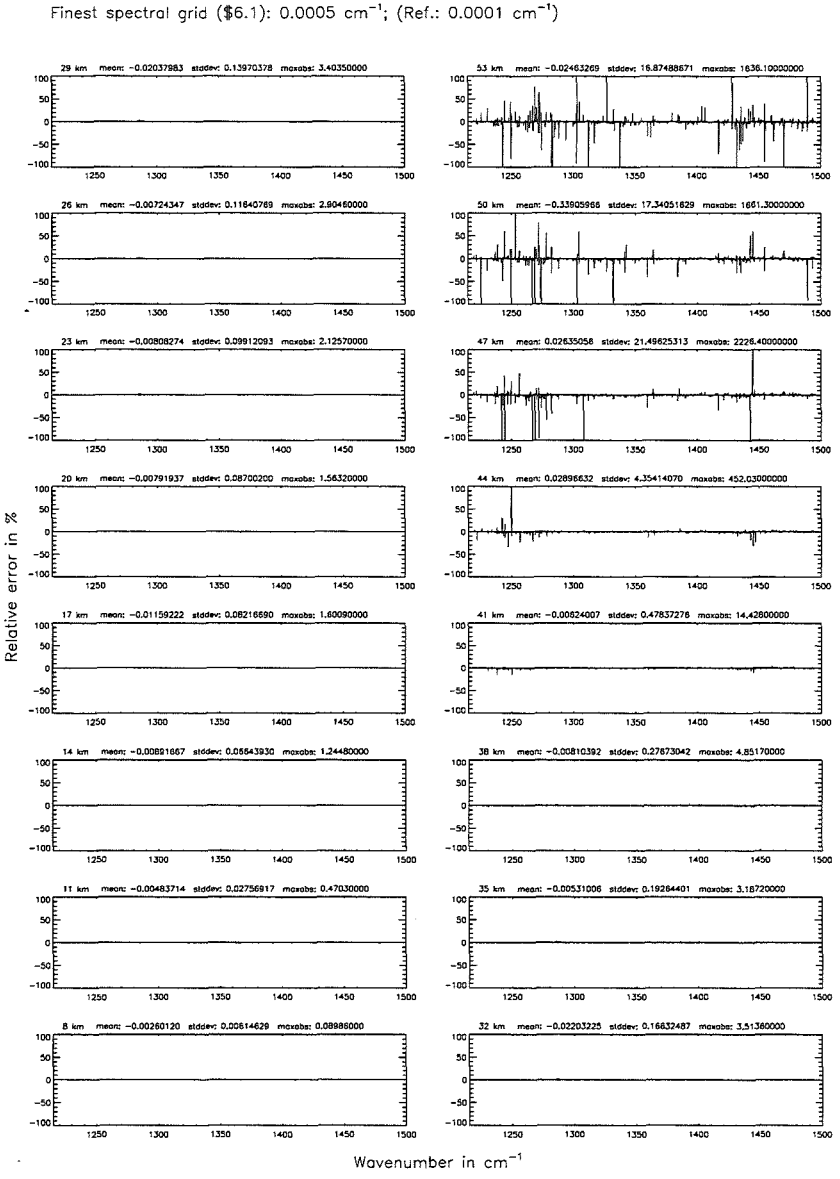


Figure 133: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

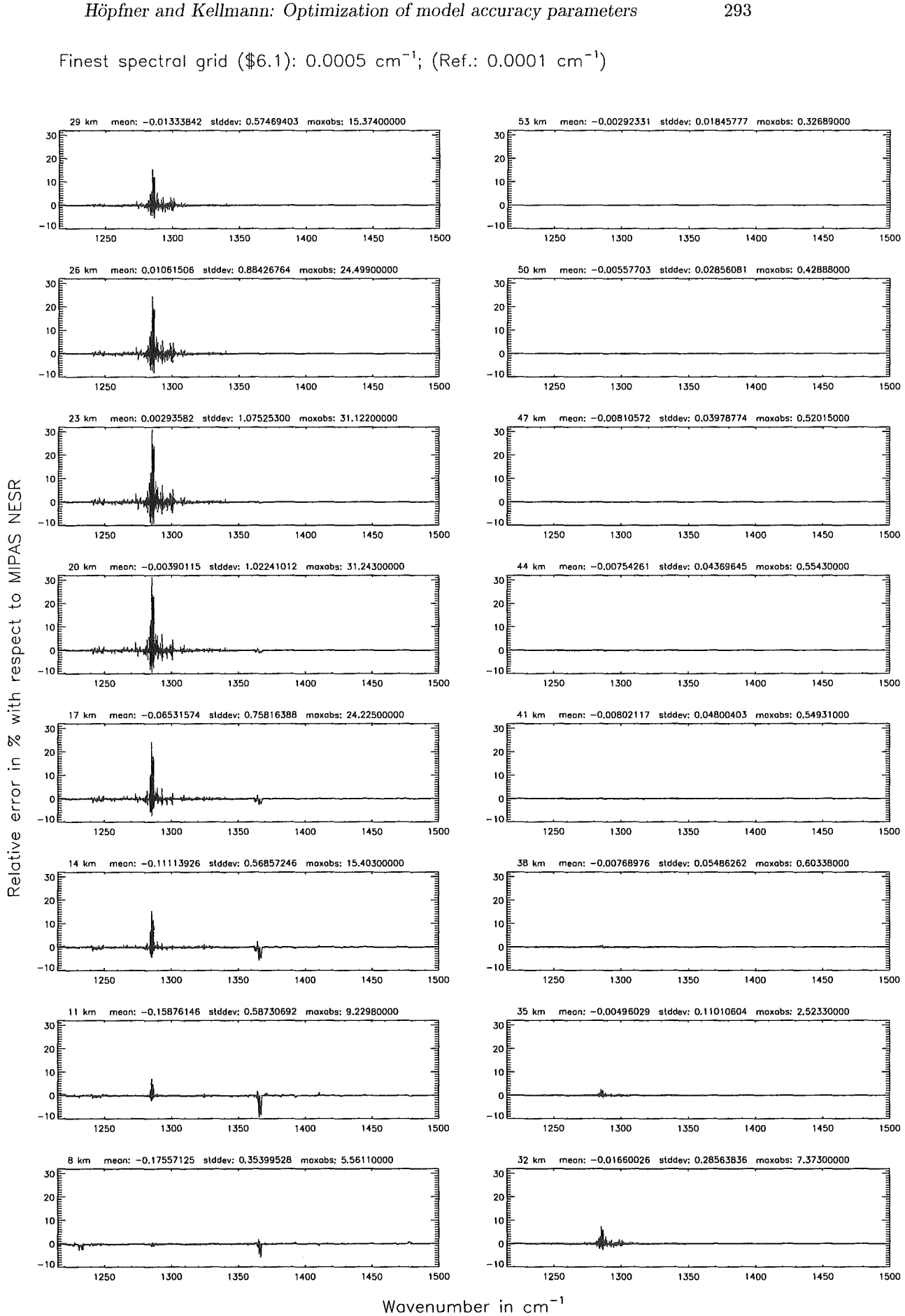
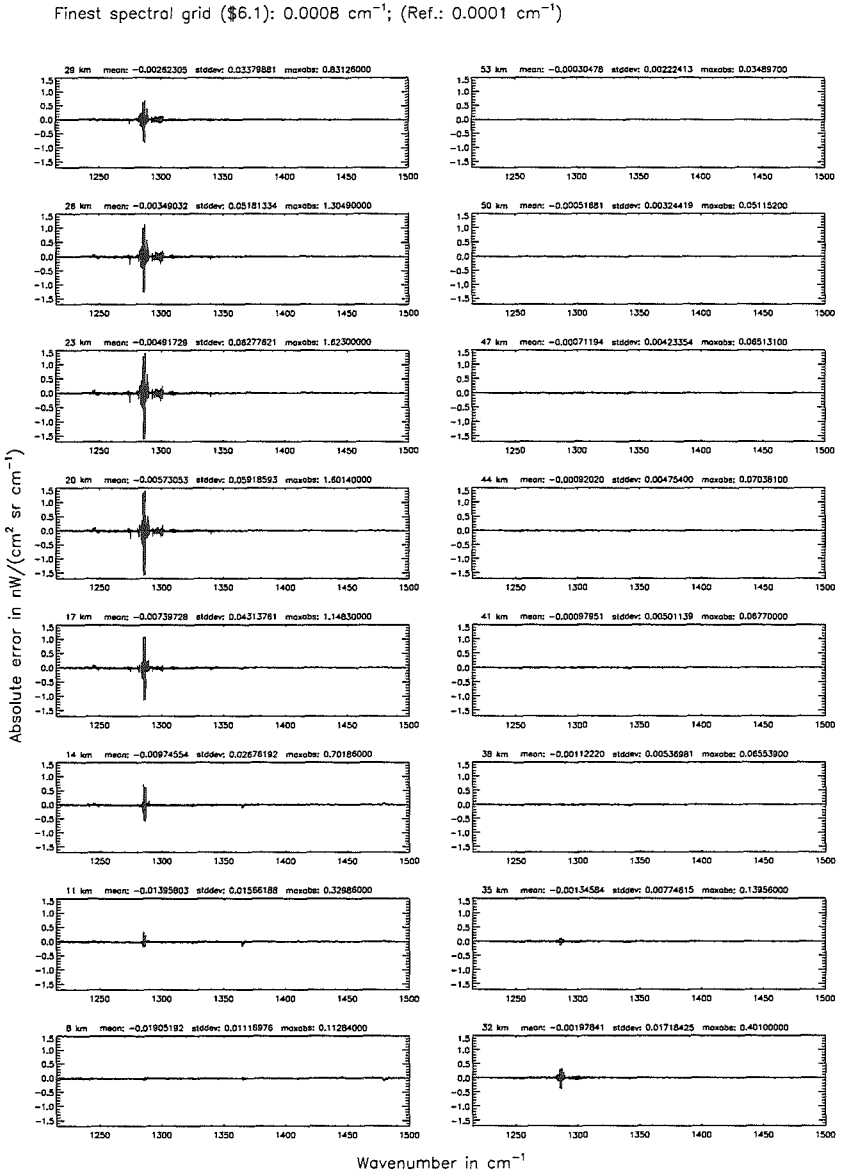


Figure 134: Relative error [%] with respect to MIPAS NESR



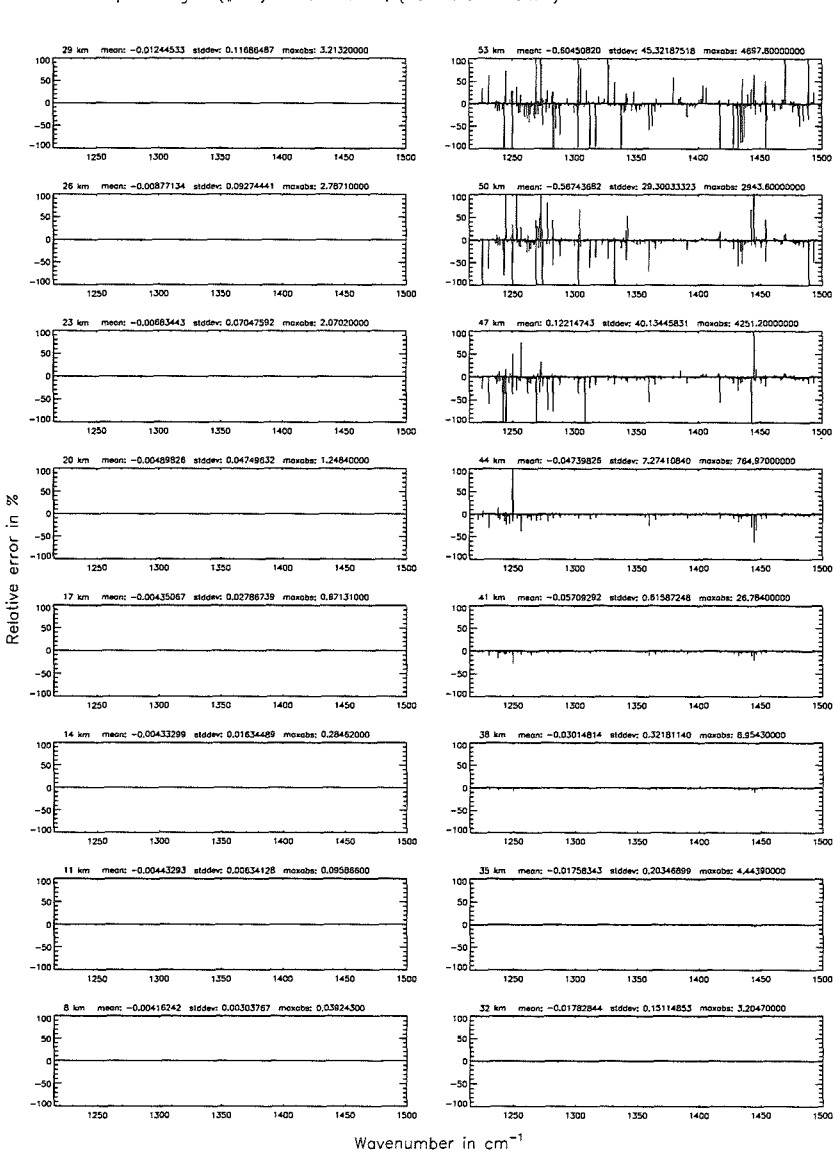


Figure 135: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

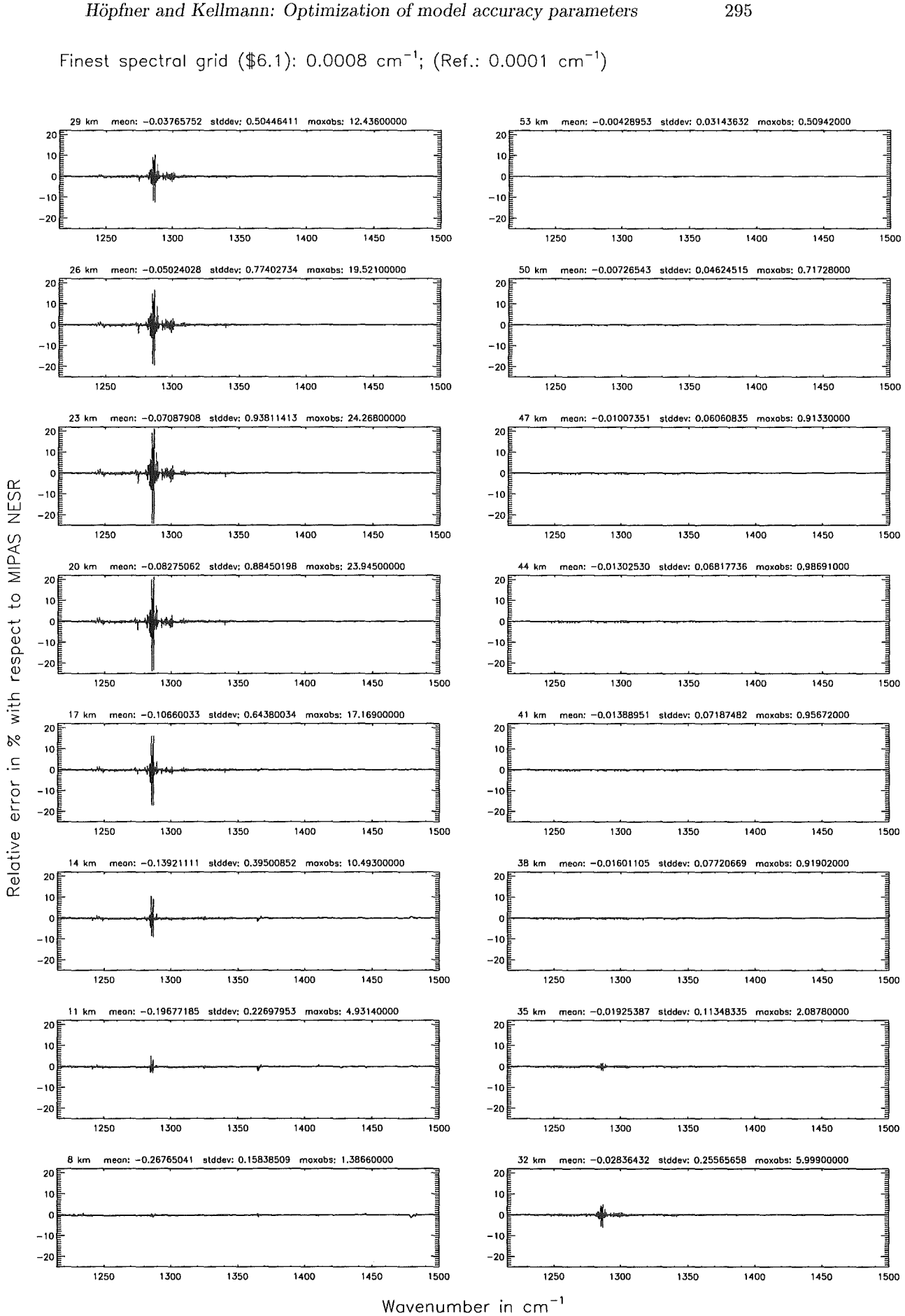
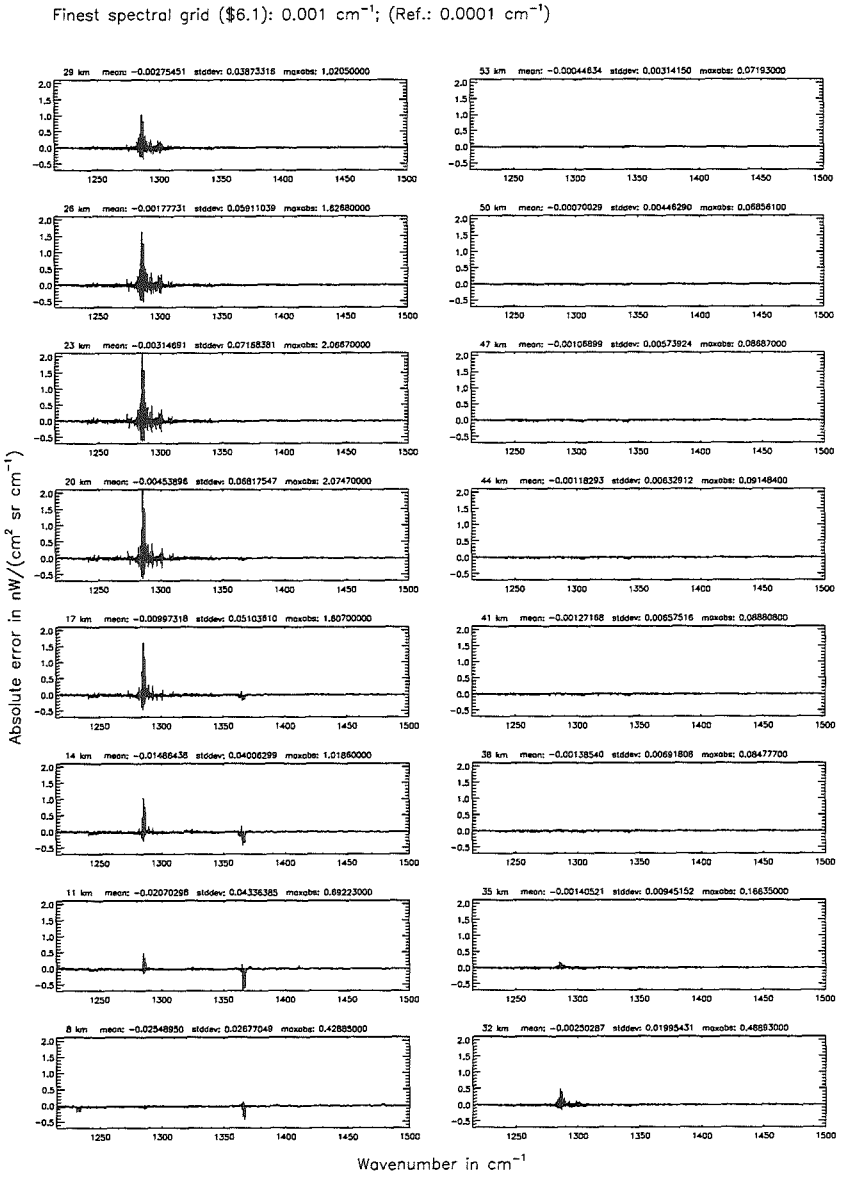


Figure 136: Relative error [%] with respect to MIPAS NESR



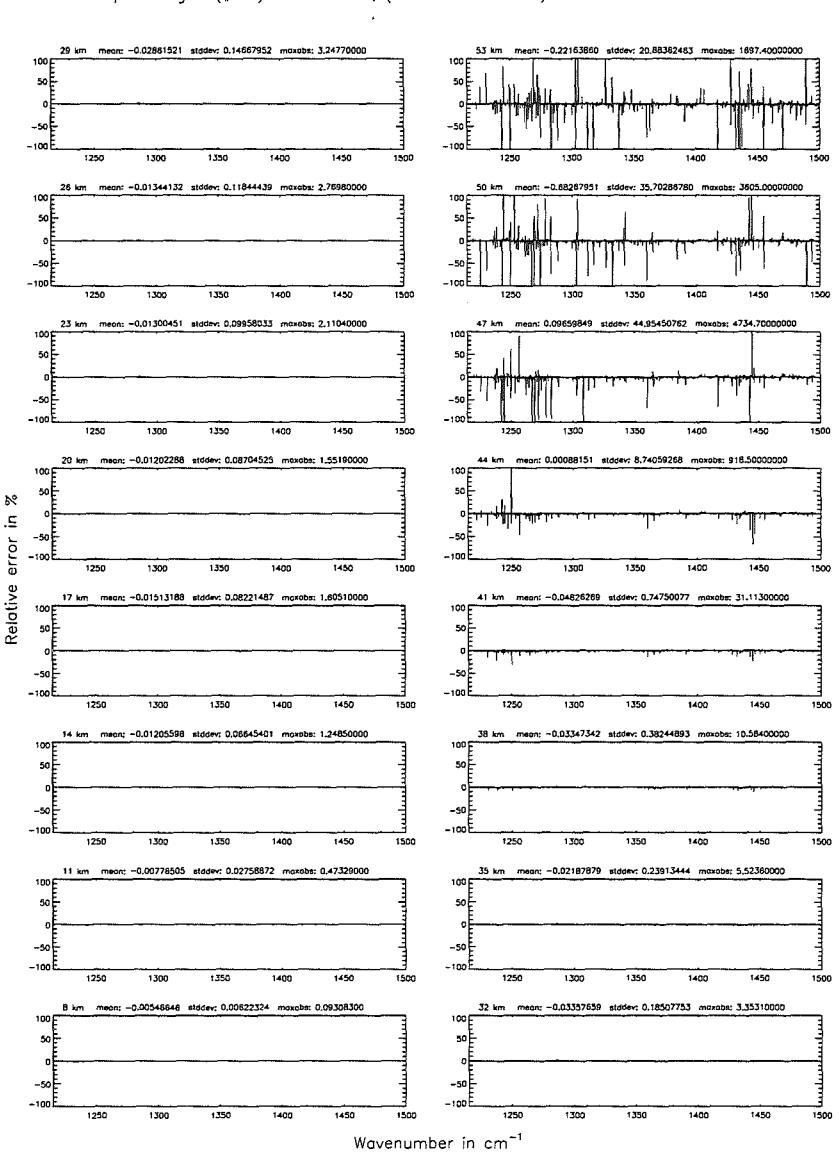


Figure 137: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

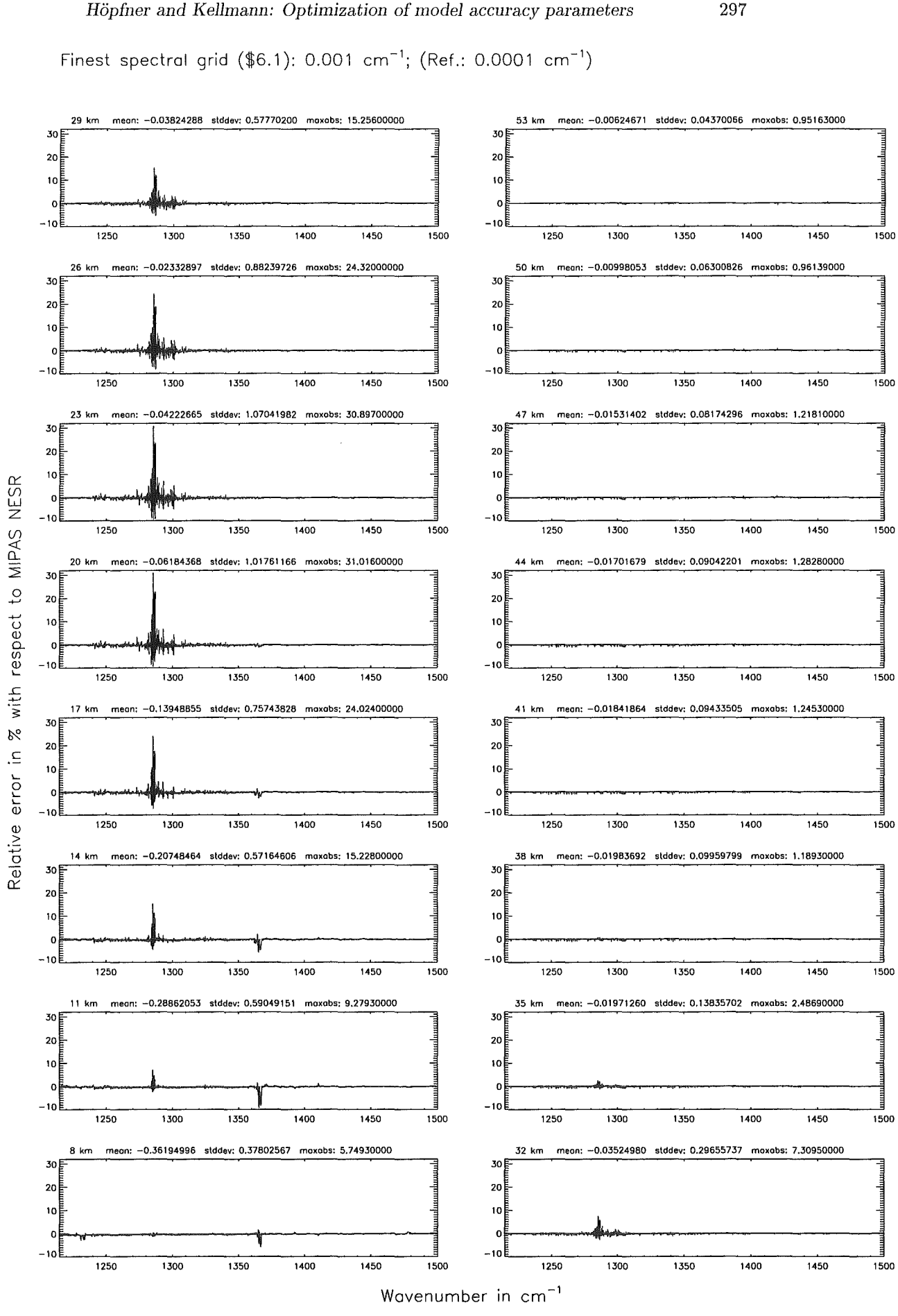
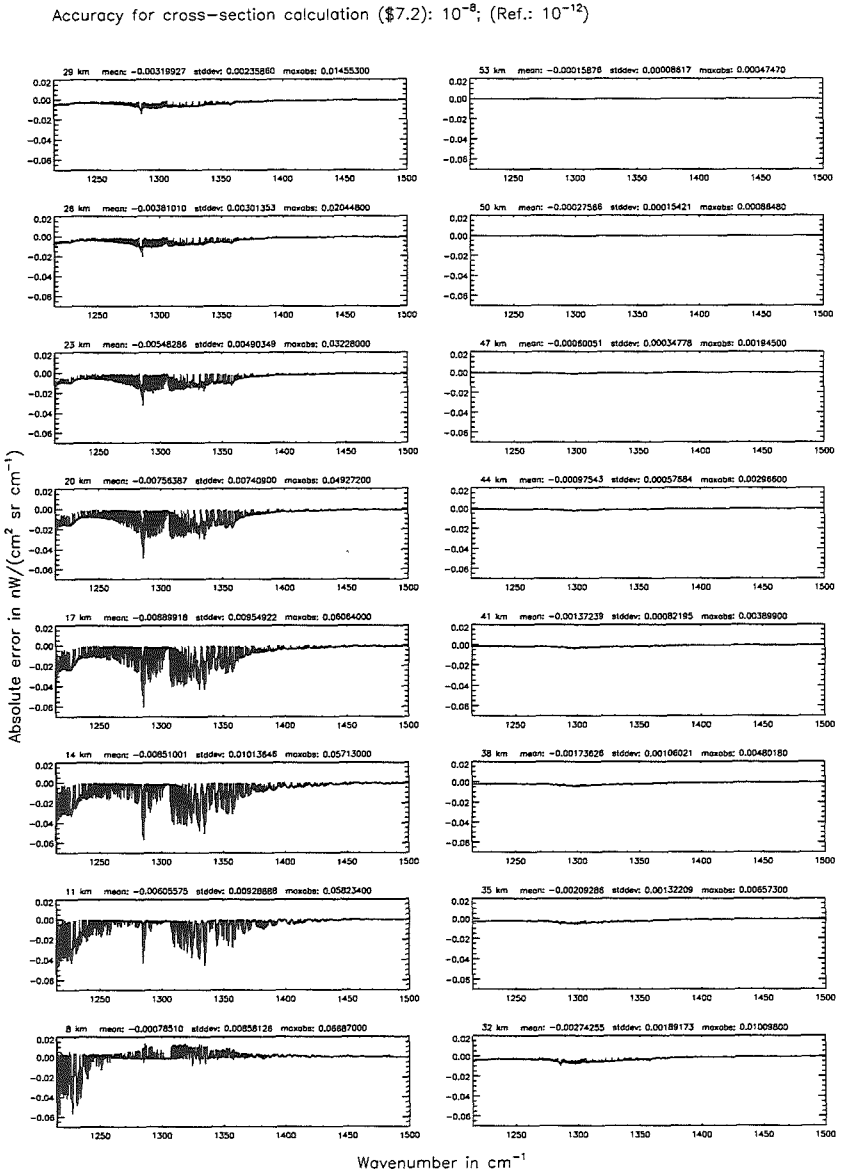


Figure 138: Relative error [%] with respect to MIPAS NESR



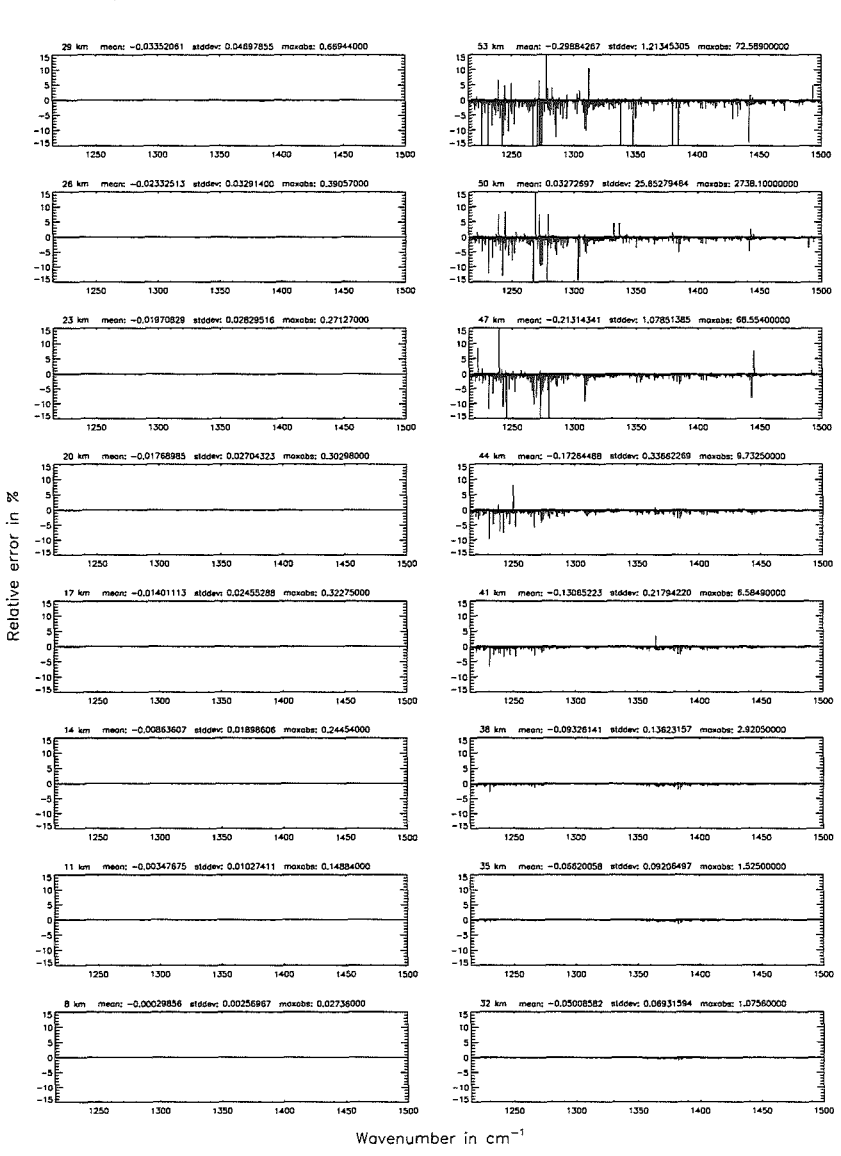


Figure 139: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

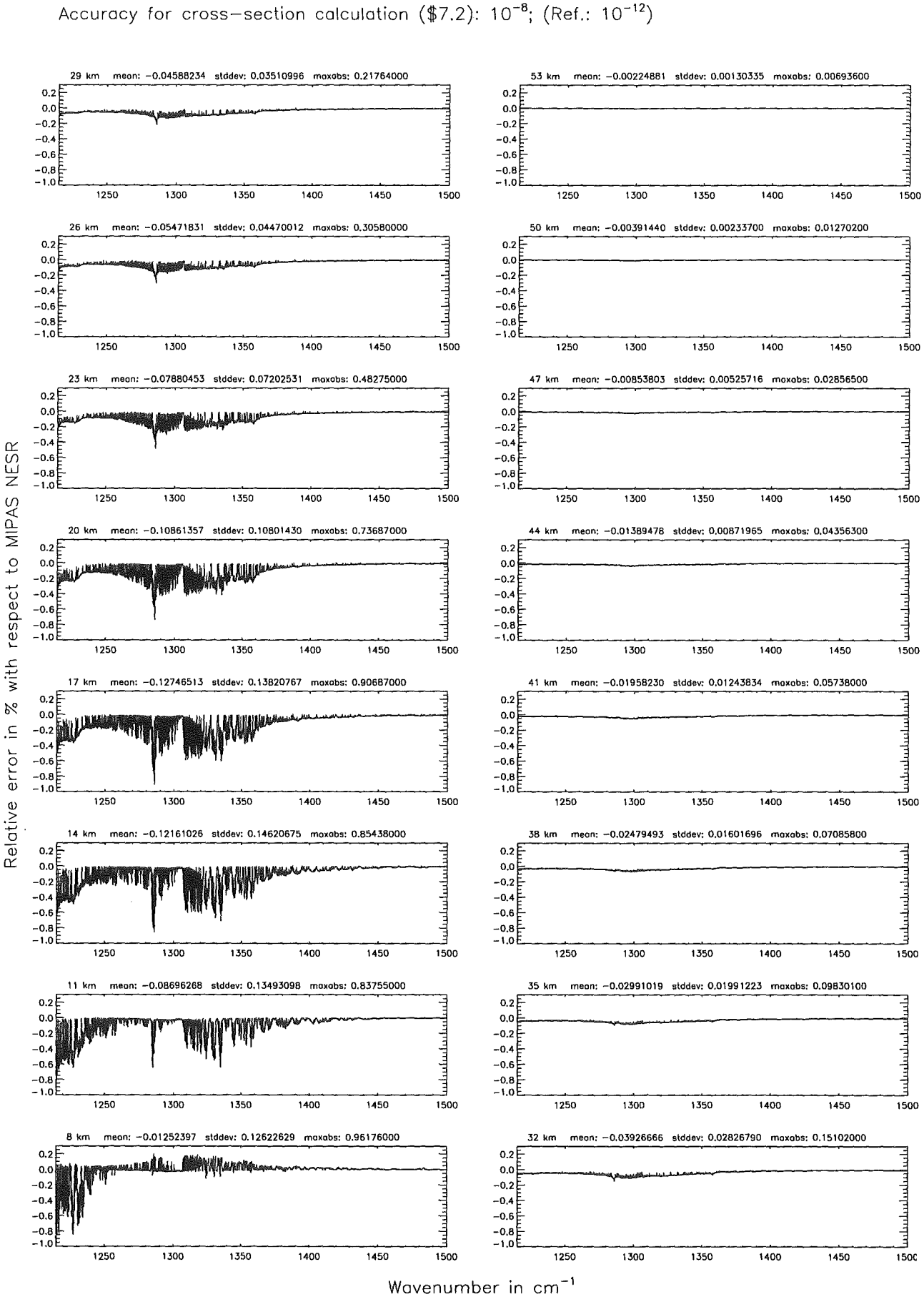
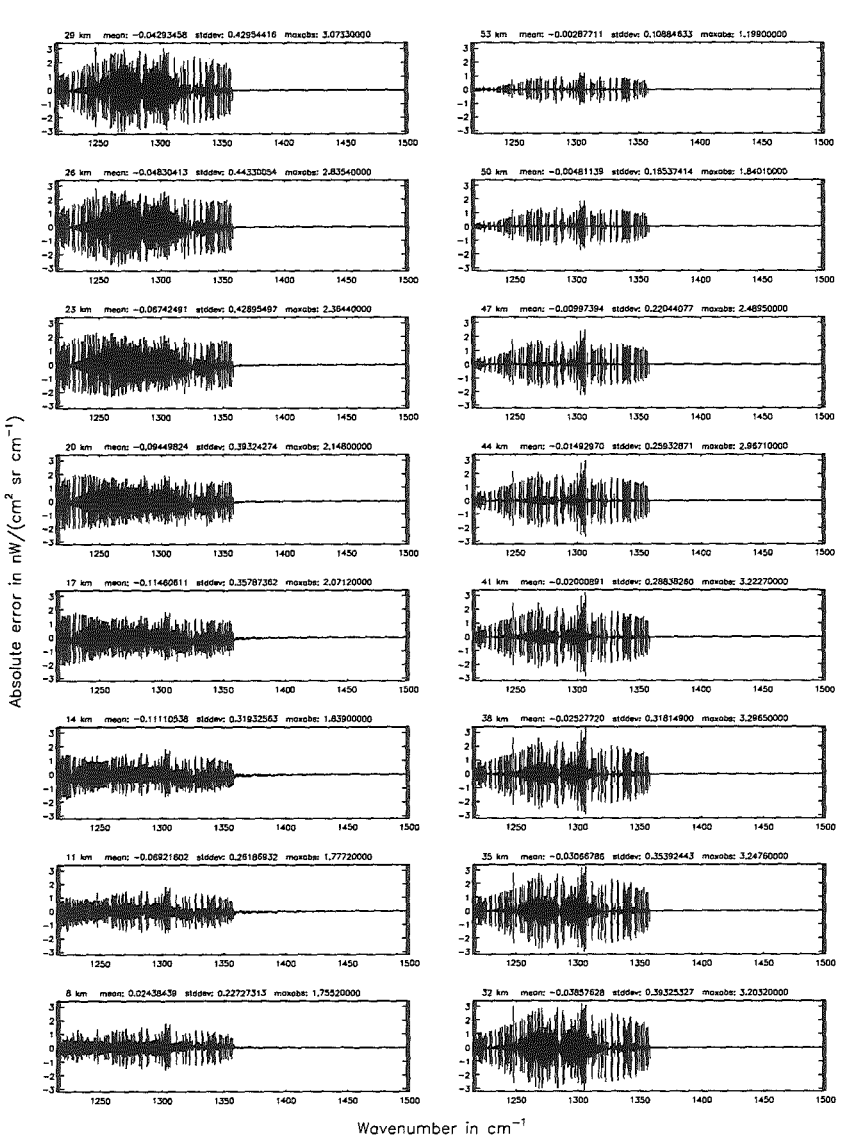


Figure 140: Relative error [%] with respect to MIPAS NESR



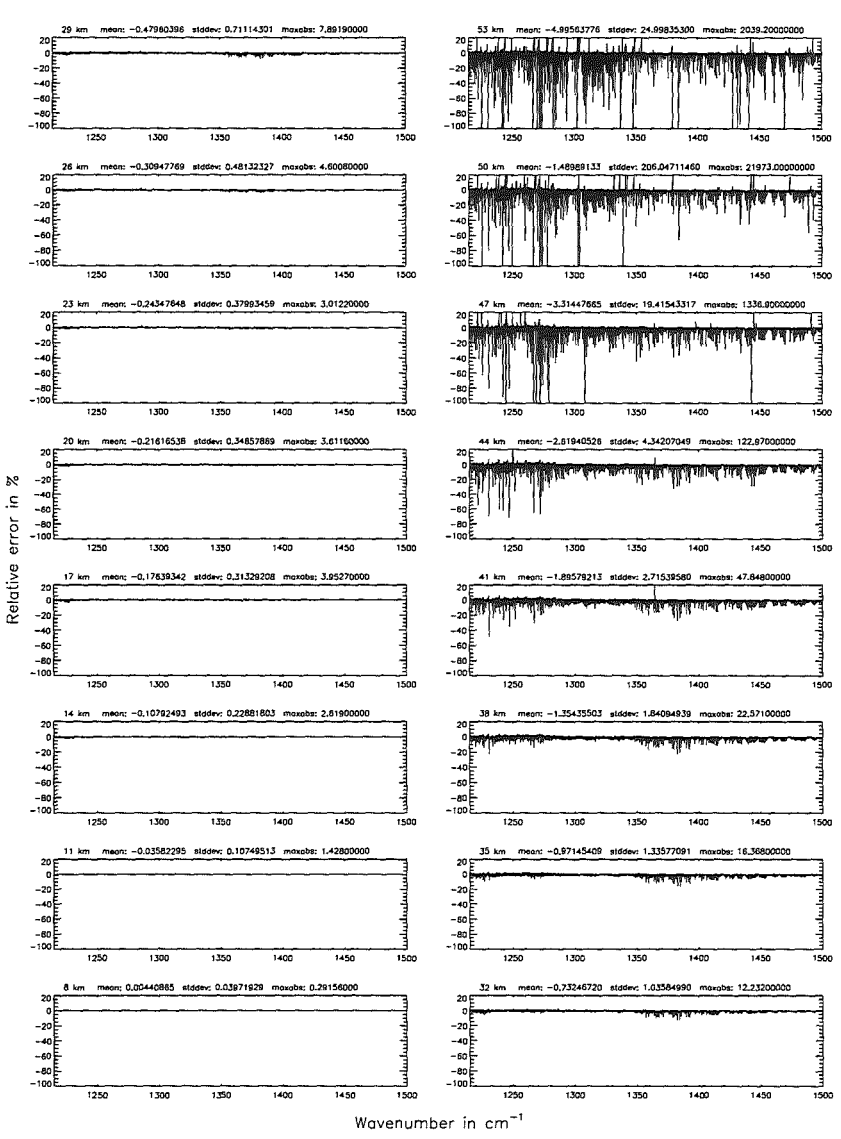


Figure 141: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

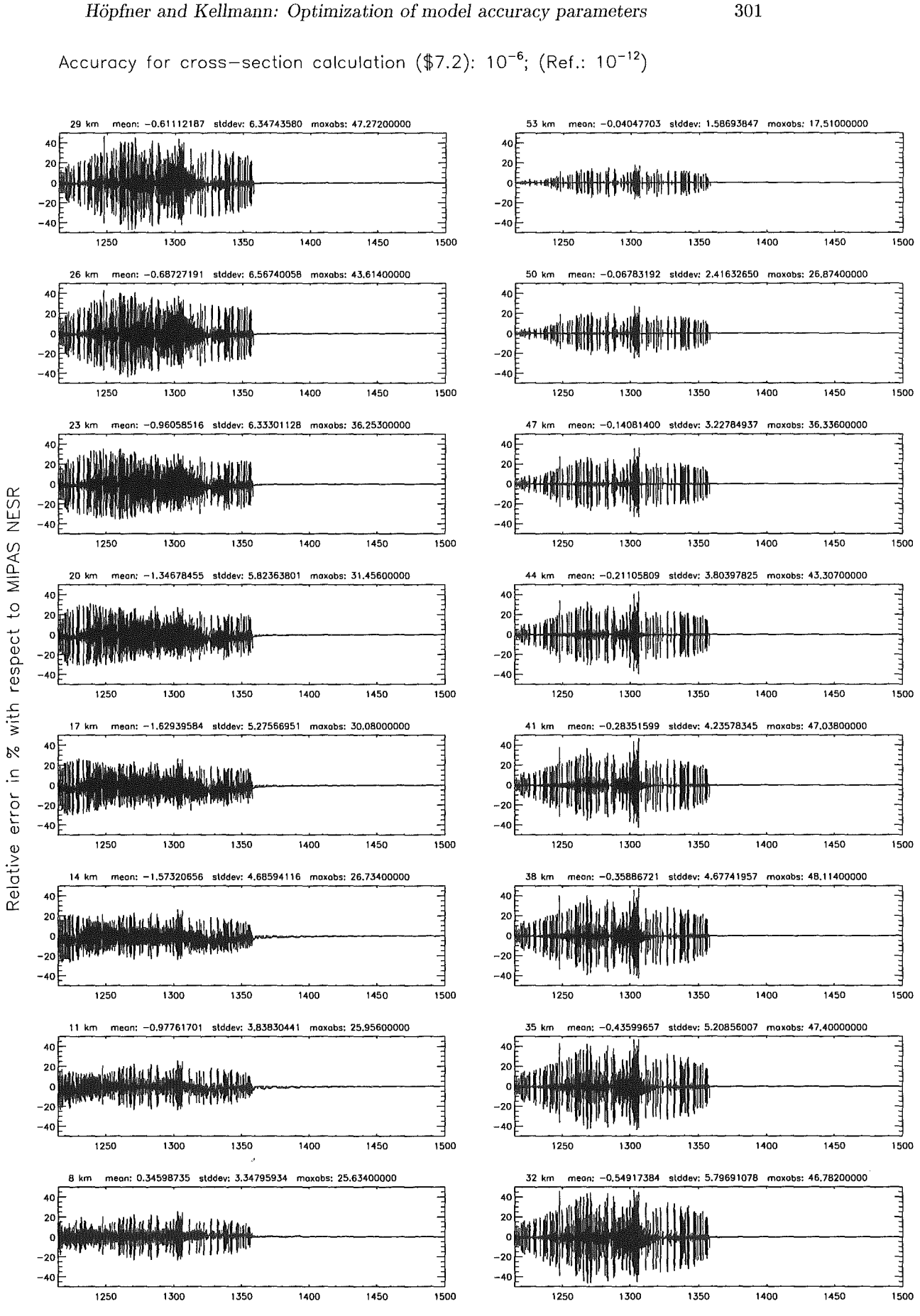
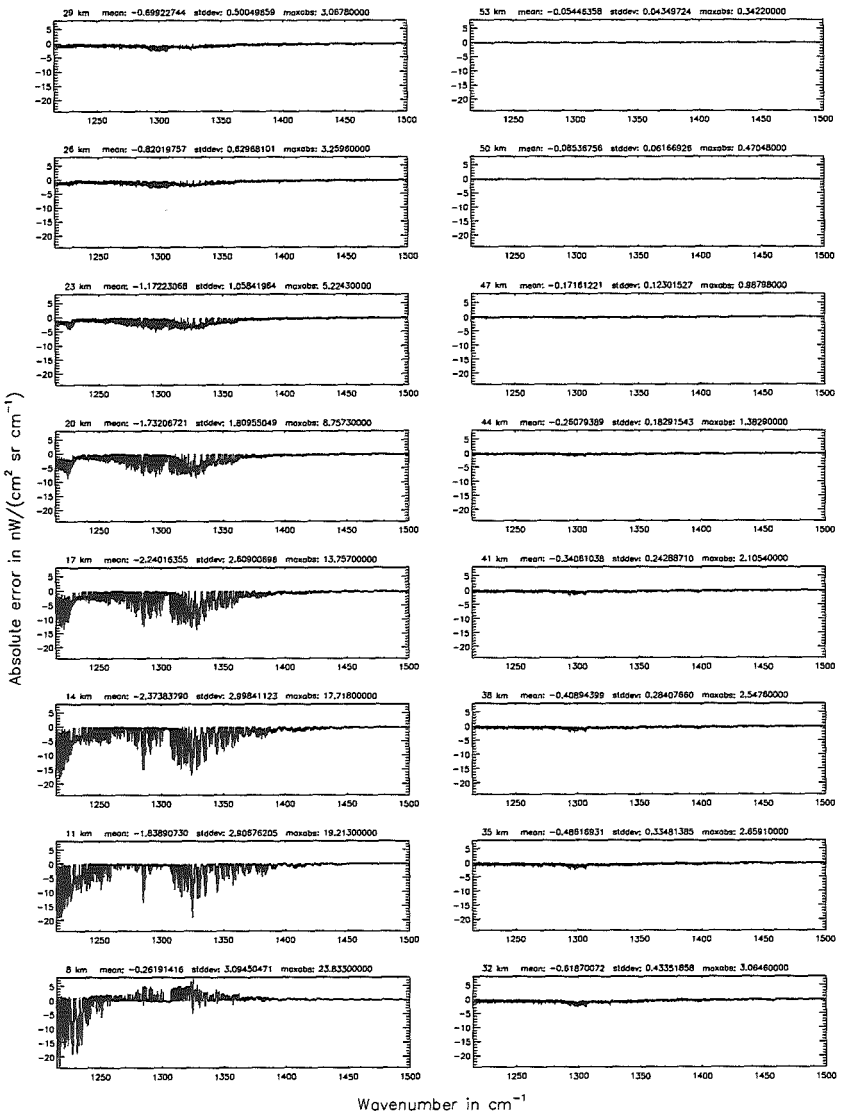


Figure 142: Relative error [%] with respect to MIPAS NESR



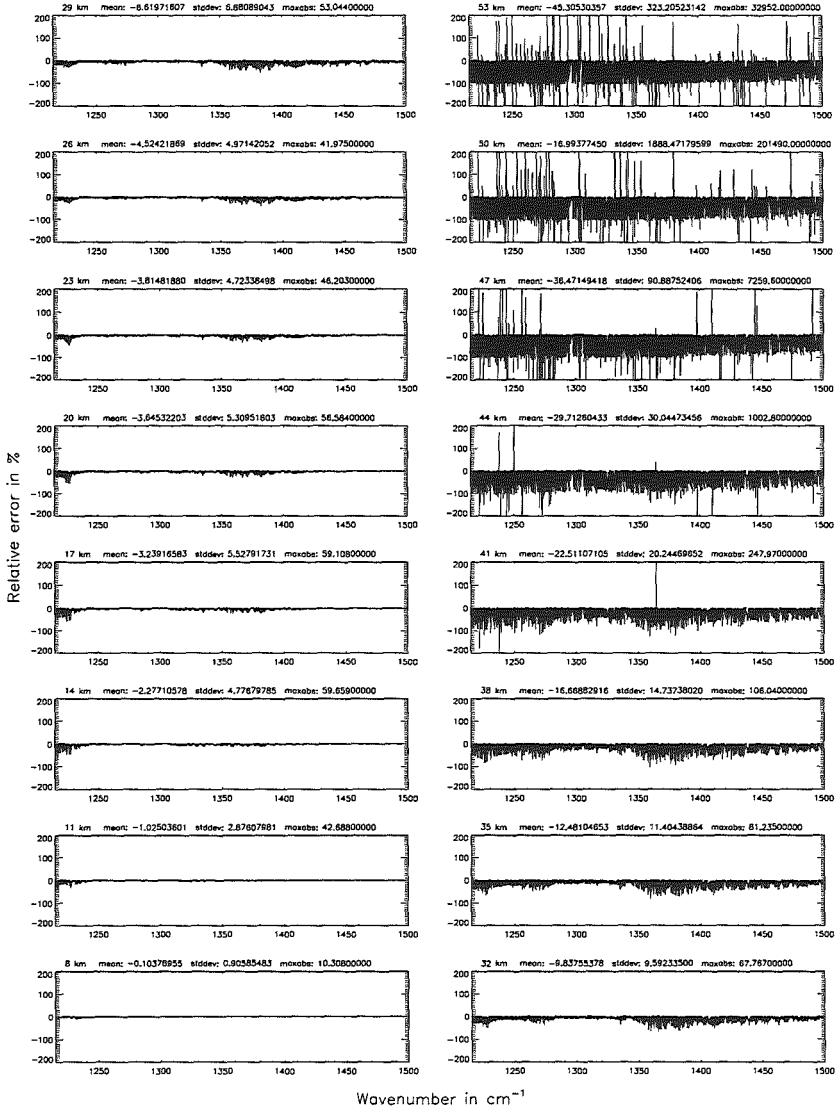


Figure 143: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

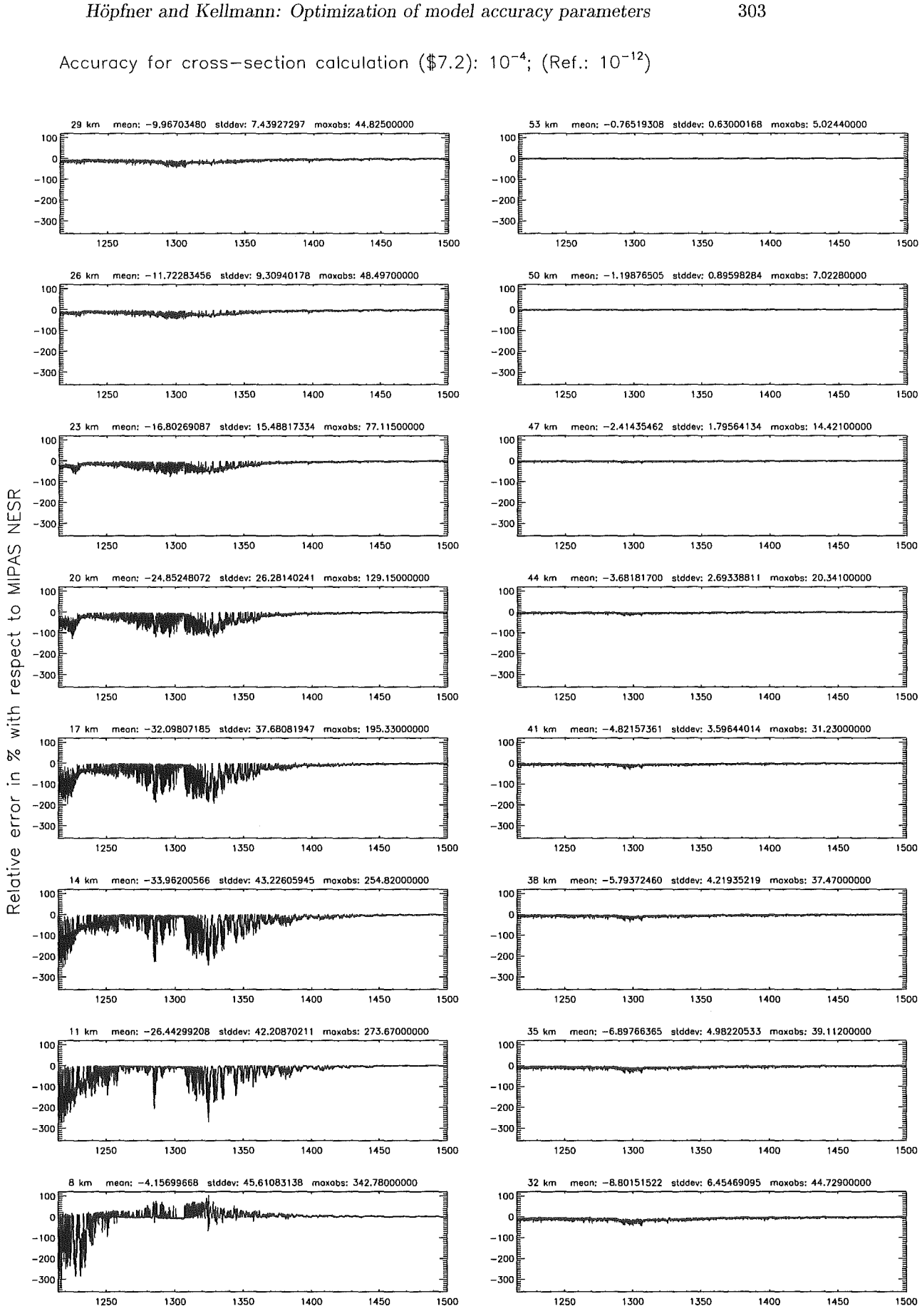
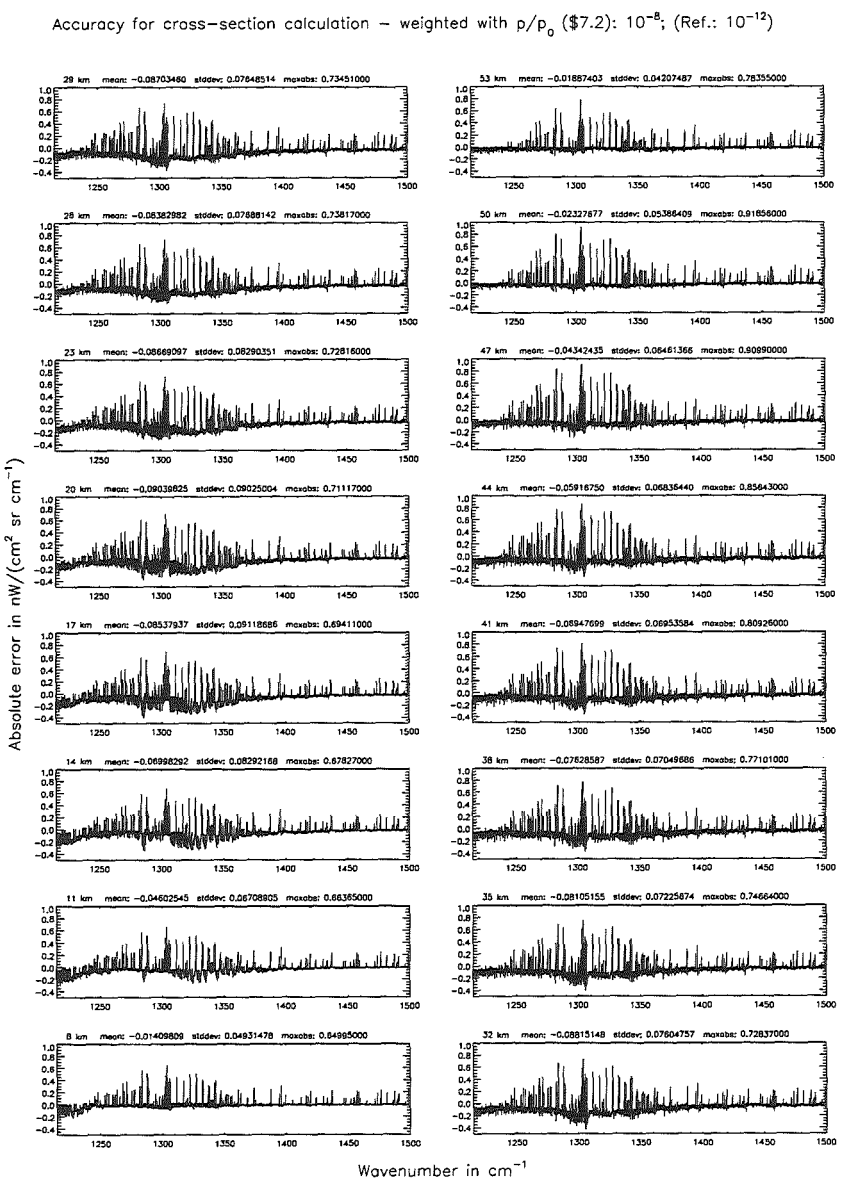


Figure 144: Relative error [%] with respect to MIPAS NESR



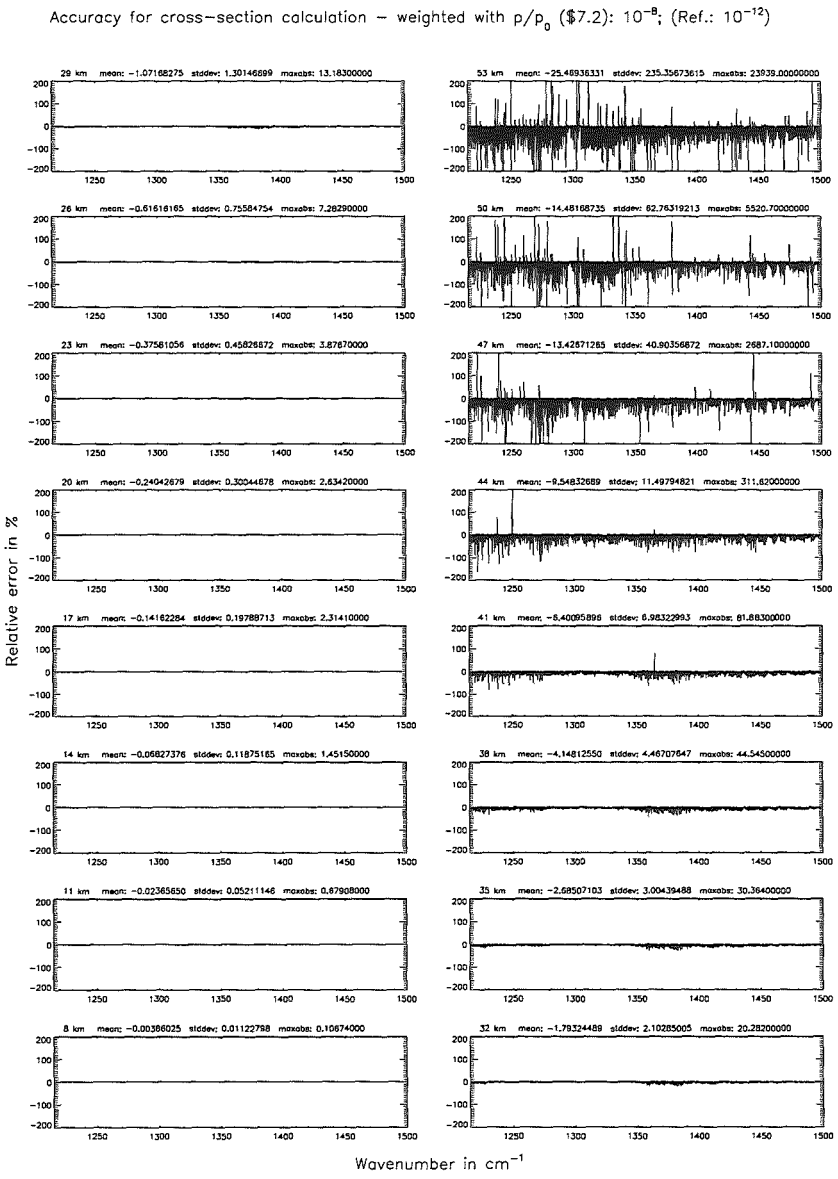


Figure 145: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

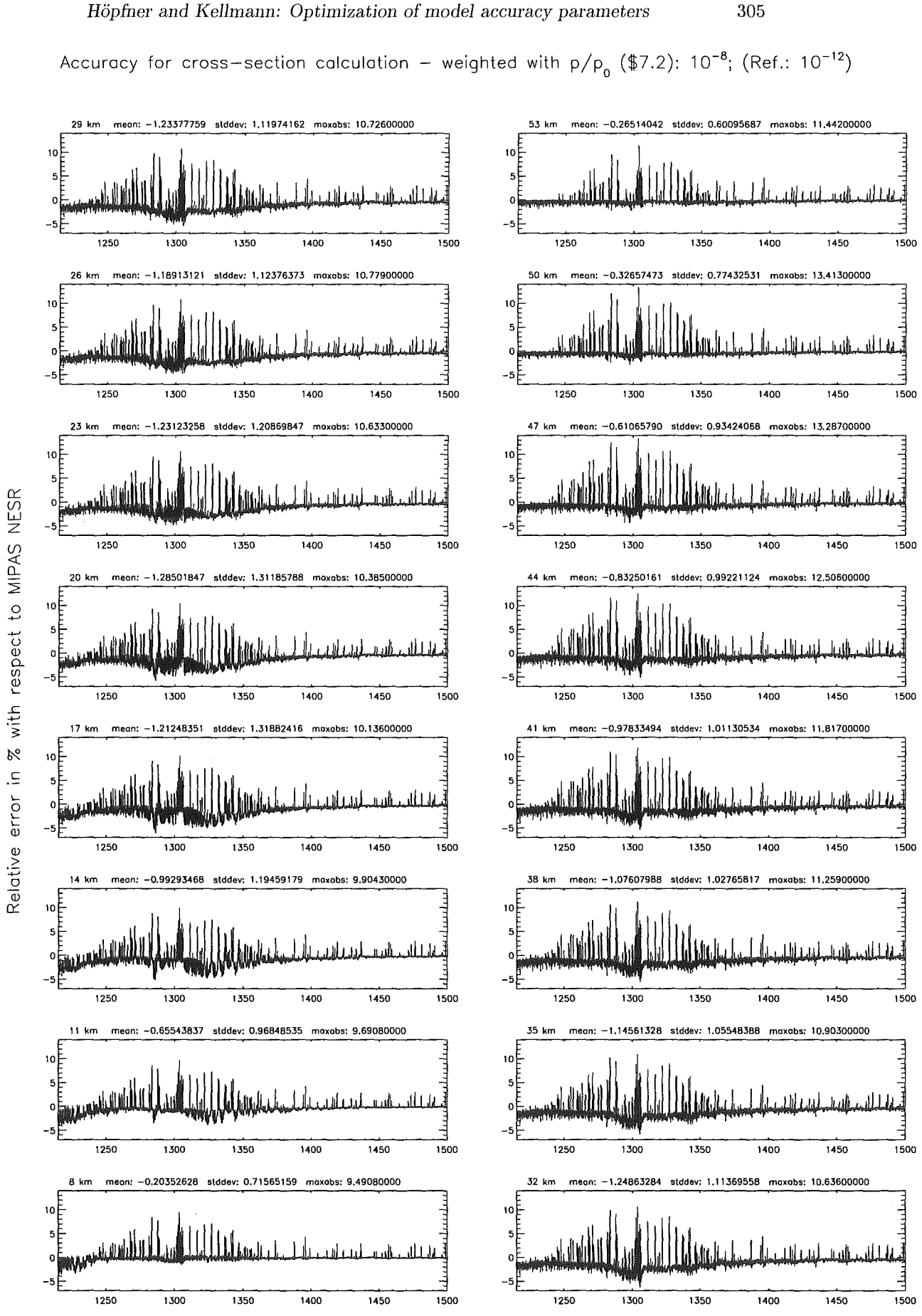


Figure 146: Relative error [%] with respect to MIPAS NESR

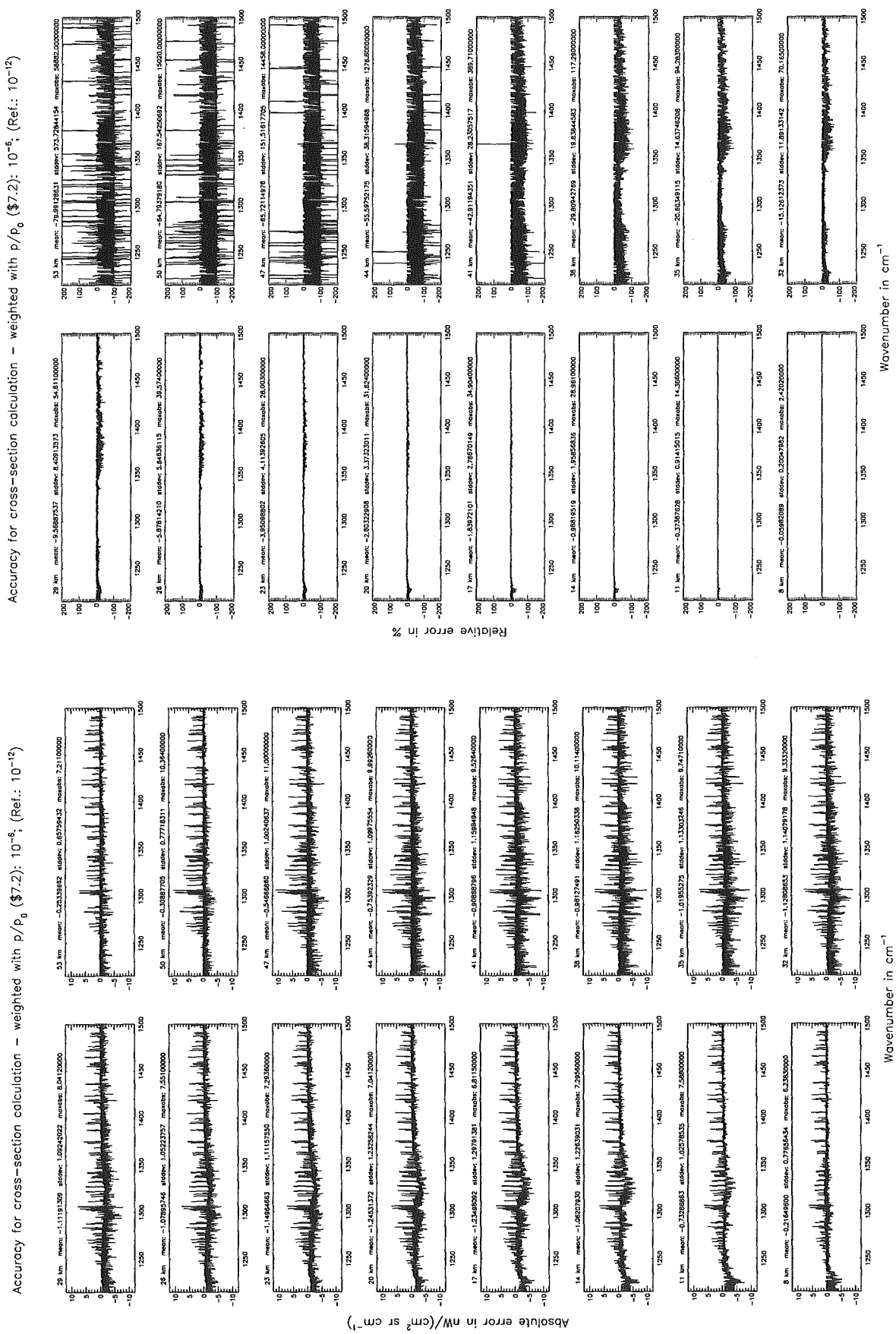


Figure 147: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

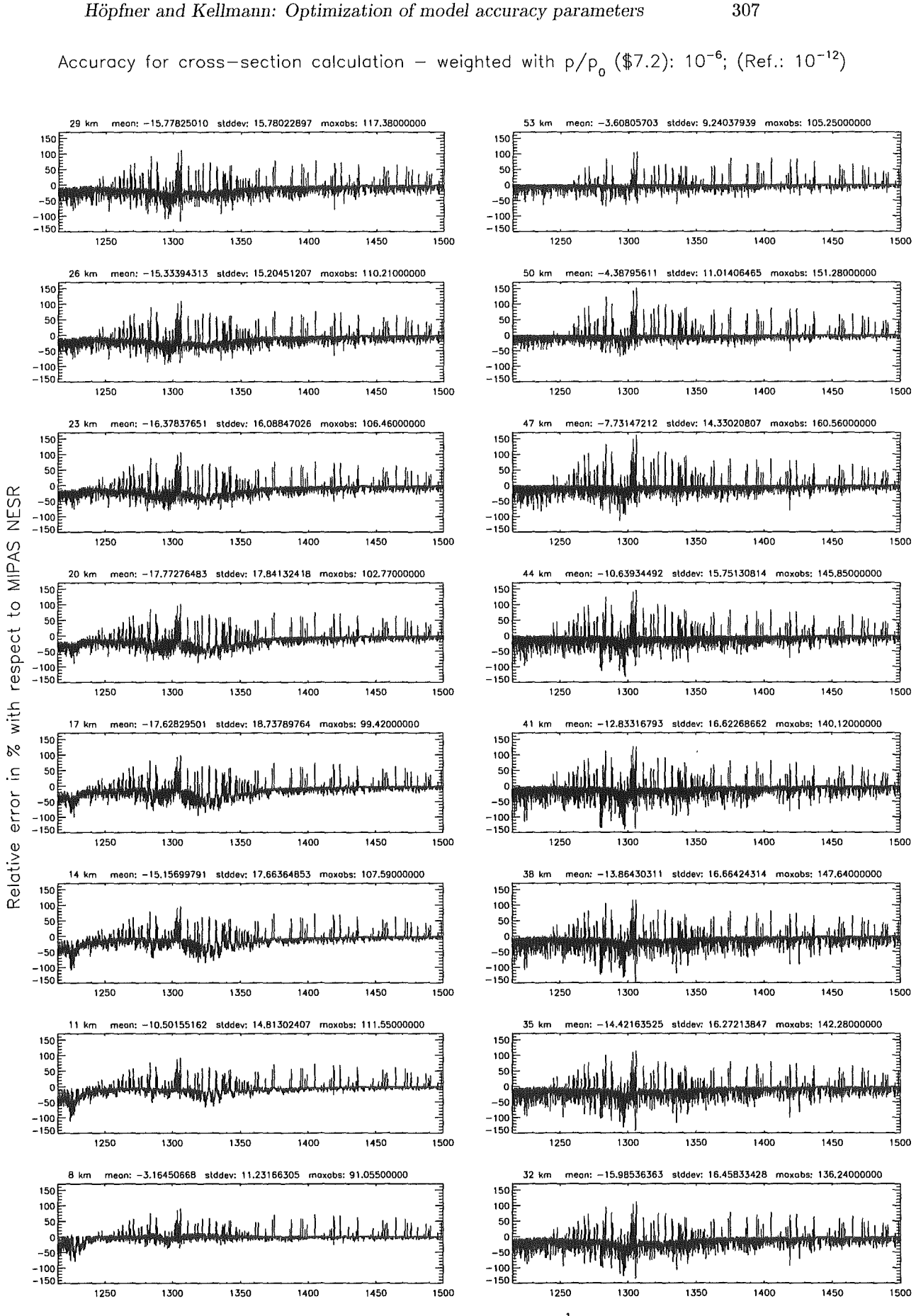
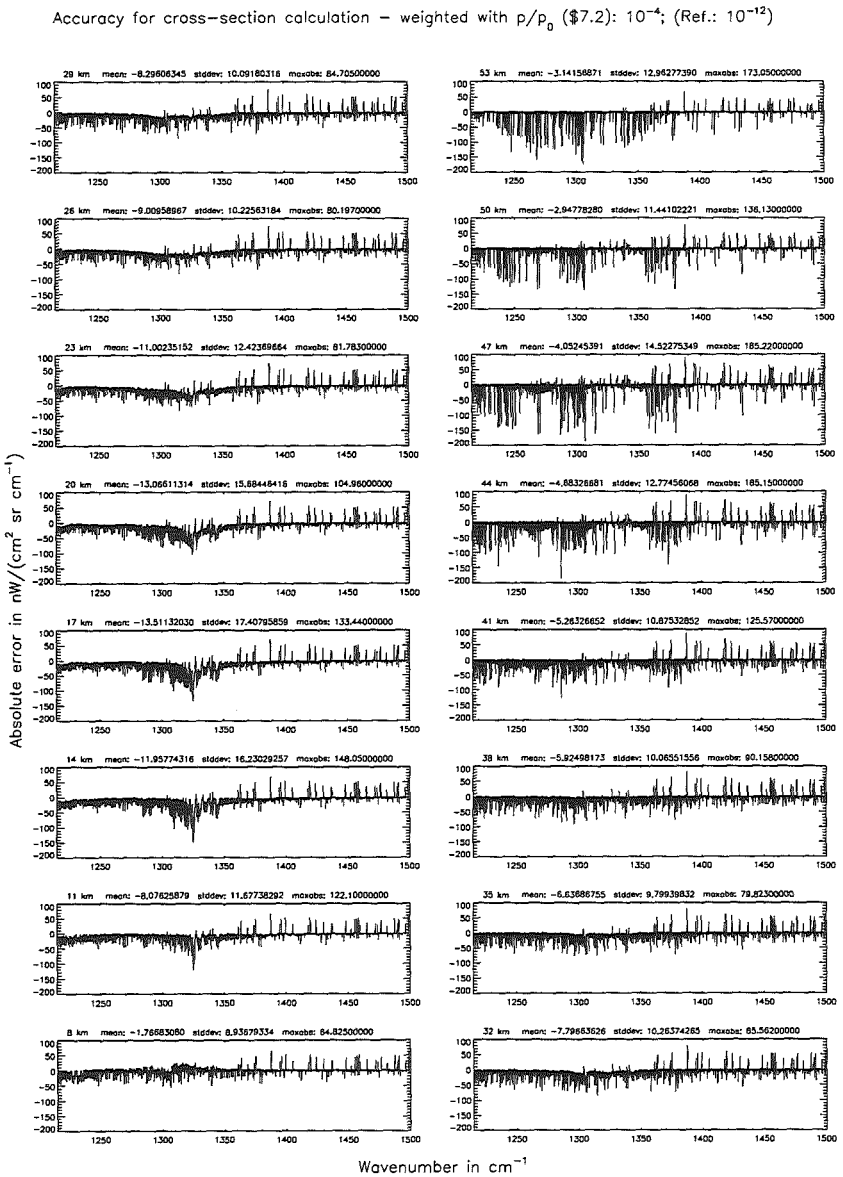


Figure 148: Relative error [%] with respect to MIPAS NESR



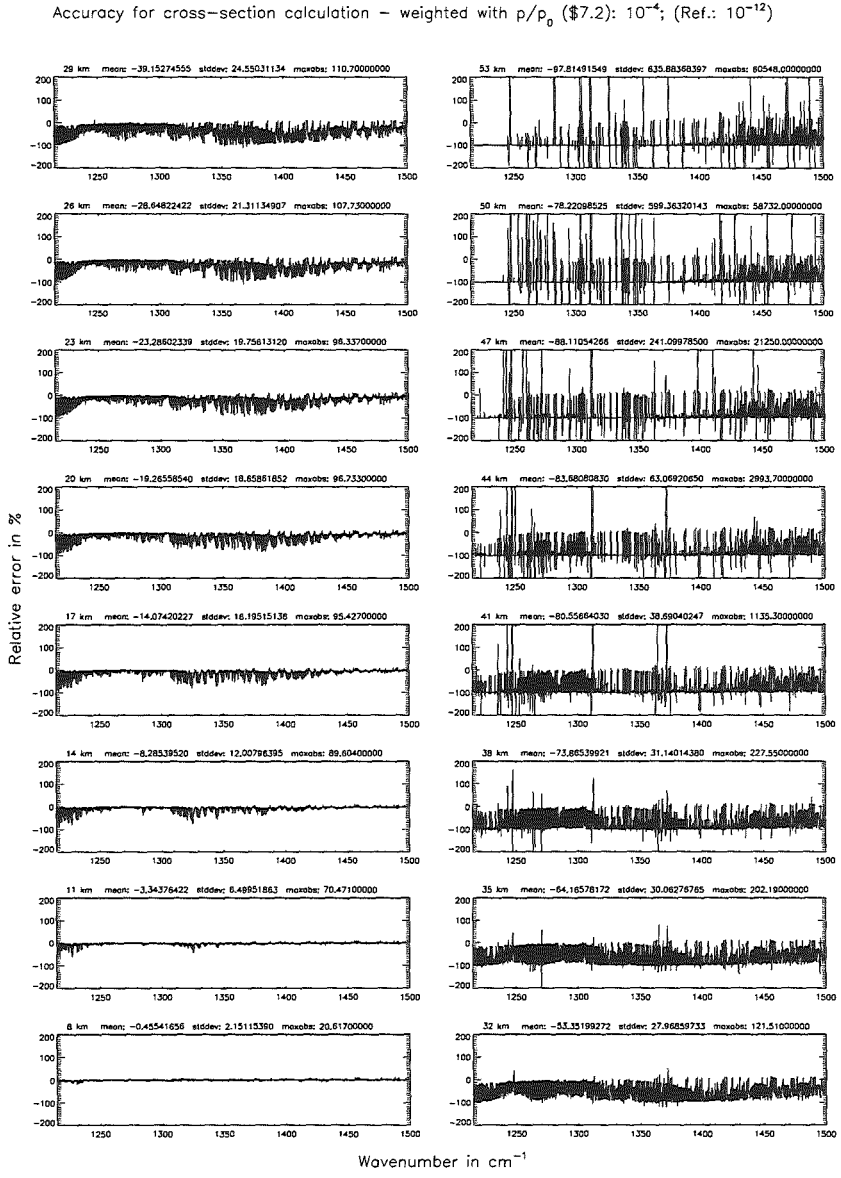


Figure 149: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

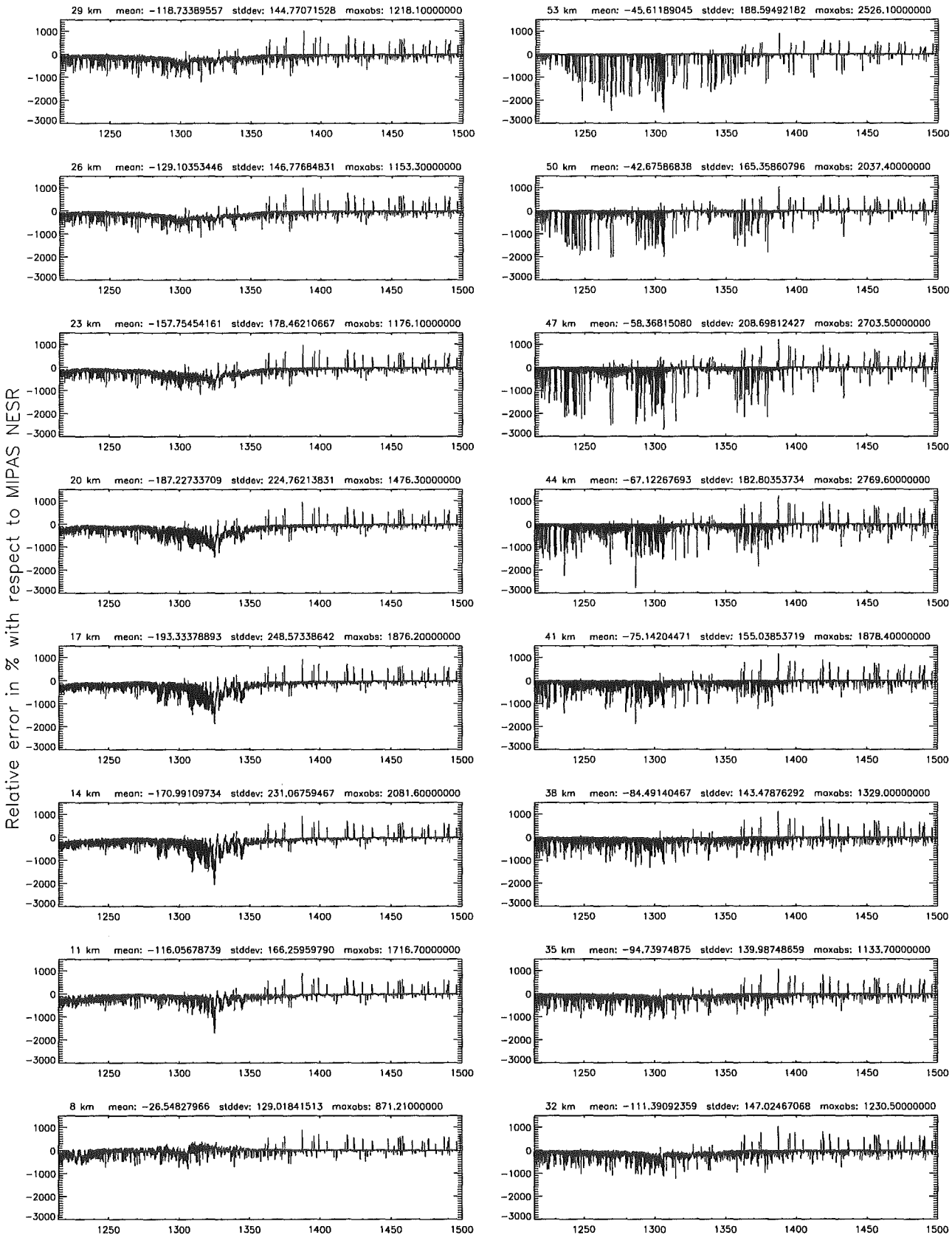
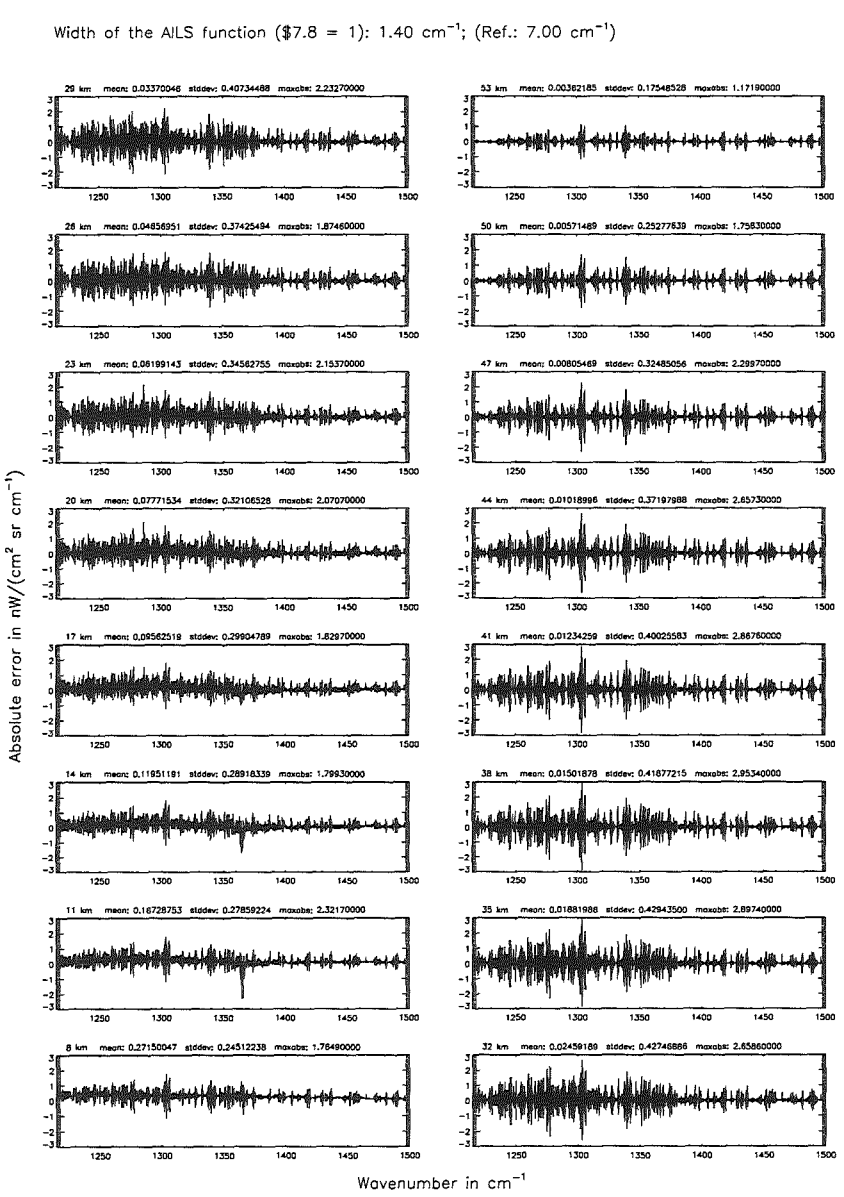


Figure 150: Relative error [%] with respect to MIPAS NESR



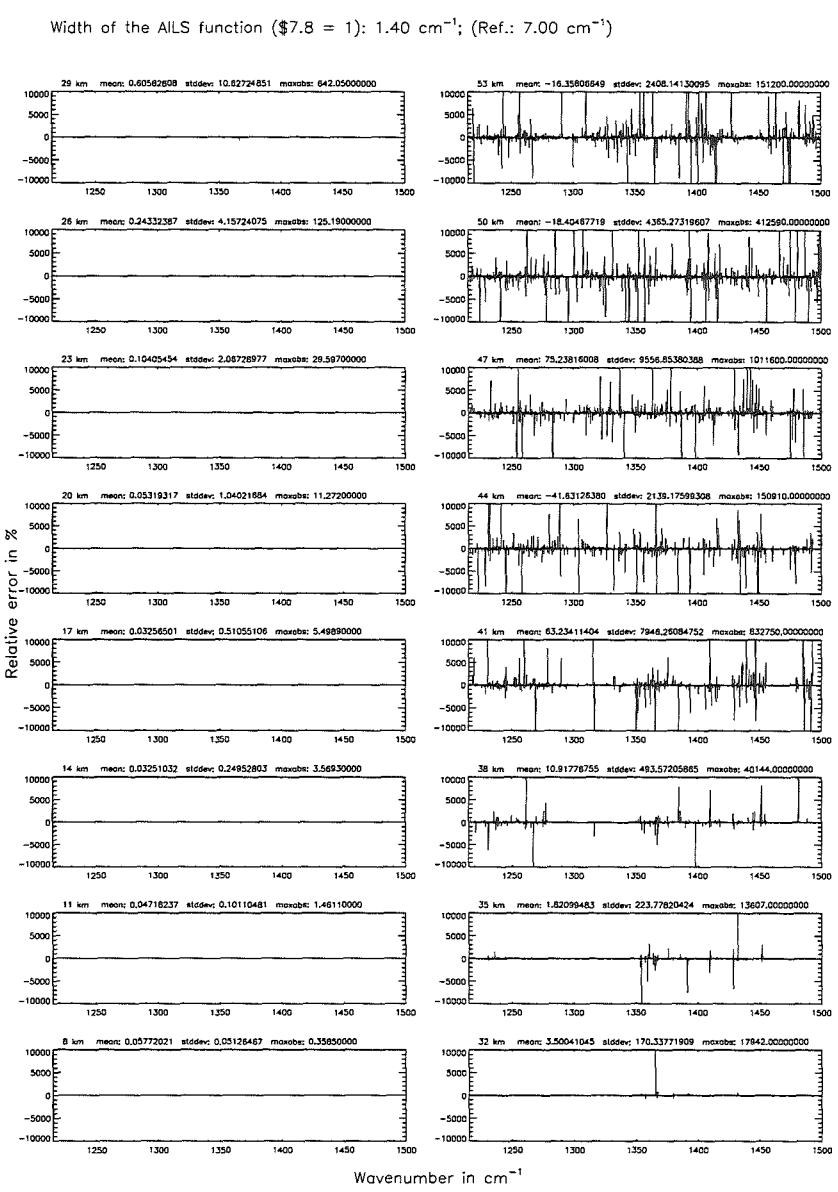


Figure 151: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

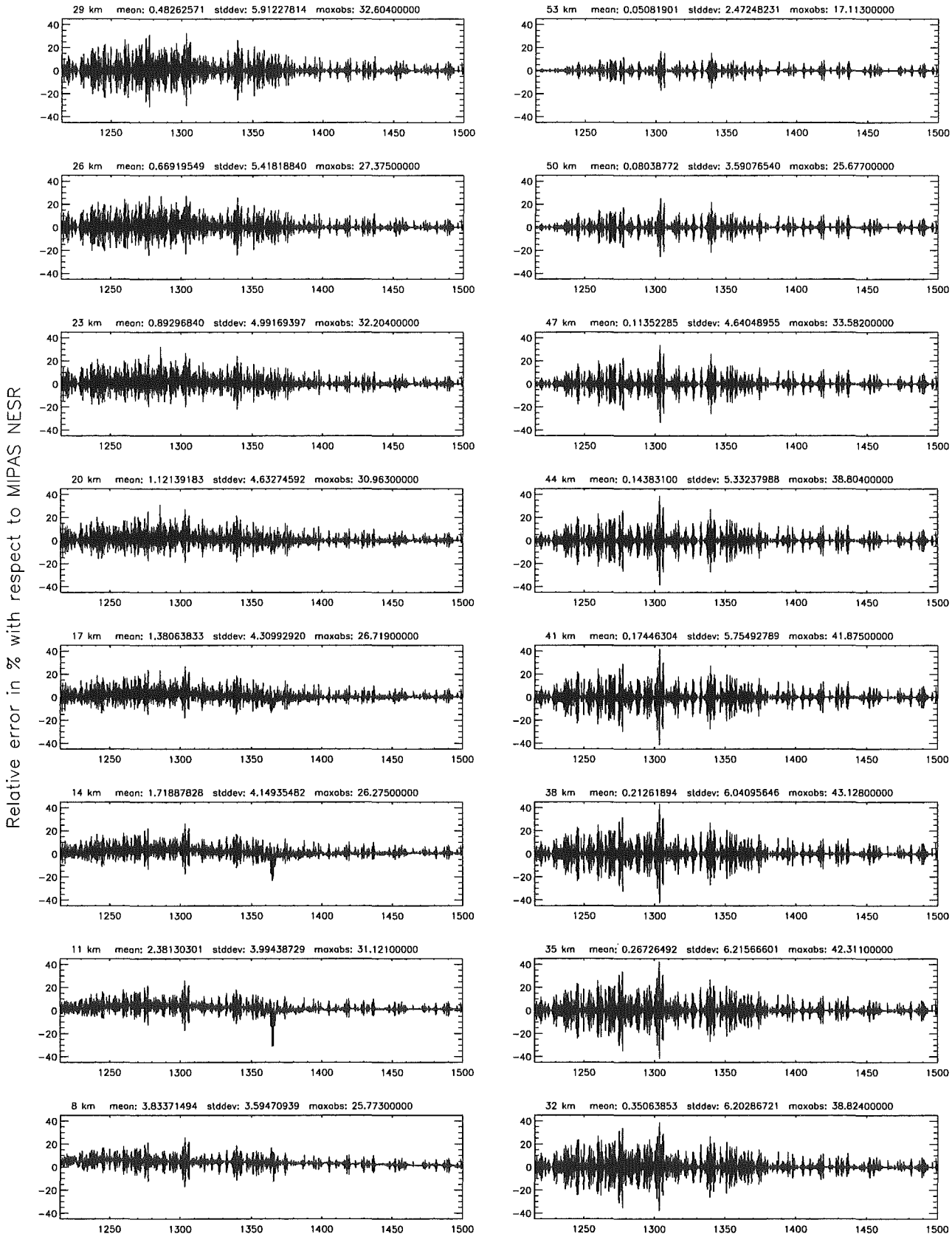
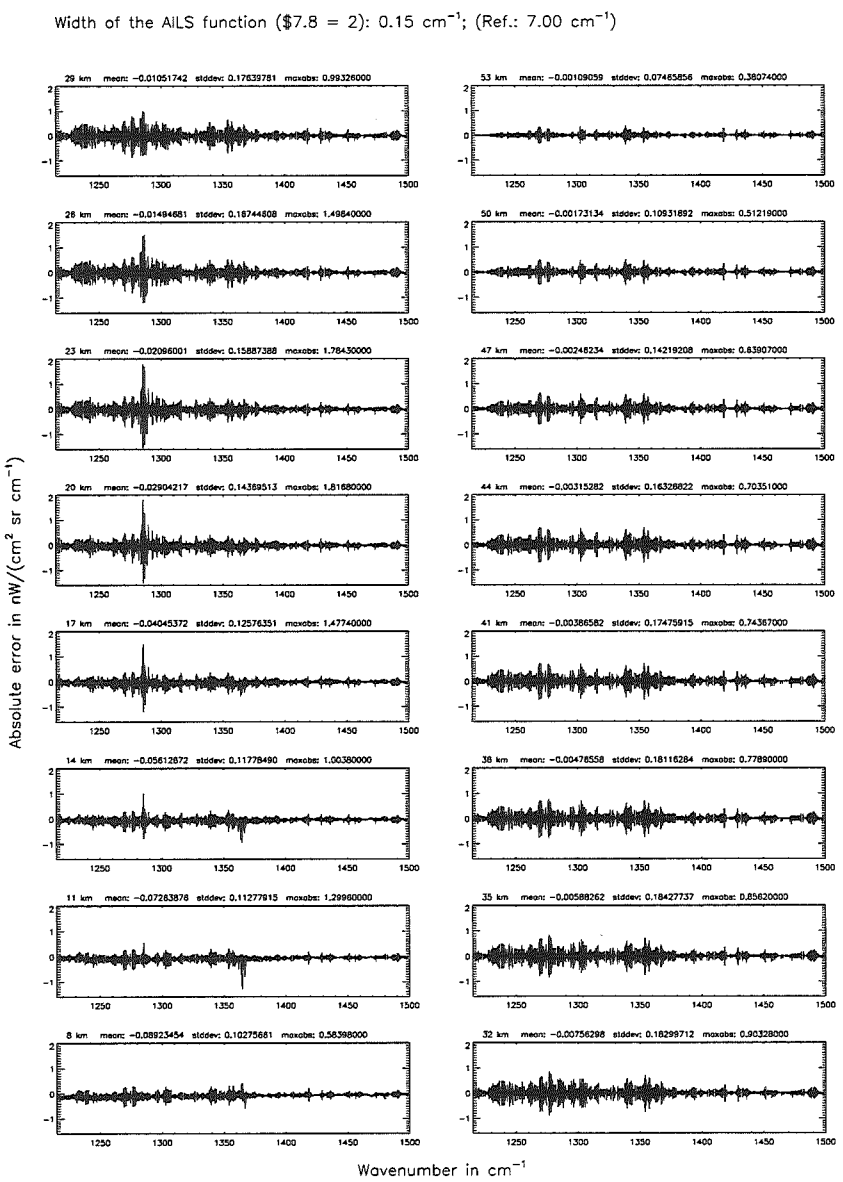


Figure 152: Relative error [%] with respect to MIPAS NESR



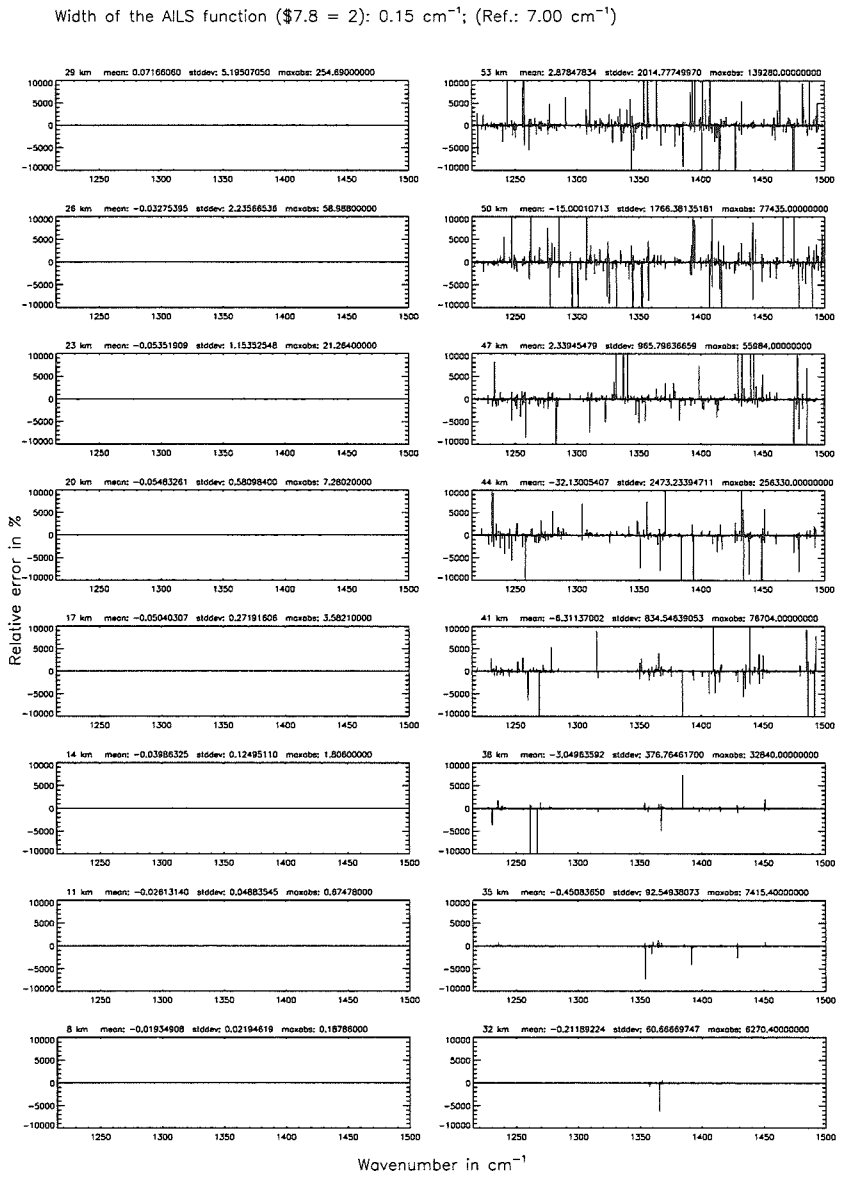


Figure 153: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

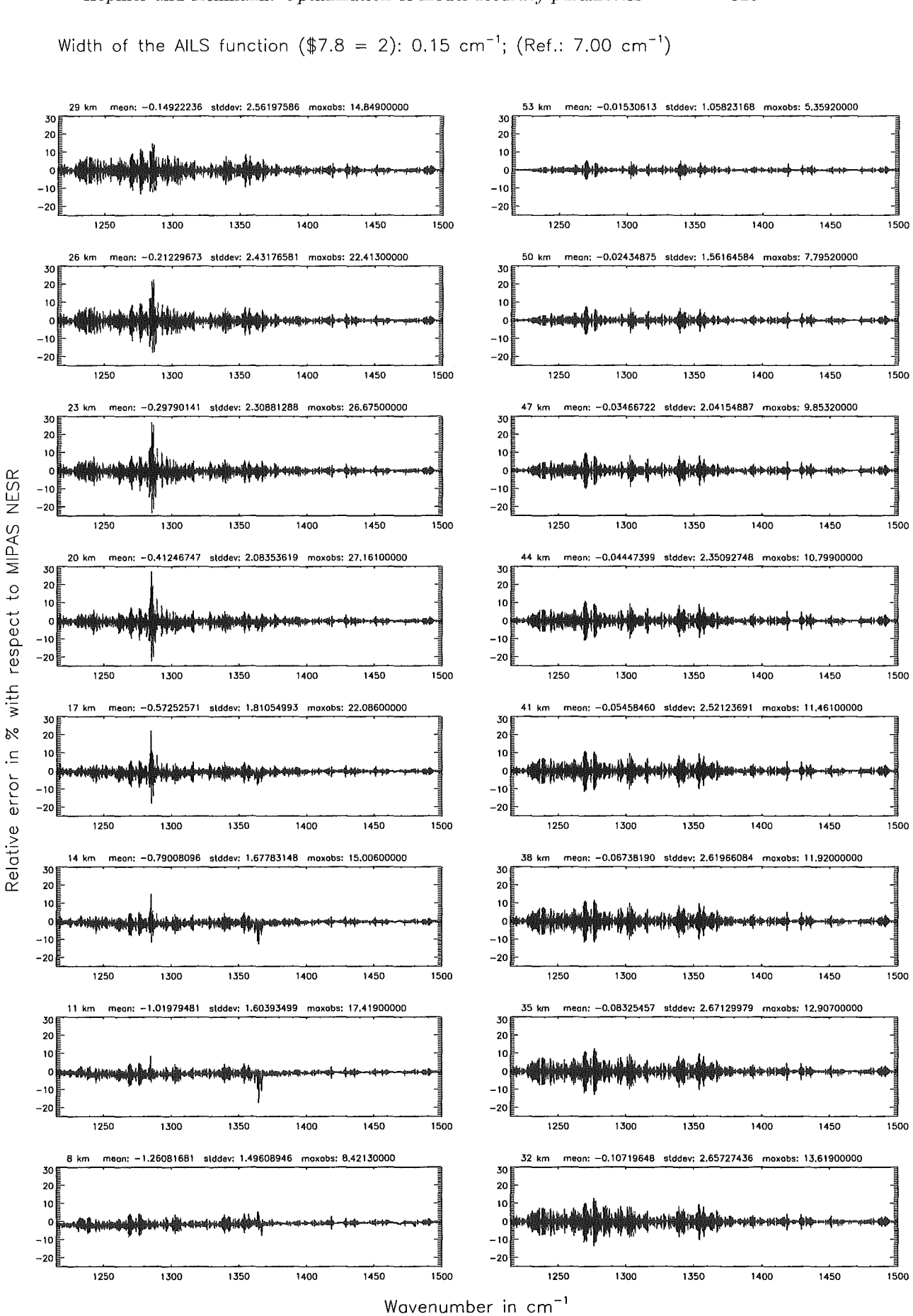
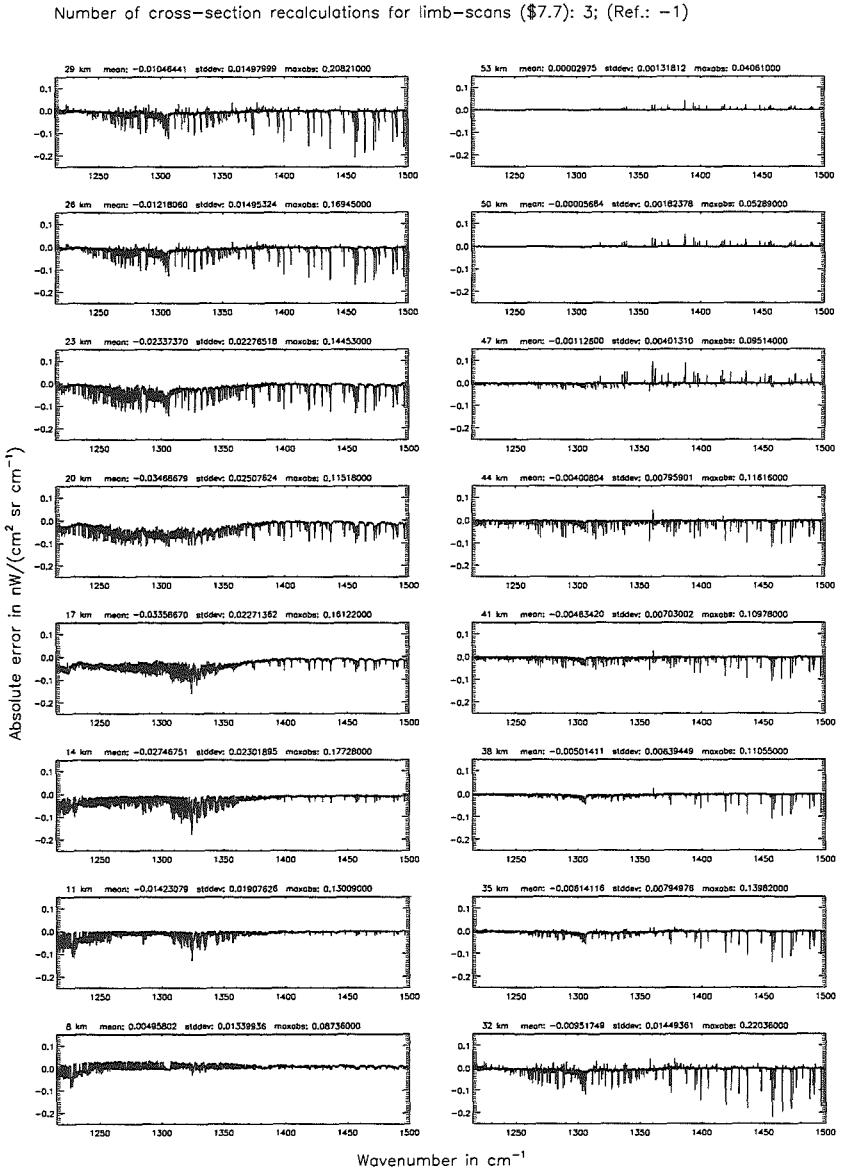


Figure 154: Relative error [%] with respect to MIPAS NESR



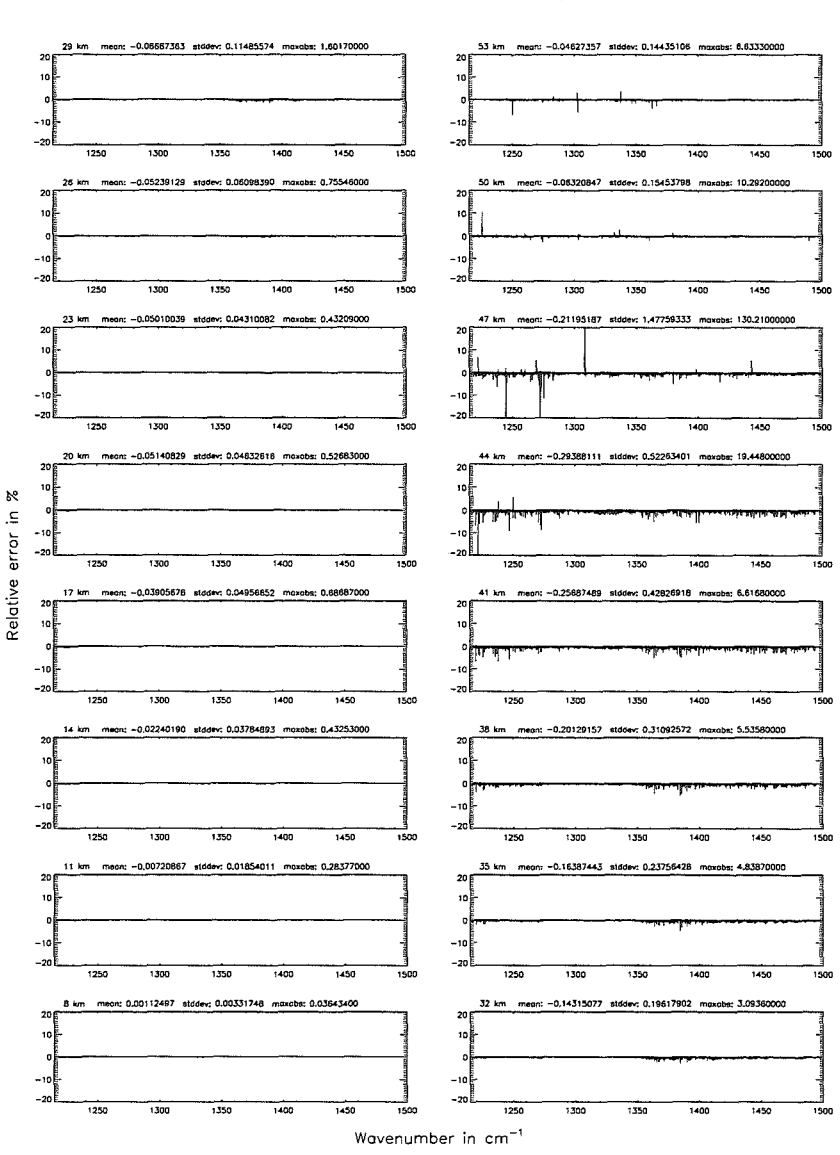


Figure 155: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

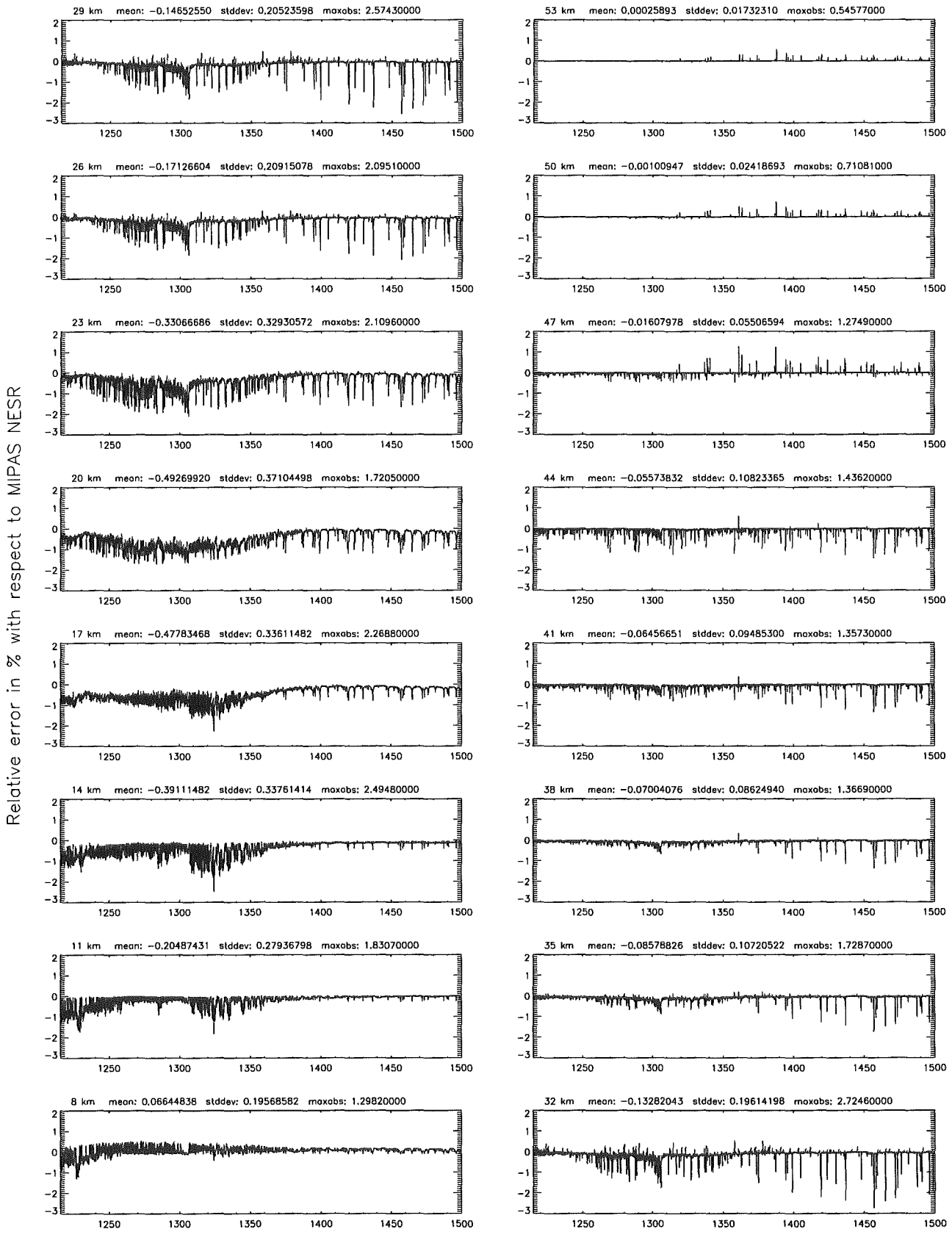
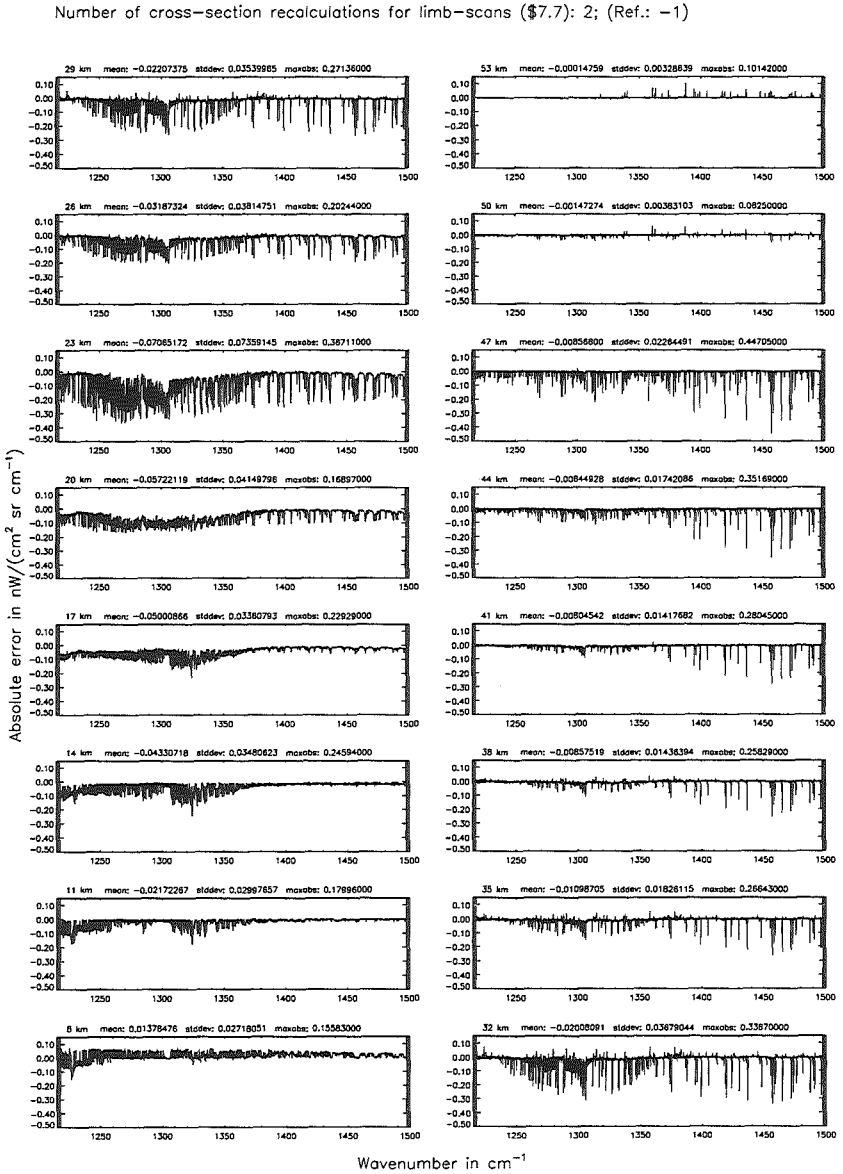


Figure 156: Relative error [%] with respect to MIPAS NESR



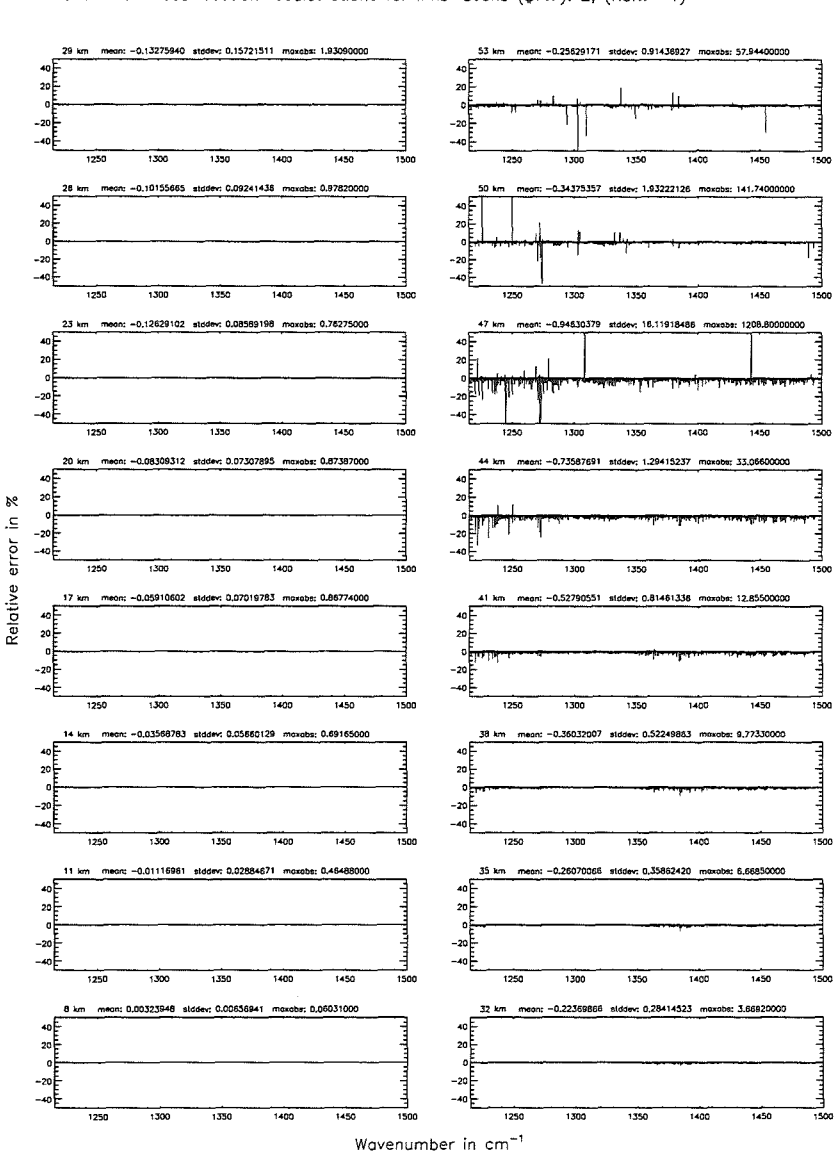


Figure 157: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

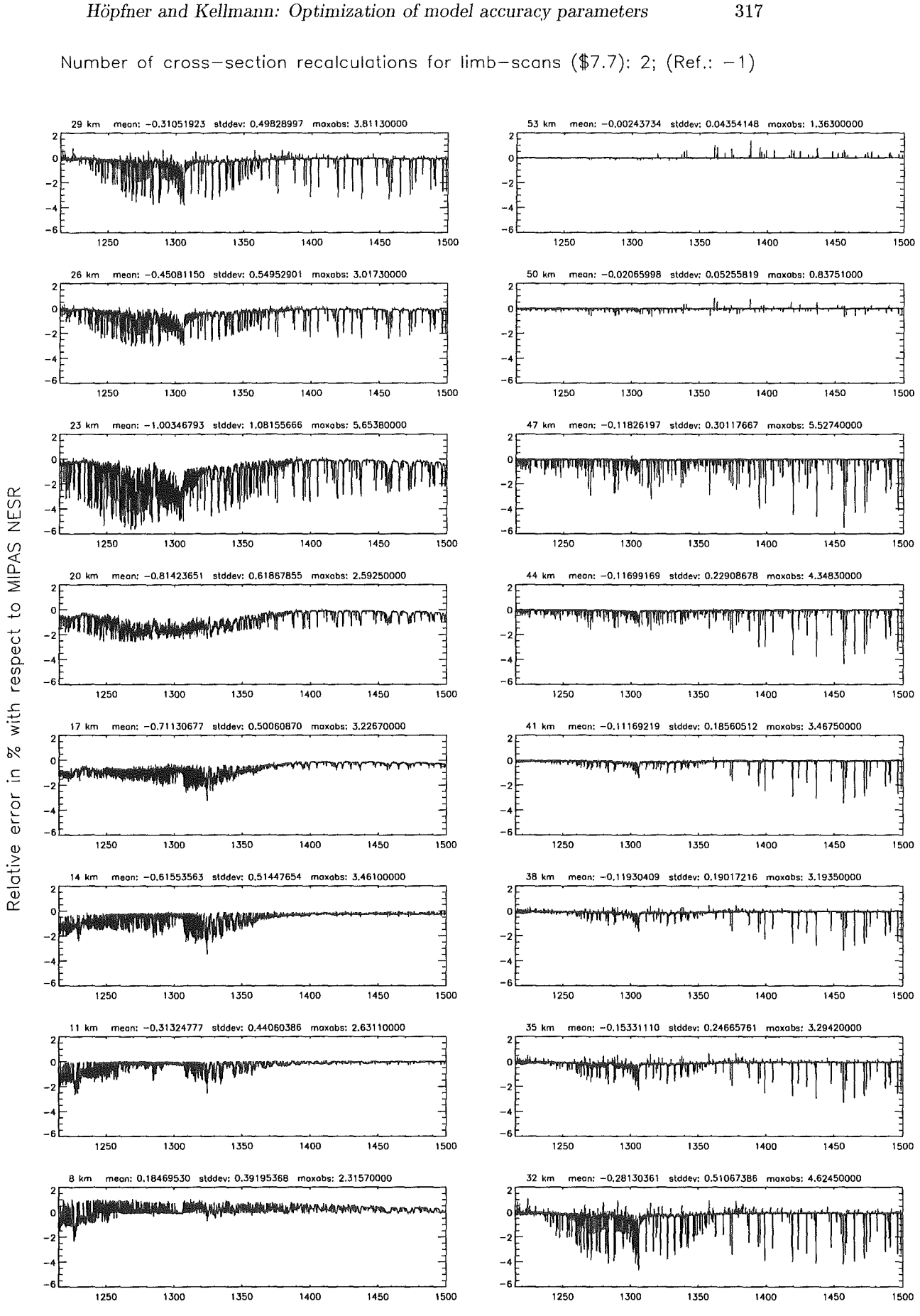
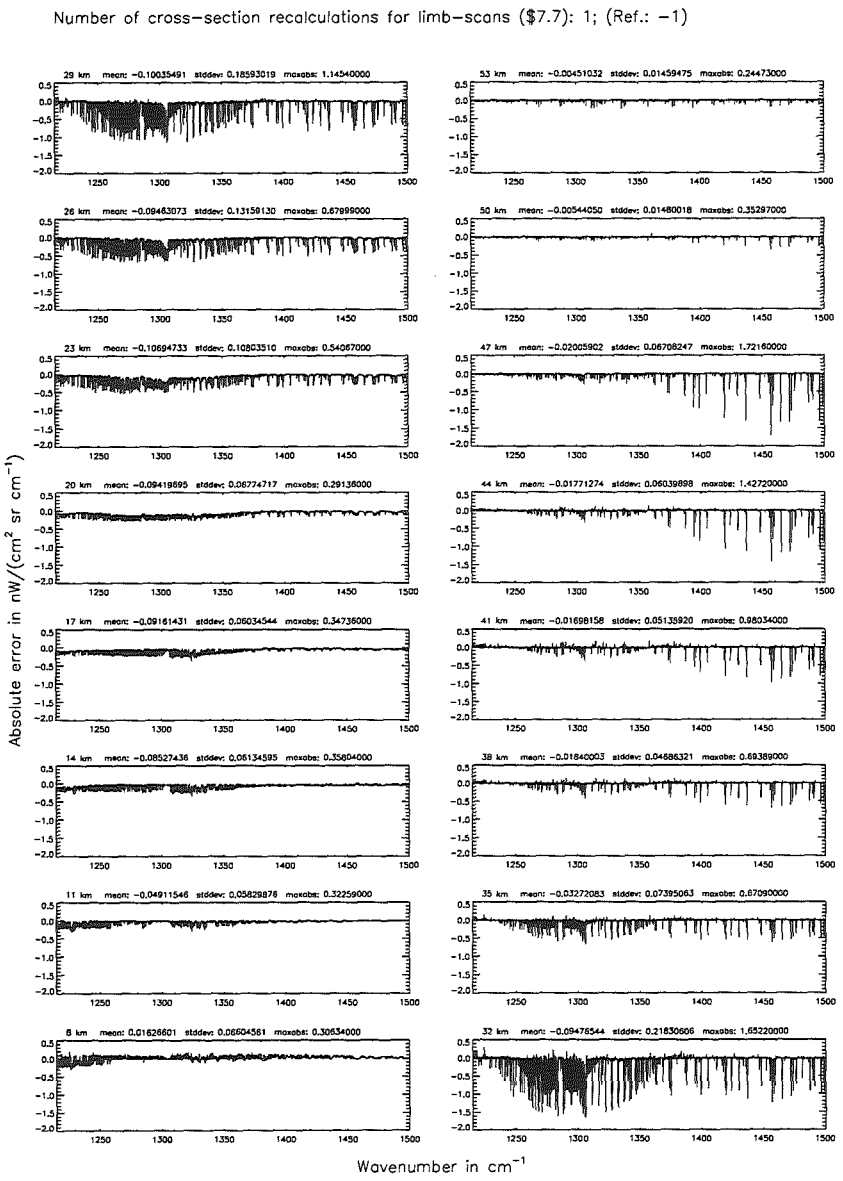


Figure 158: Relative error [%] with respect to MIPAS NESR



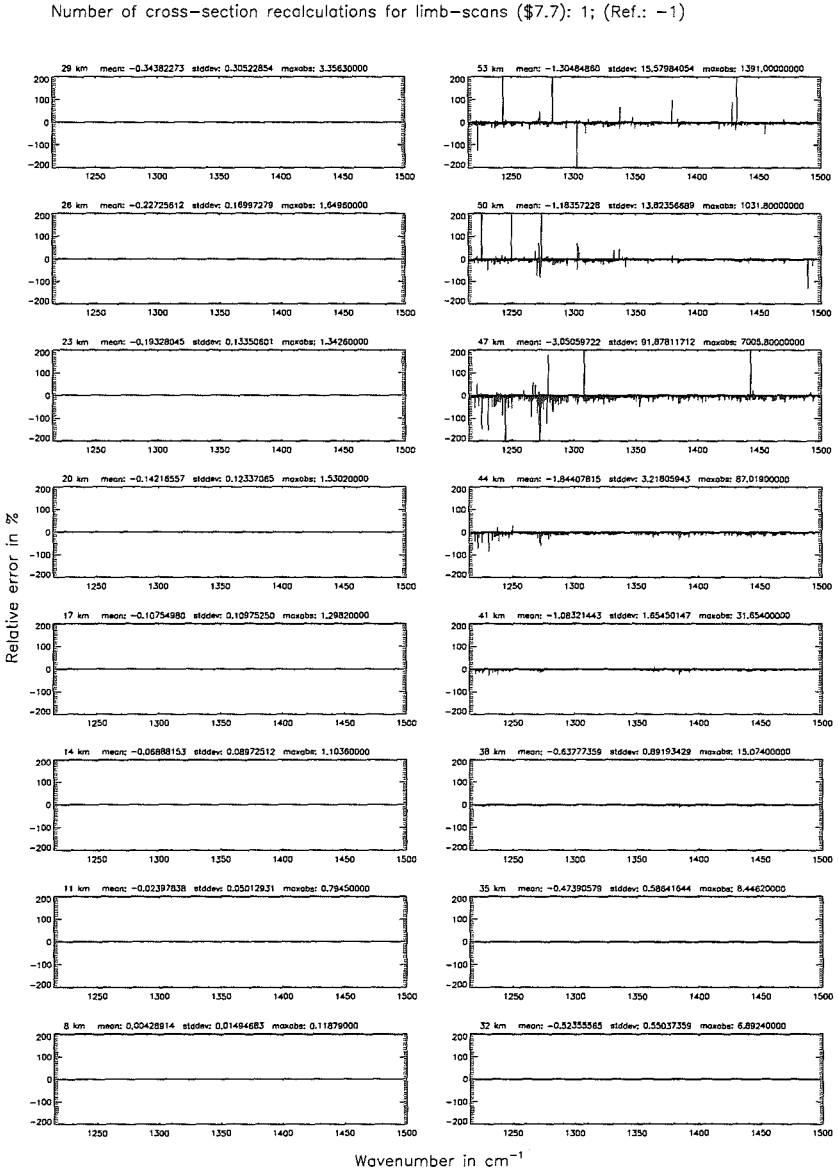


Figure 159: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

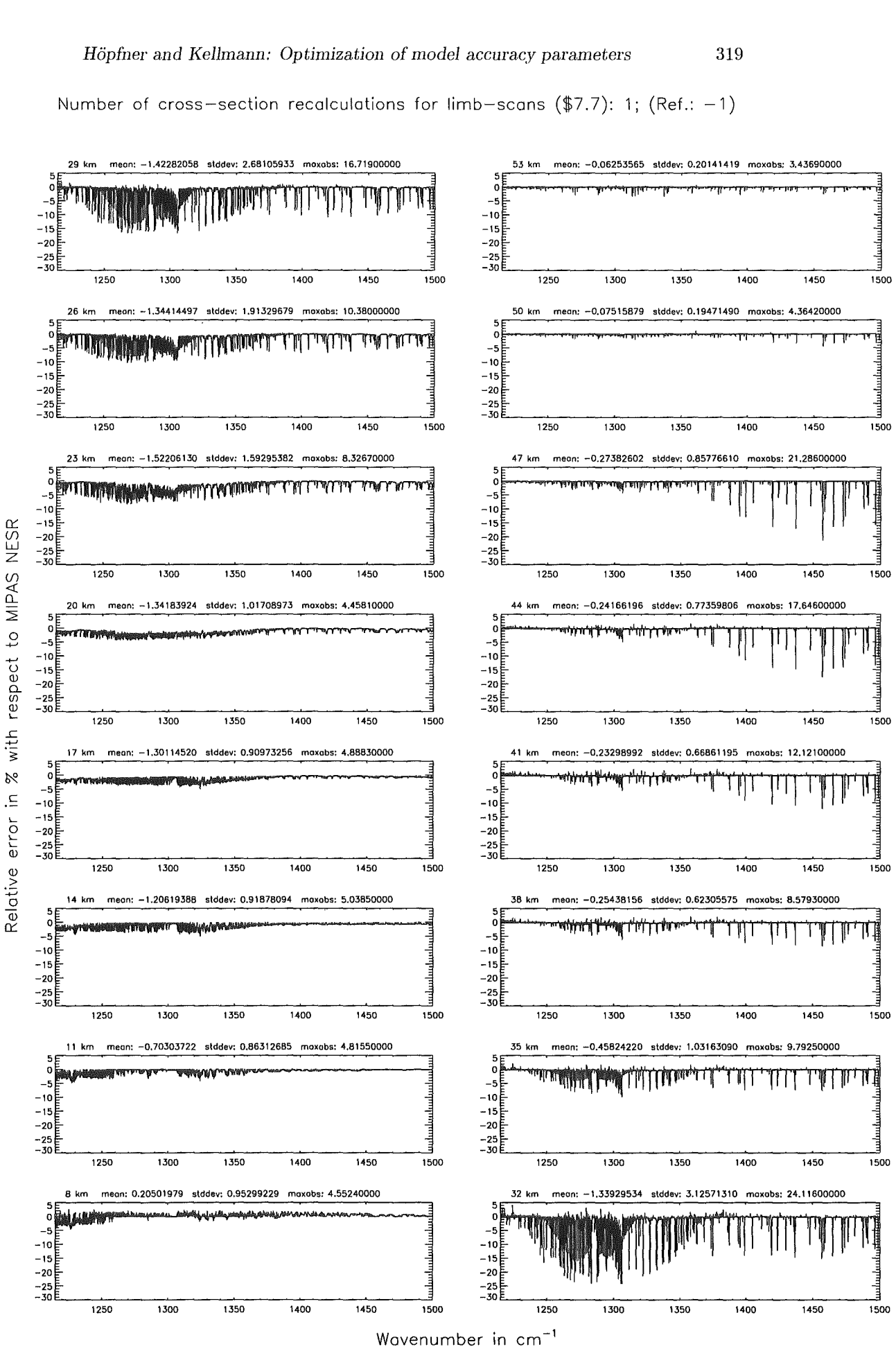
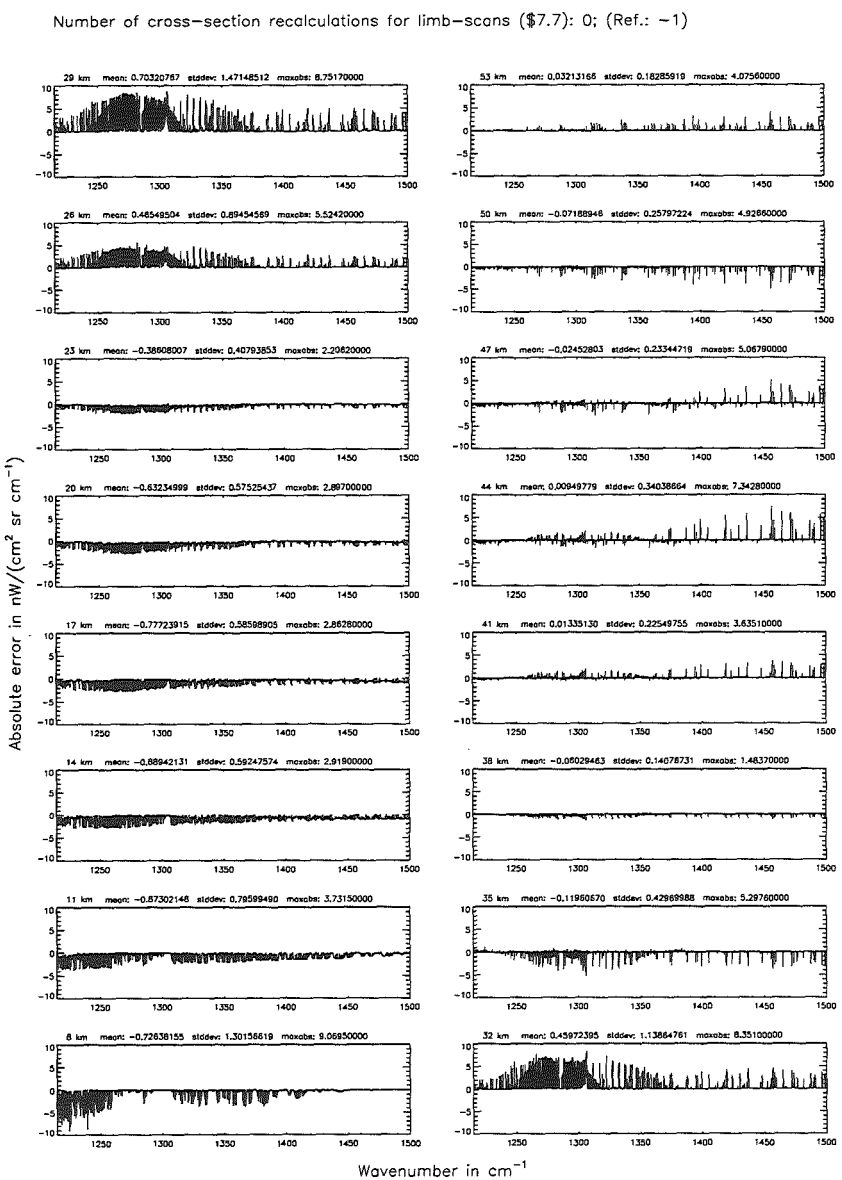


Figure 160: Relative error [%] with respect to MIPAS NESR



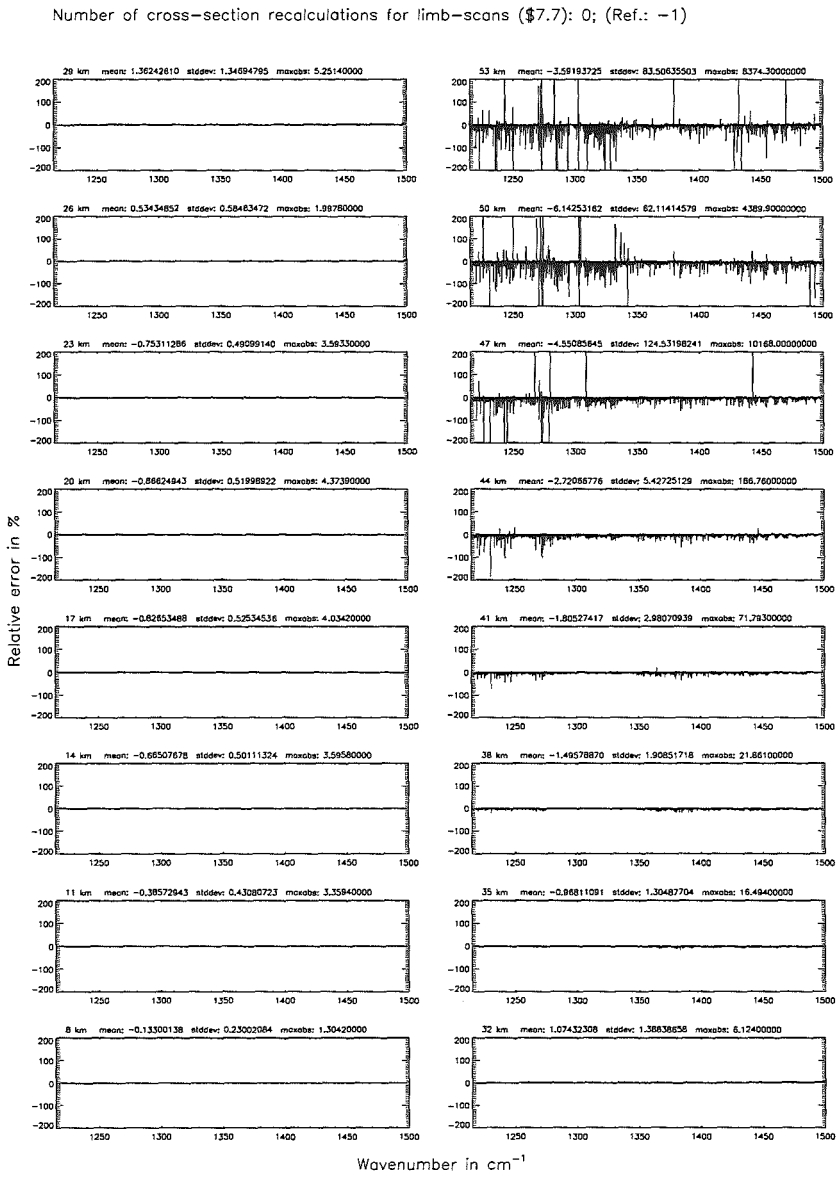


Figure 161: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

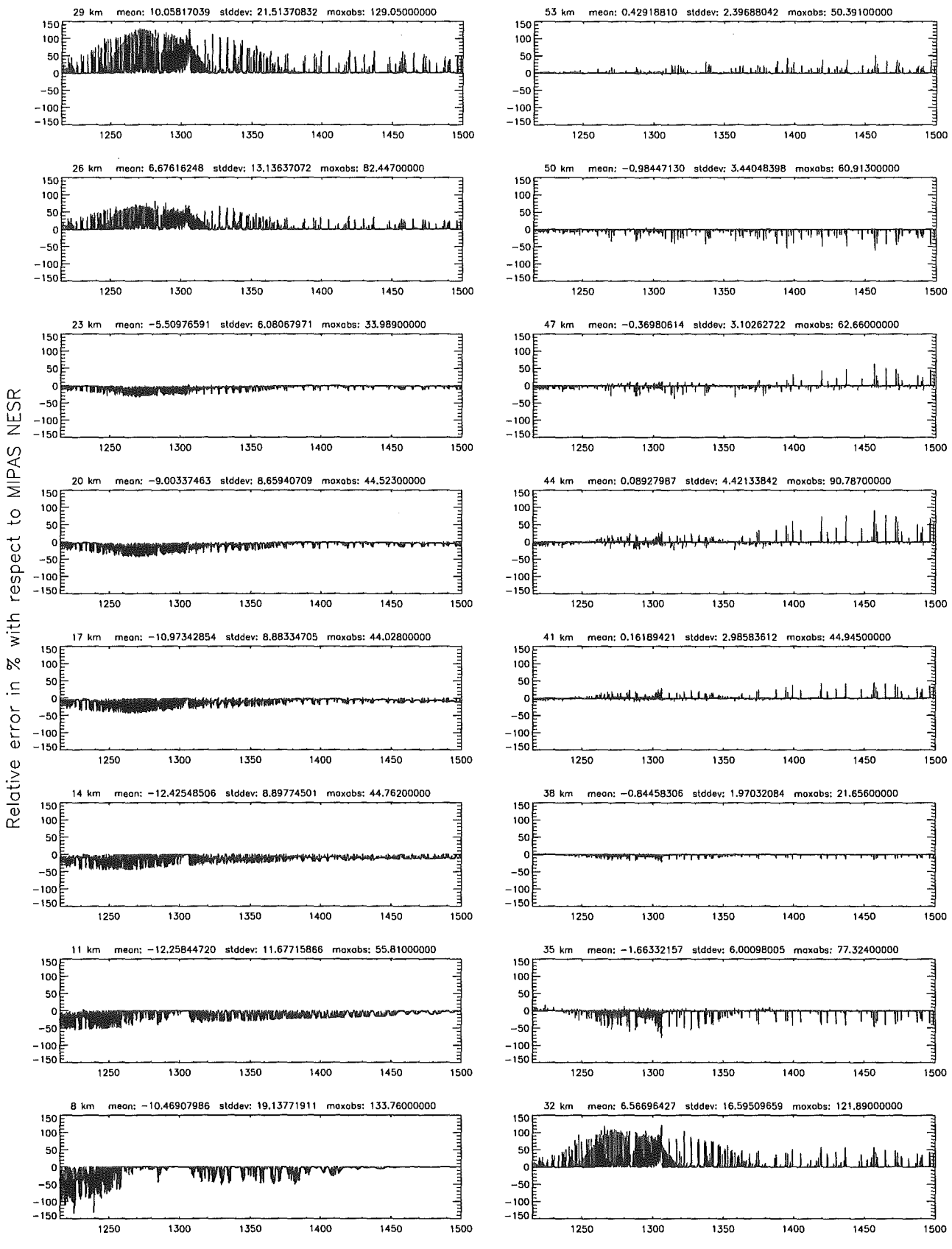


Figure 162: Relative error [%] with respect to MIPAS NESR

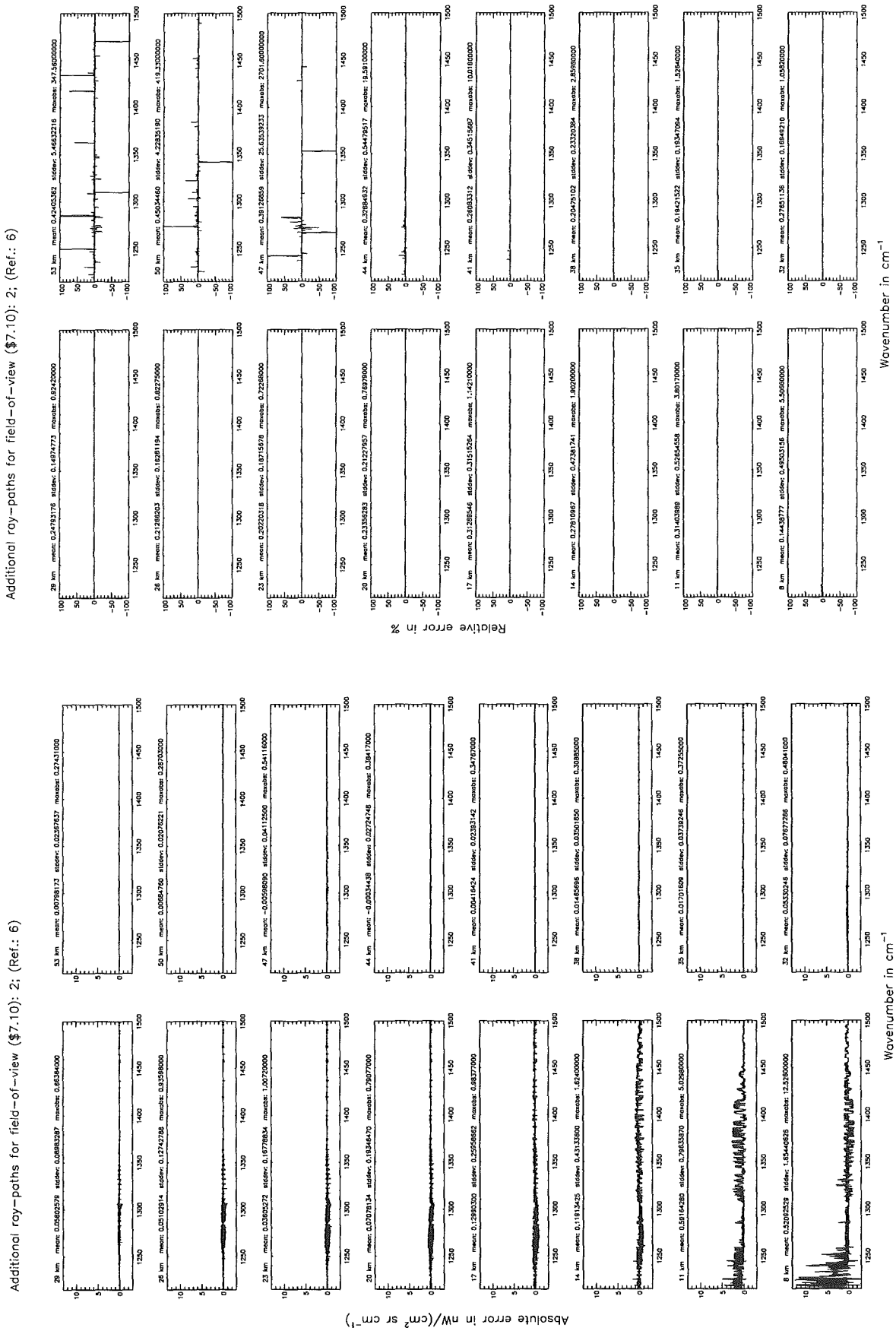


Figure 163: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

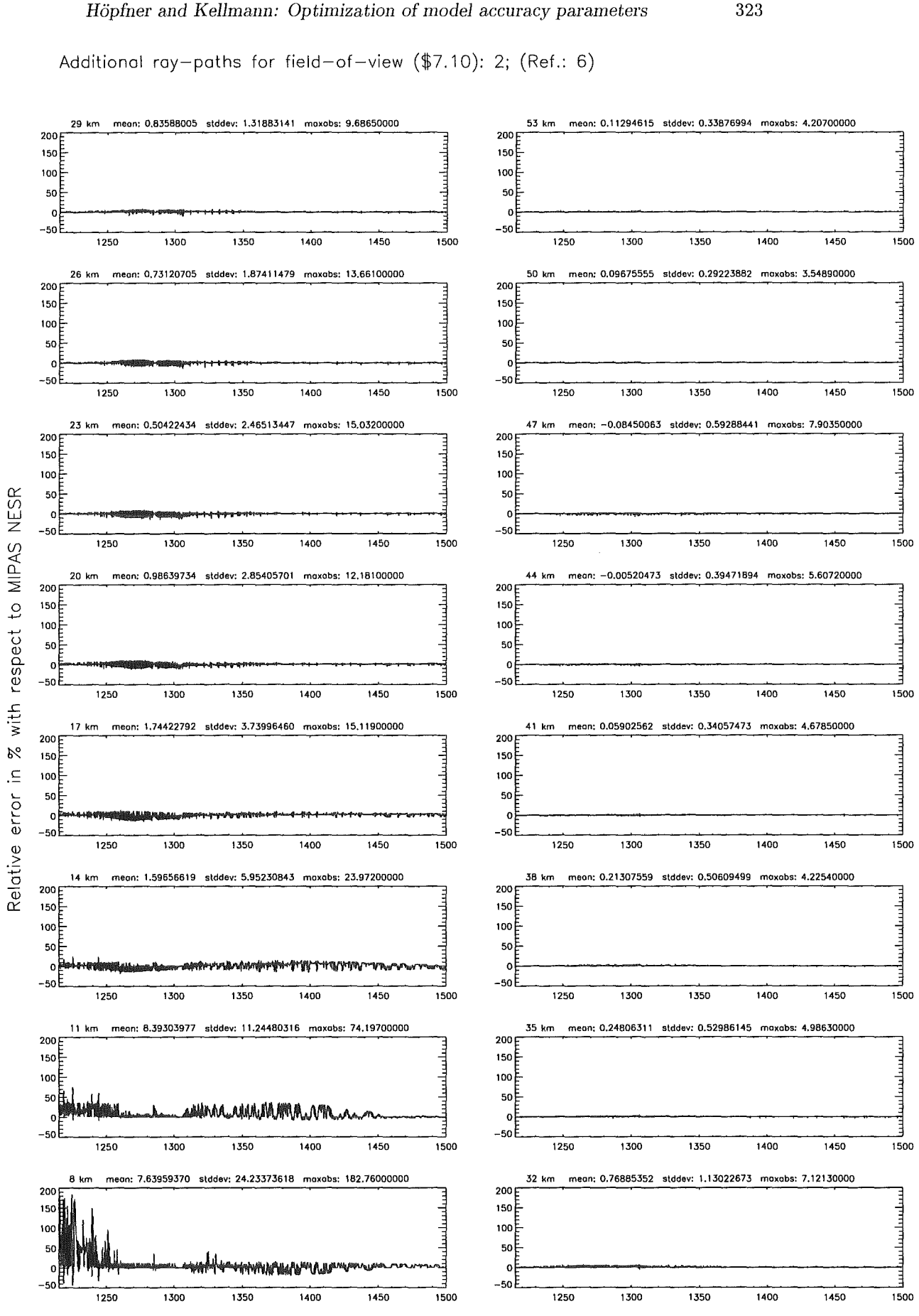


Figure 164: Relative error [%] with respect to MIPAS NESR

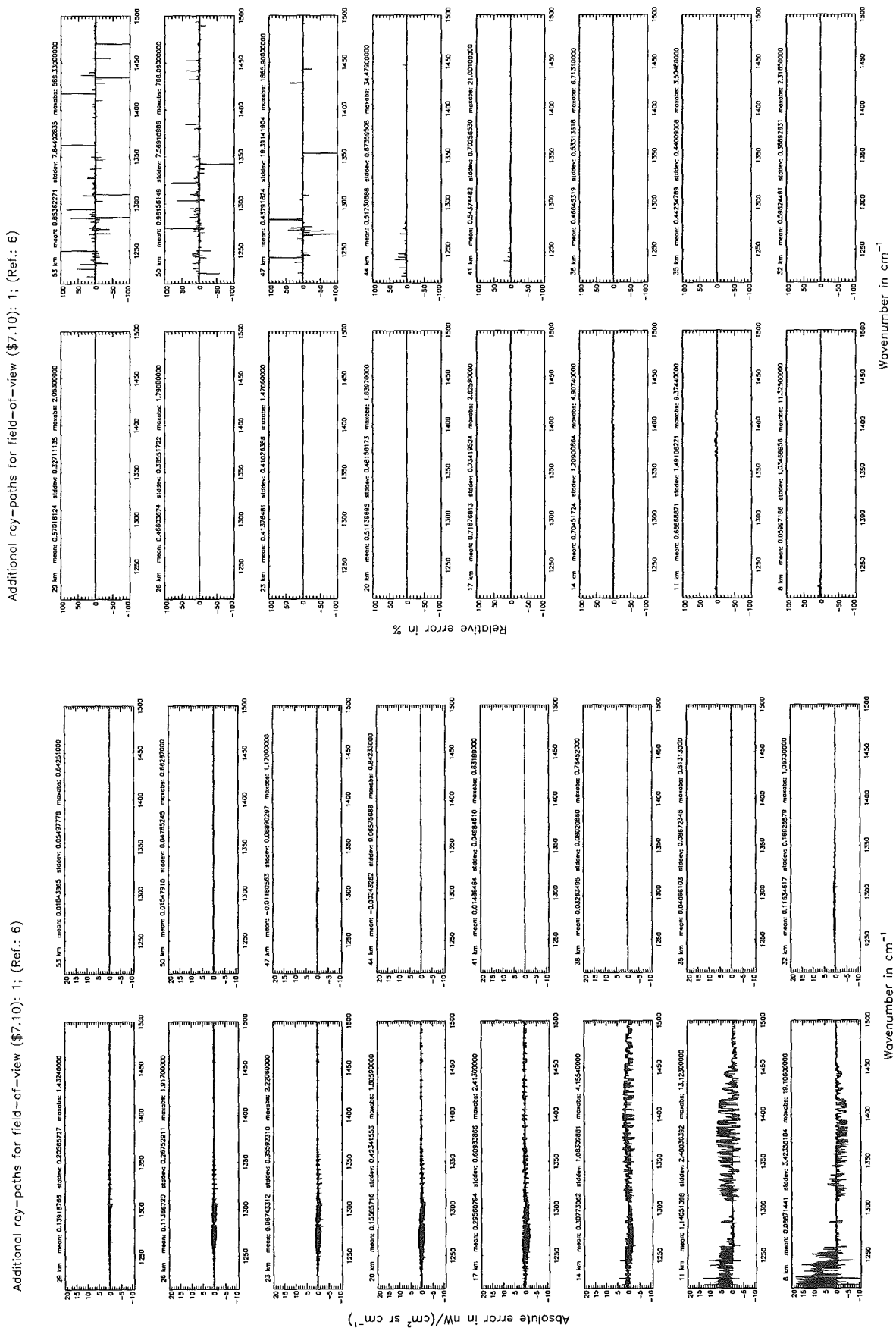


Figure 165: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

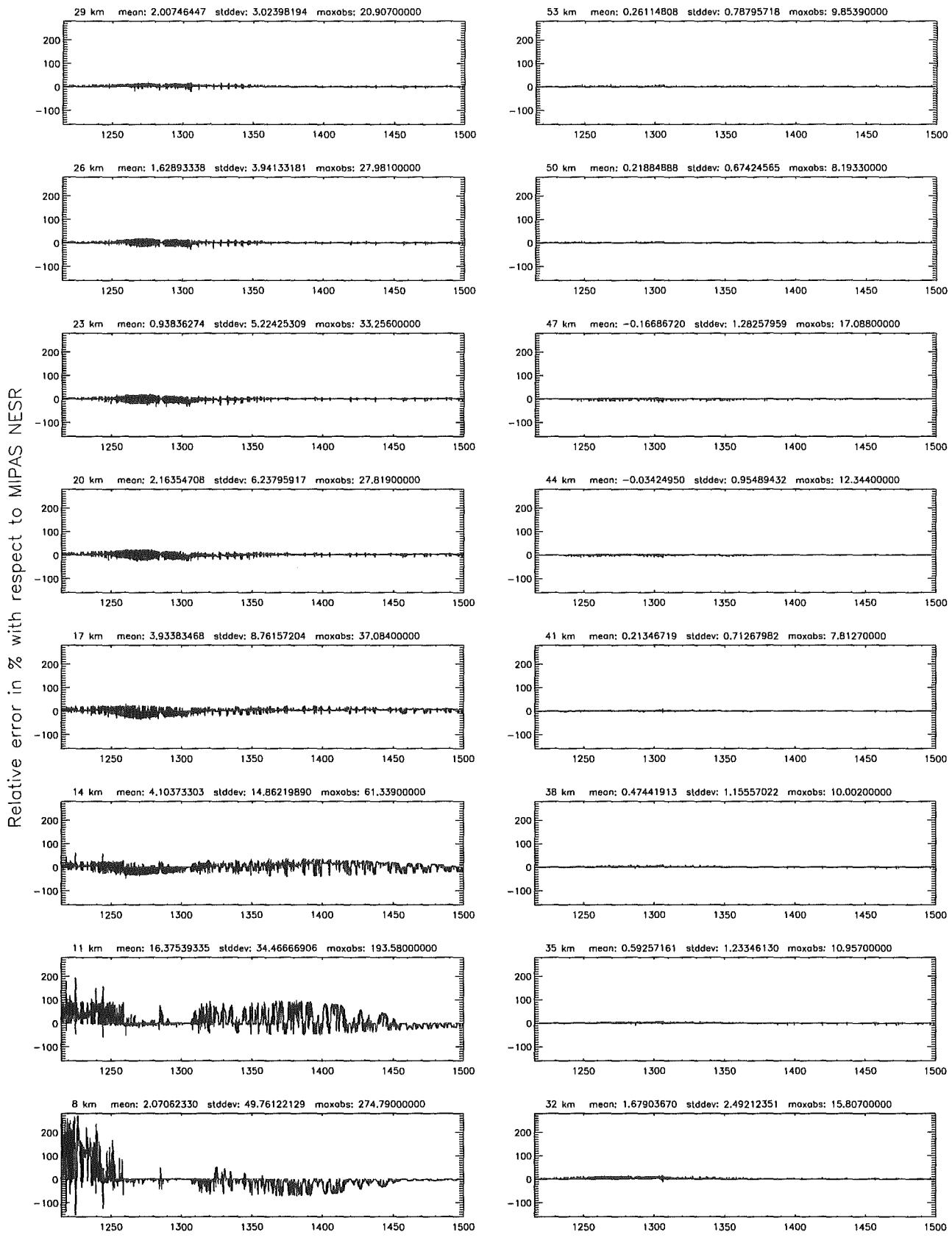
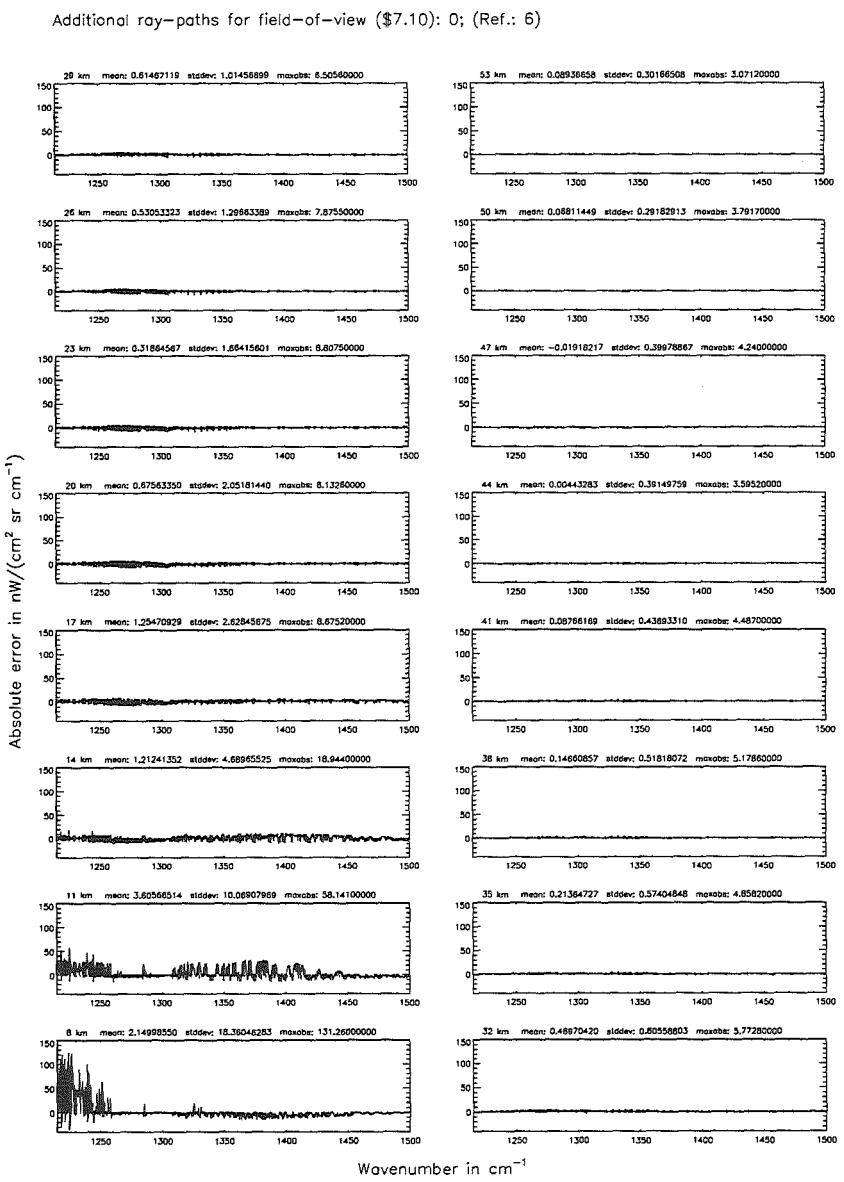


Figure 166: Relative error [%] with respect to MIPAS NESR



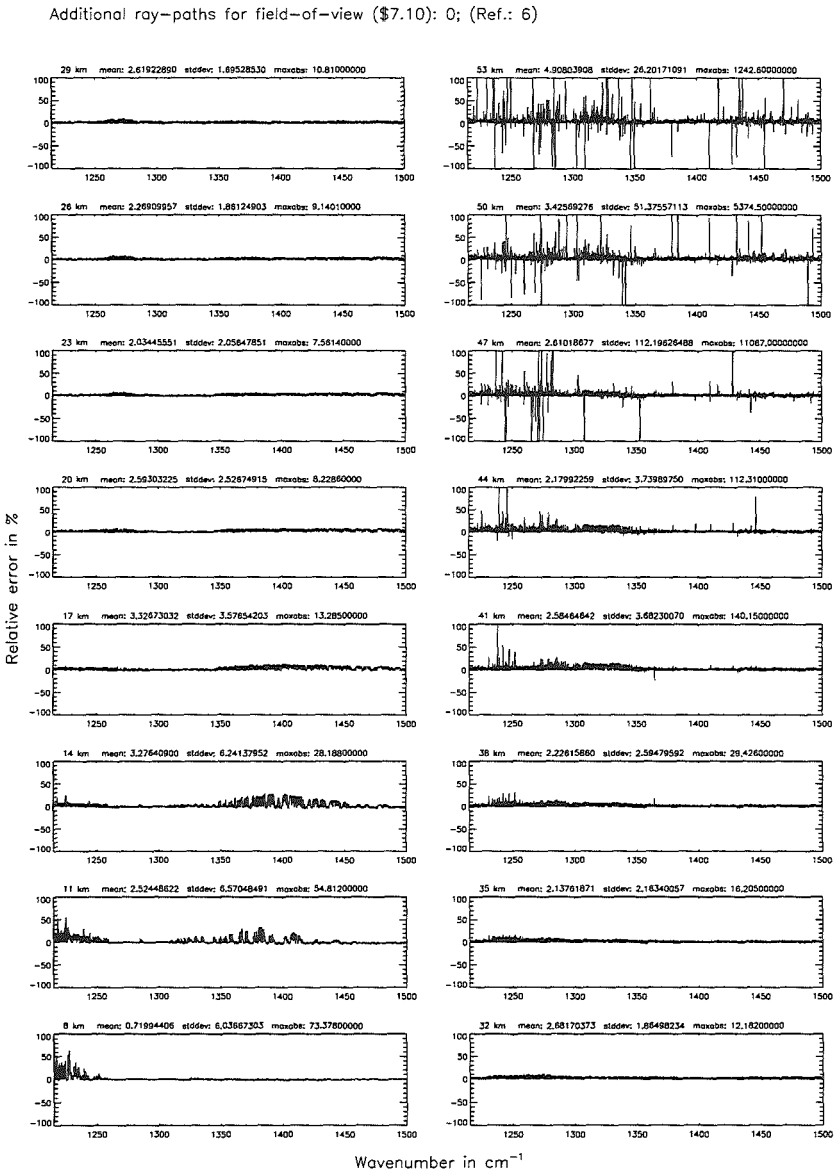


Figure 167: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

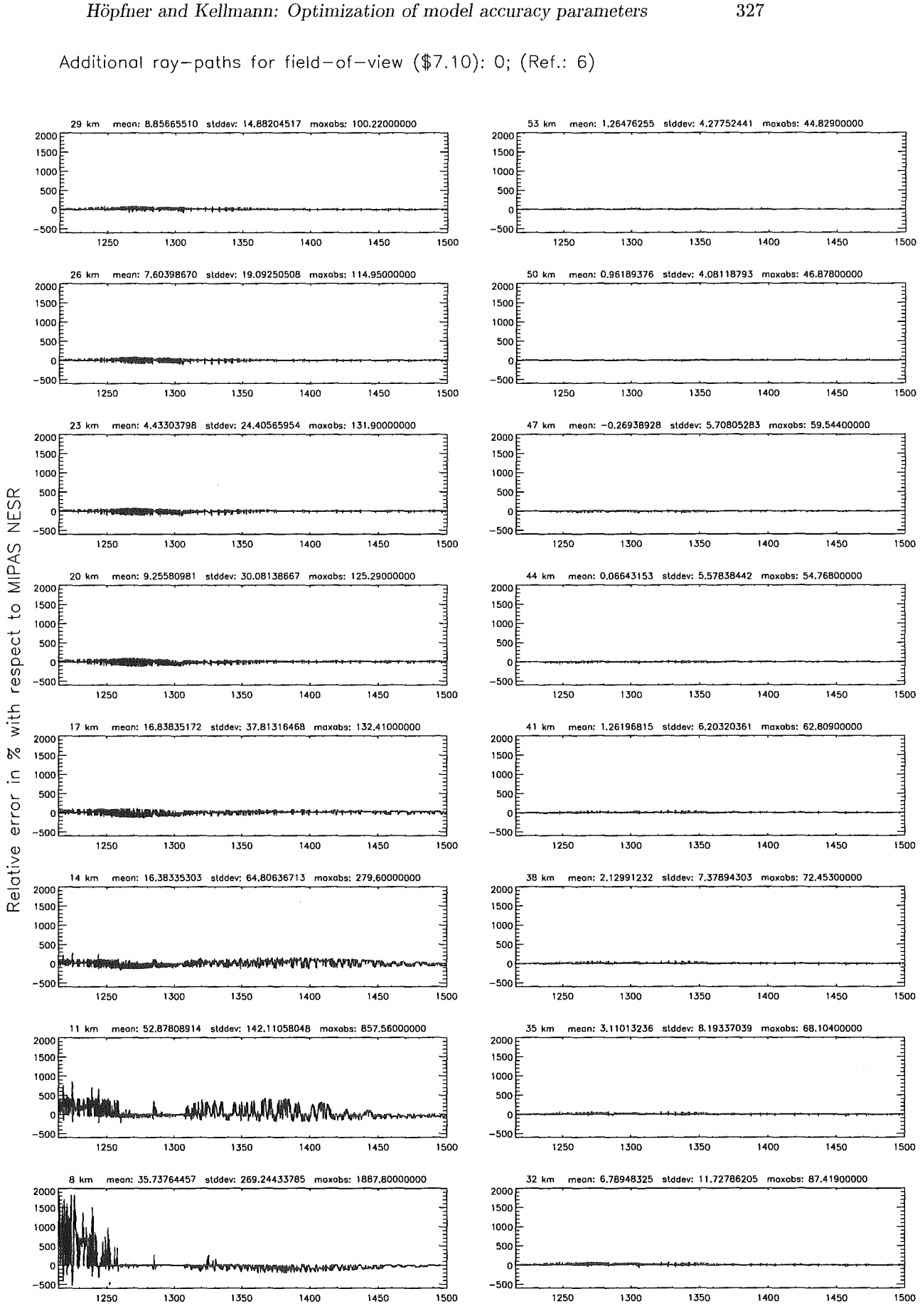
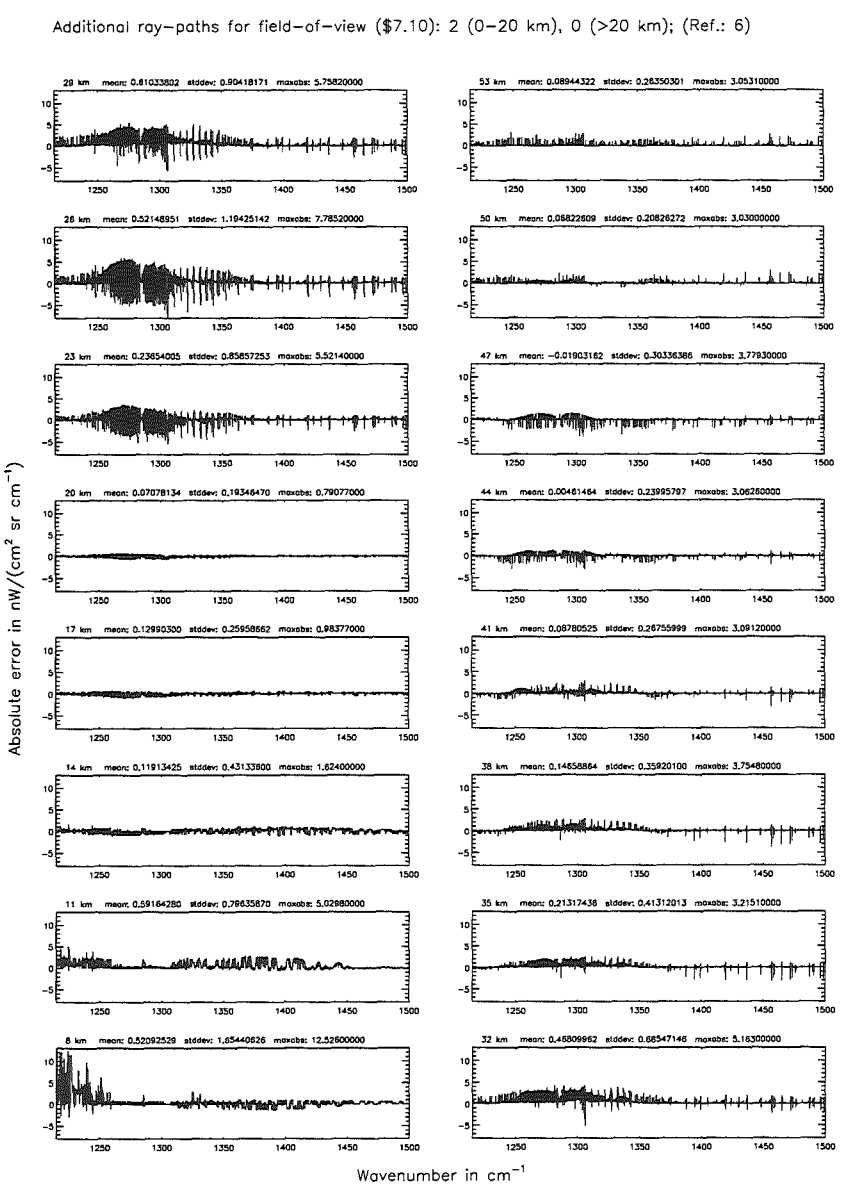


Figure 168: Relative error [%] with respect to MIPAS NESR



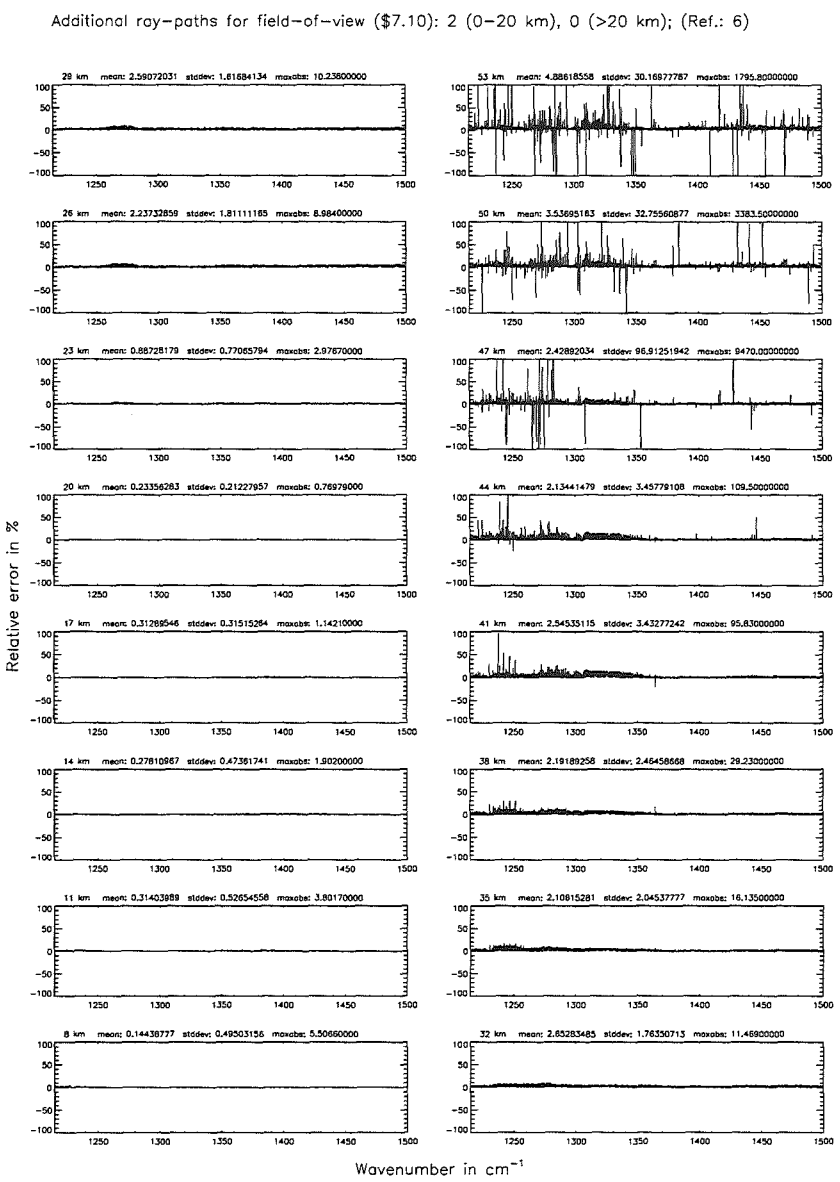


Figure 169: Absolute error [nW/(cm² sr cm⁻¹)] and relative error [%]

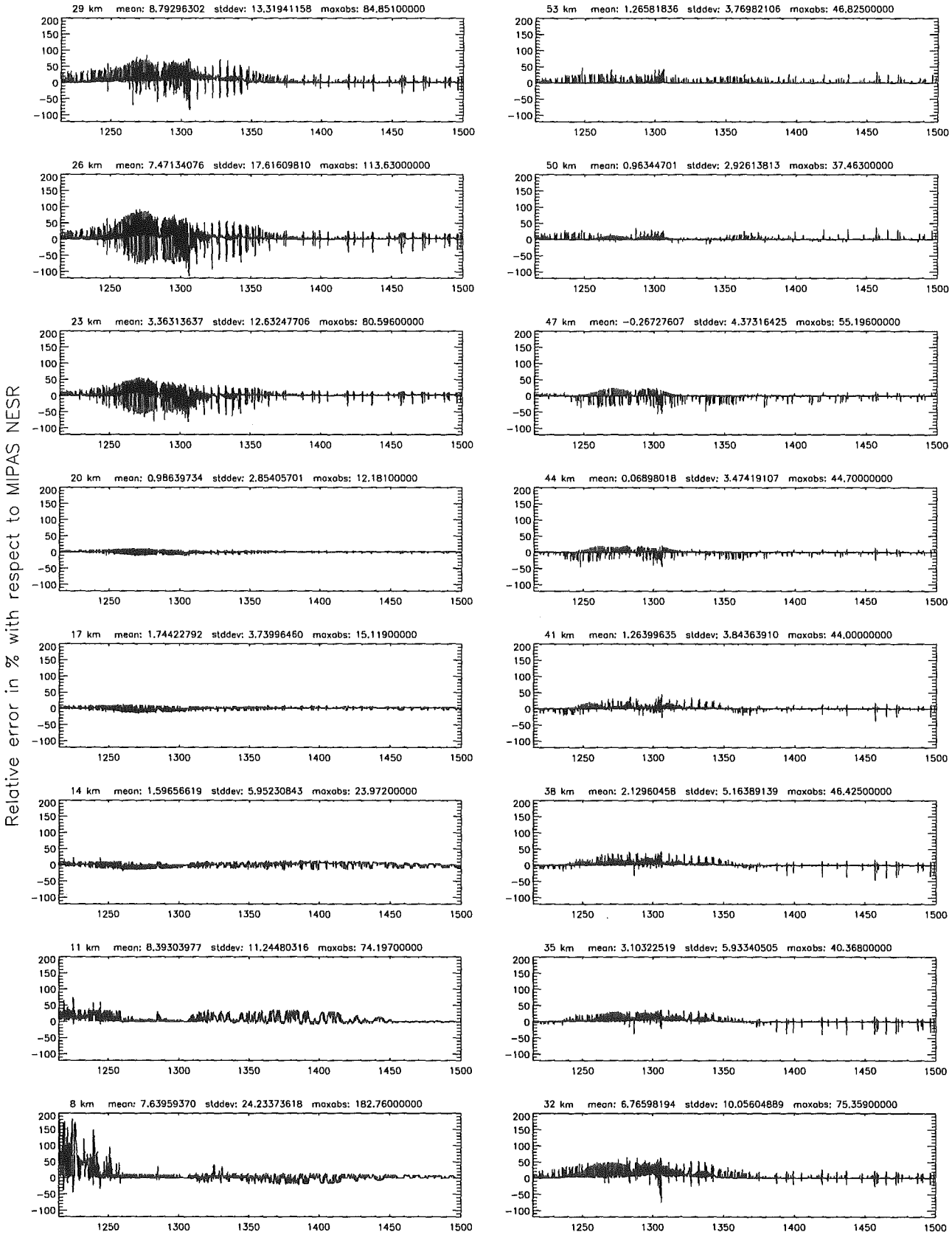
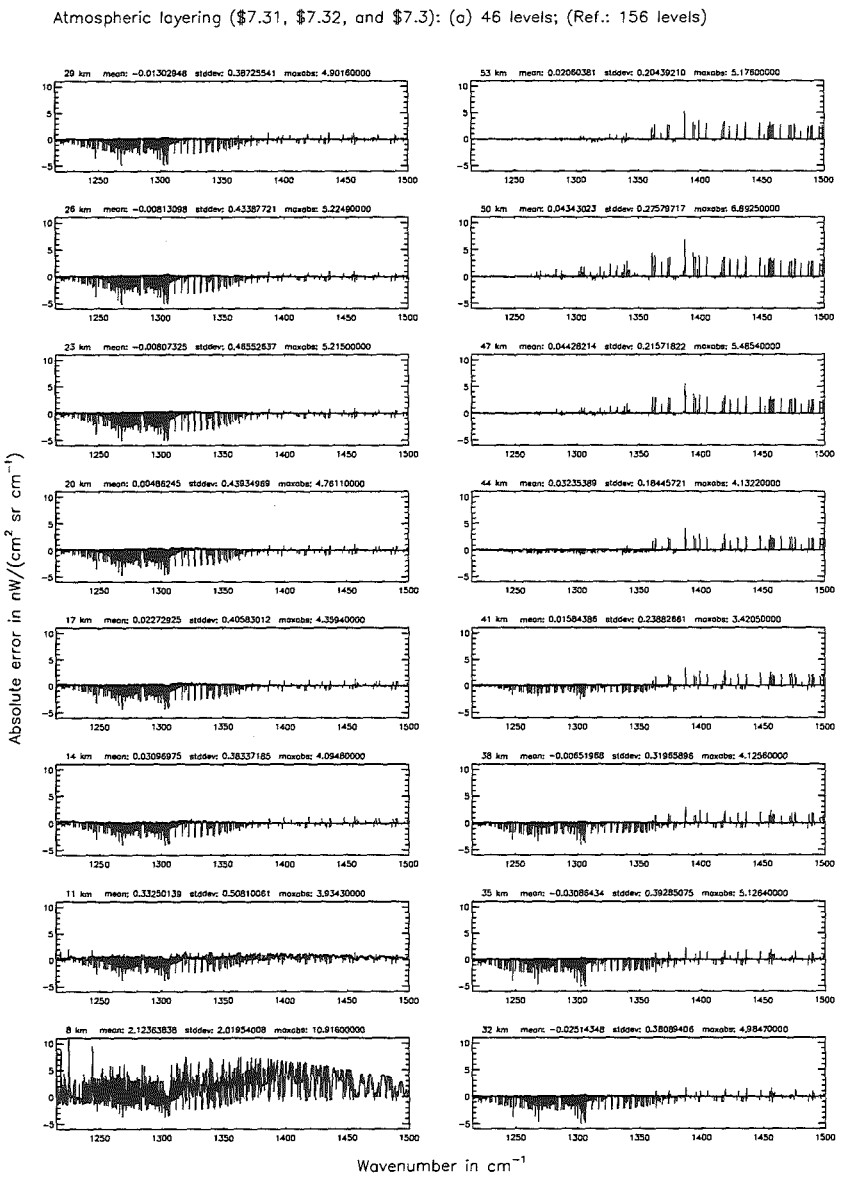


Figure 170: Relative error [%] with respect to MIPAS NESR



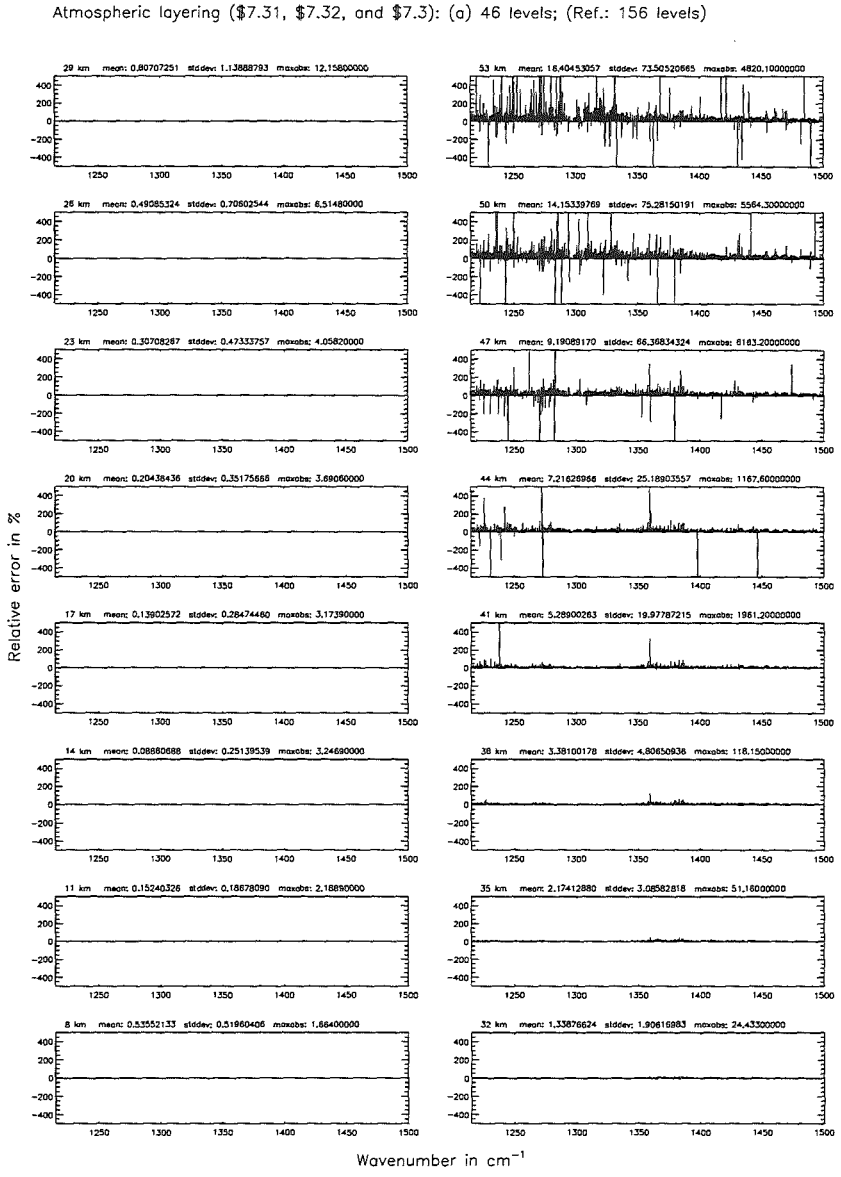


Figure 171: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

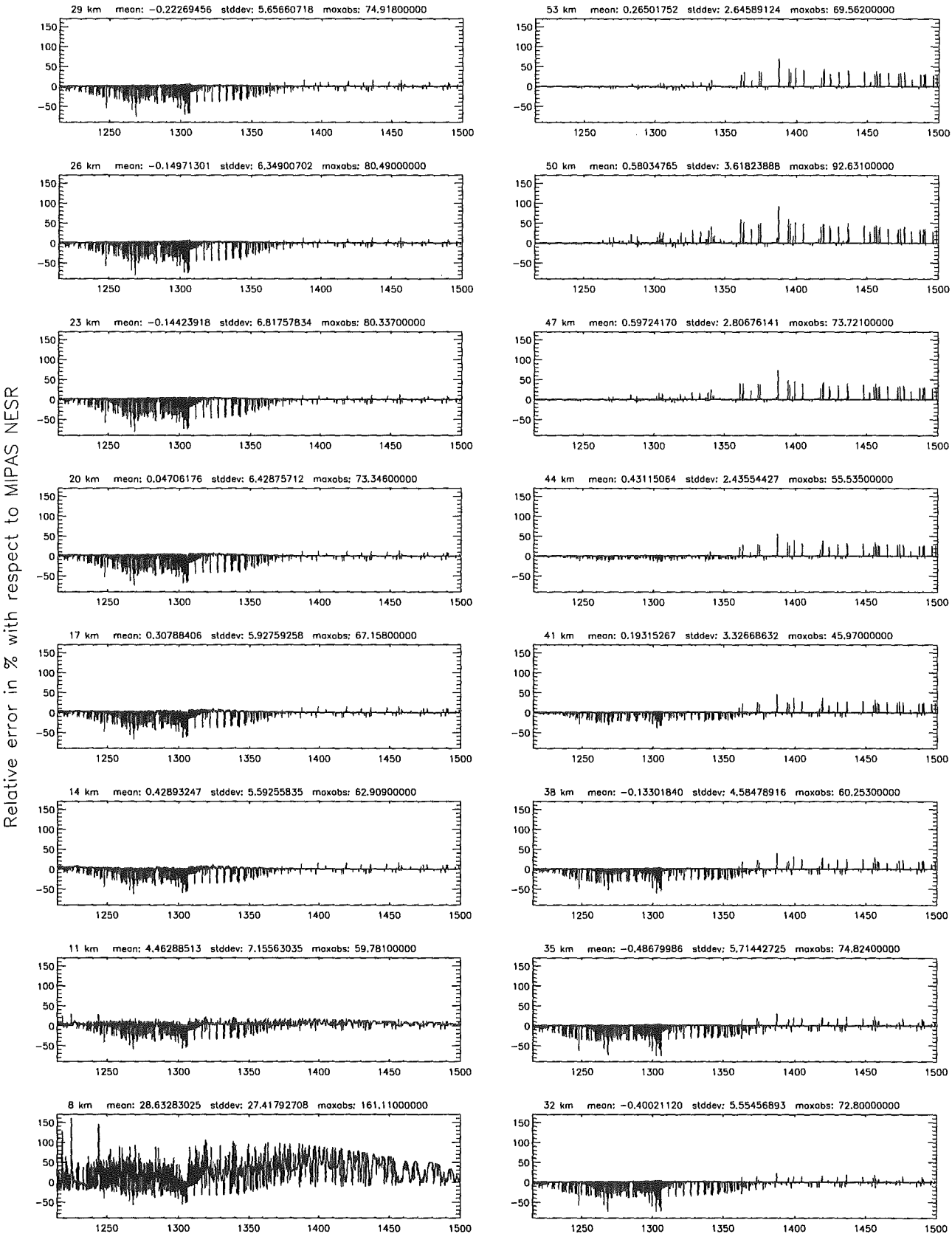
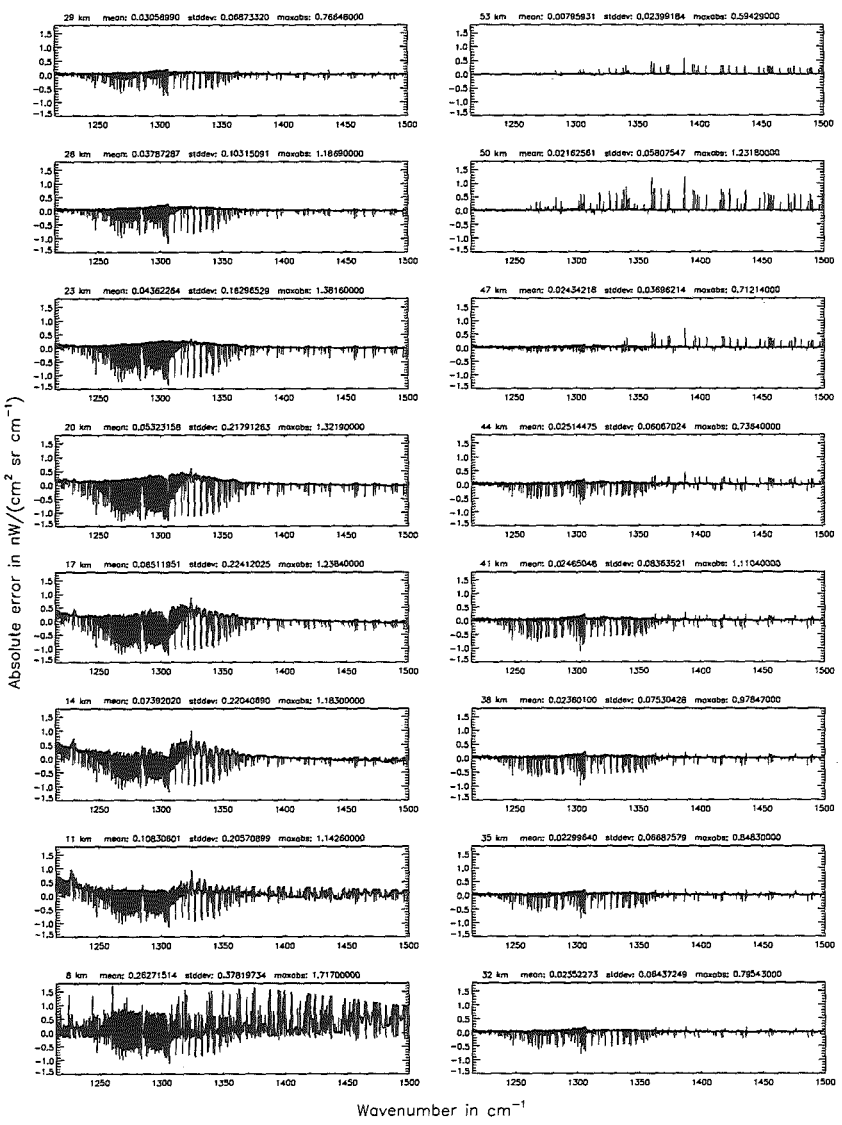


Figure 172: Relative error [%] with respect to MIPAS NESR



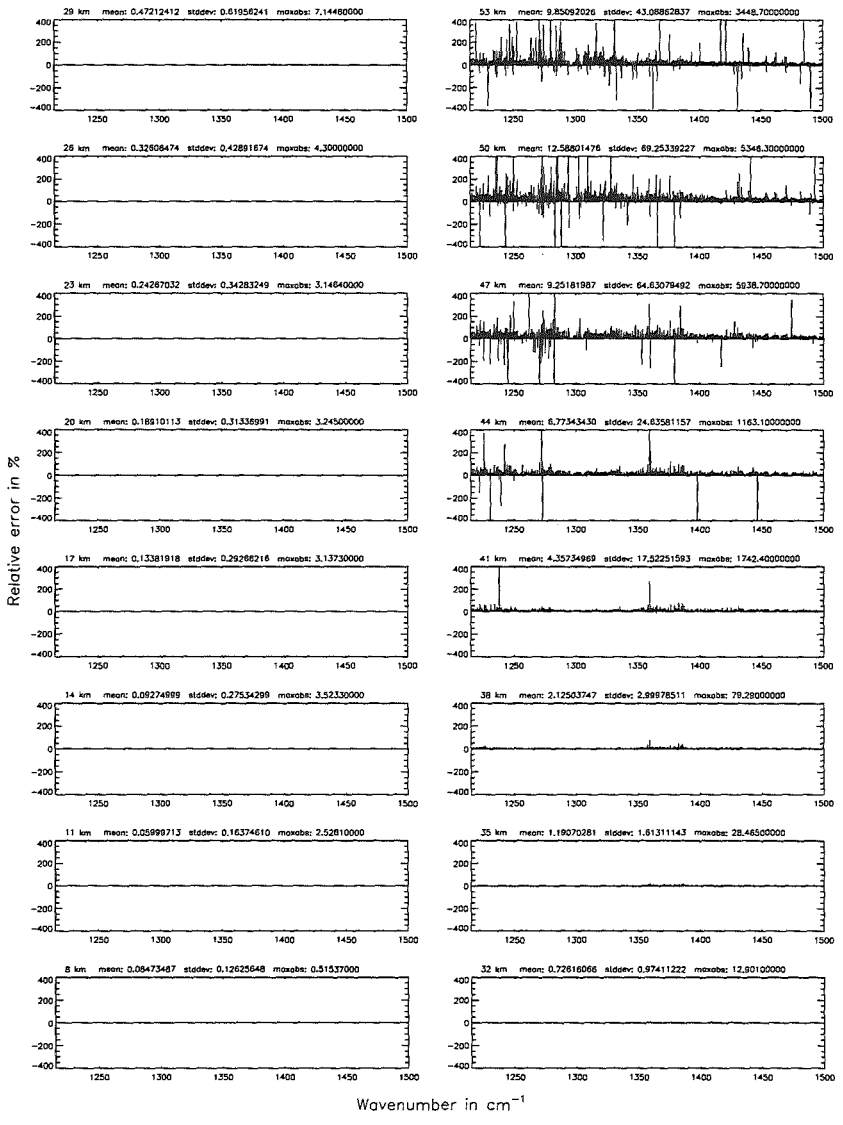


Figure 173: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

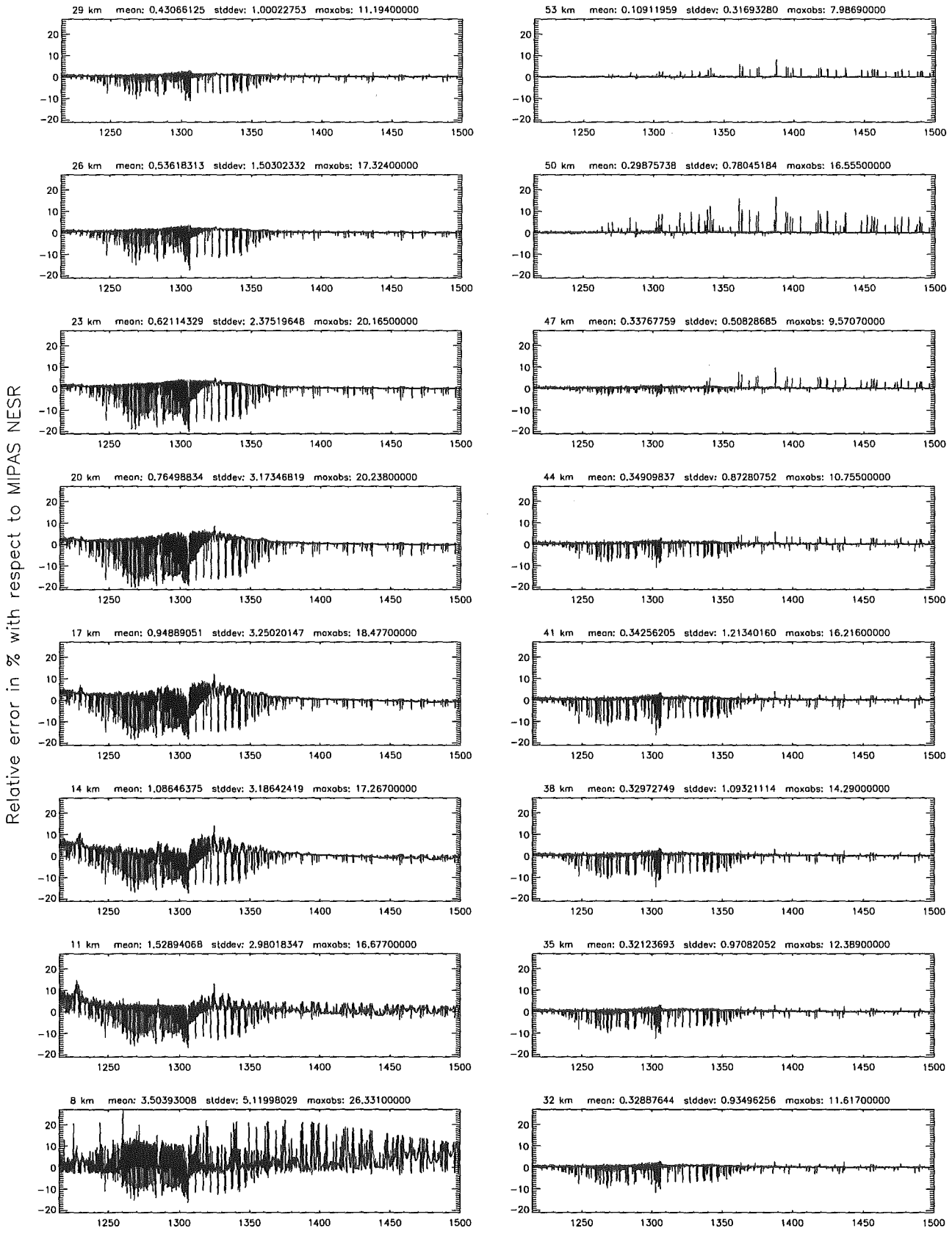
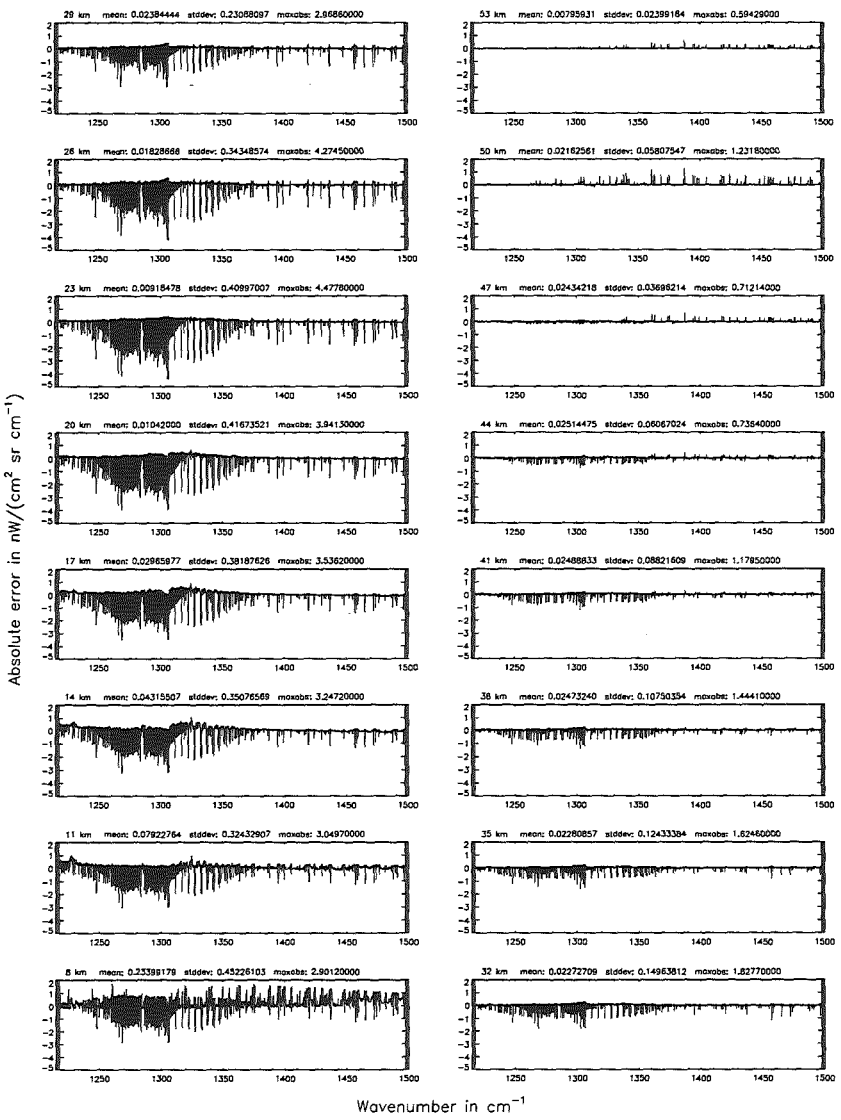


Figure 174: Relative error [%] with respect to MIPAS NESR



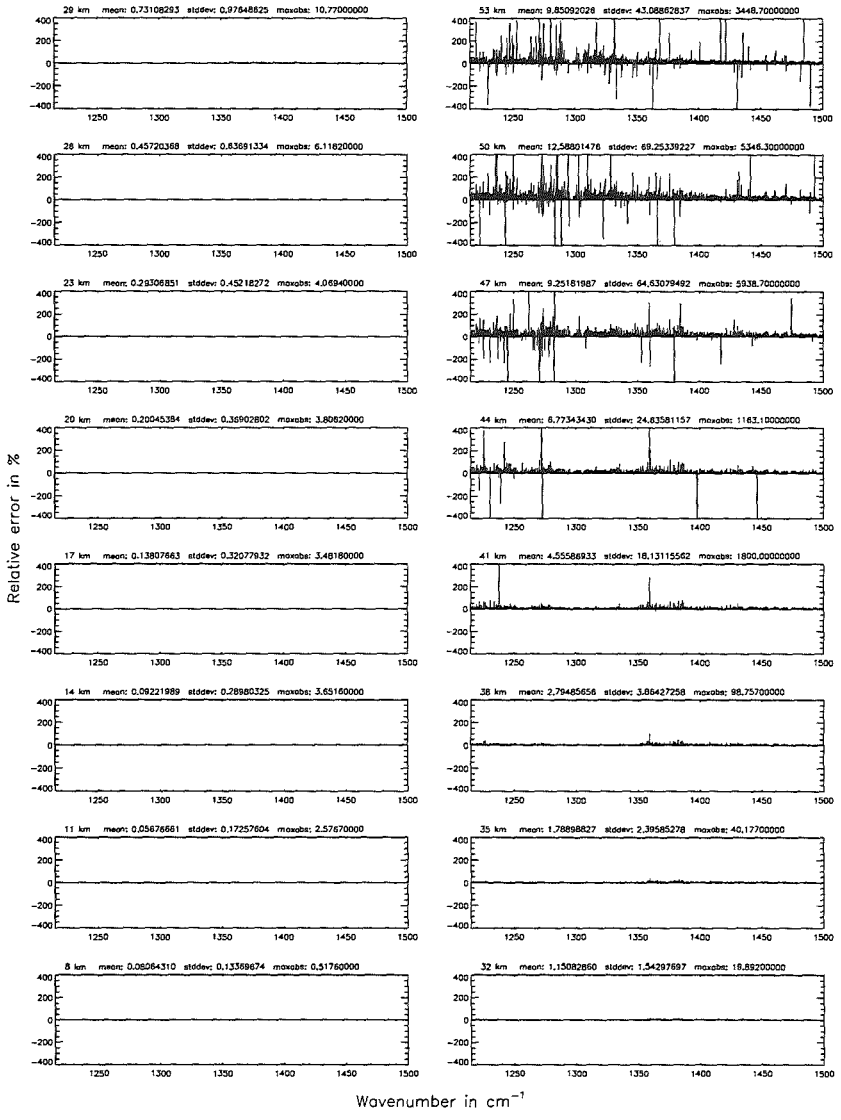


Figure 175: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (c) 66 levels; (Ref.: 156 levels)

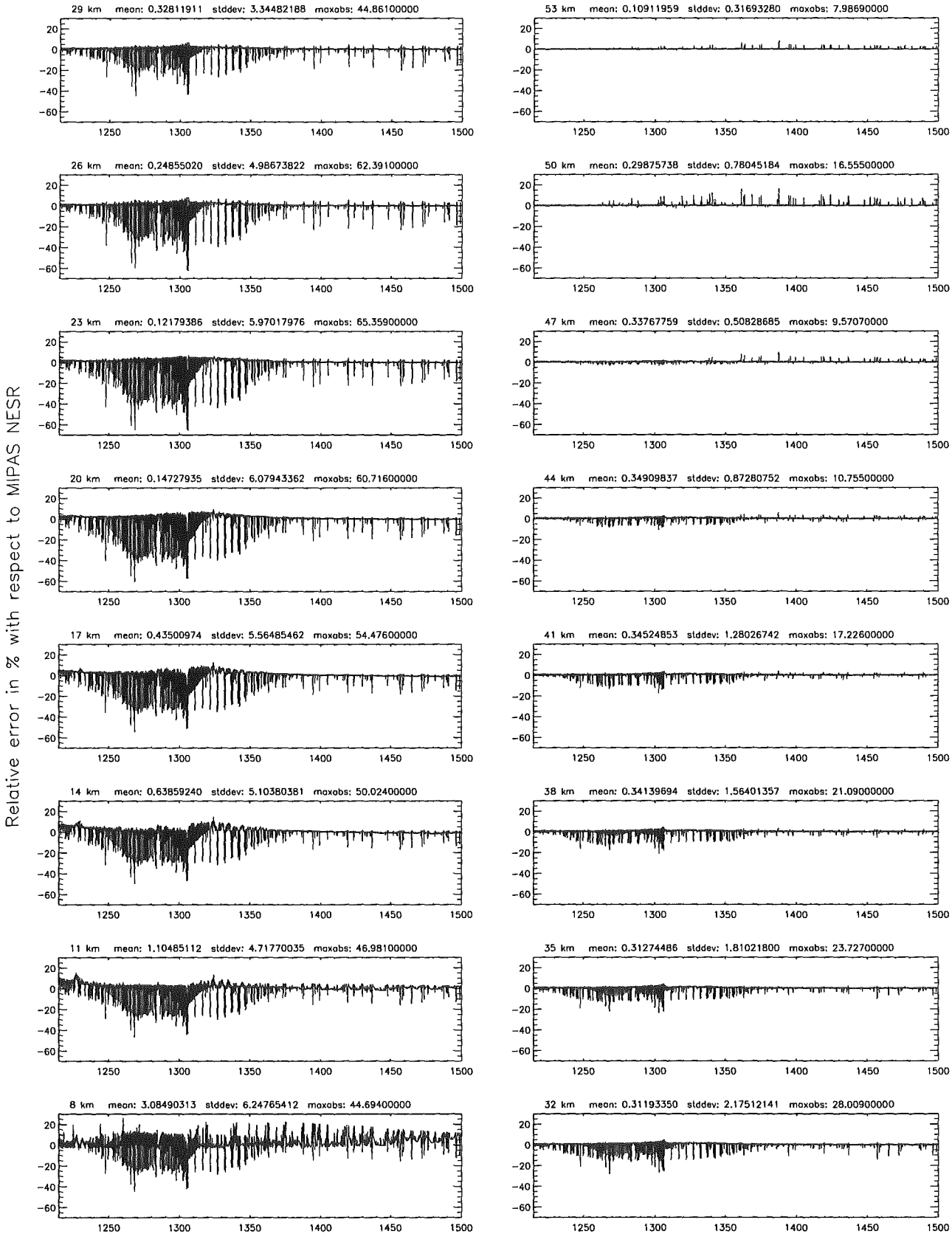
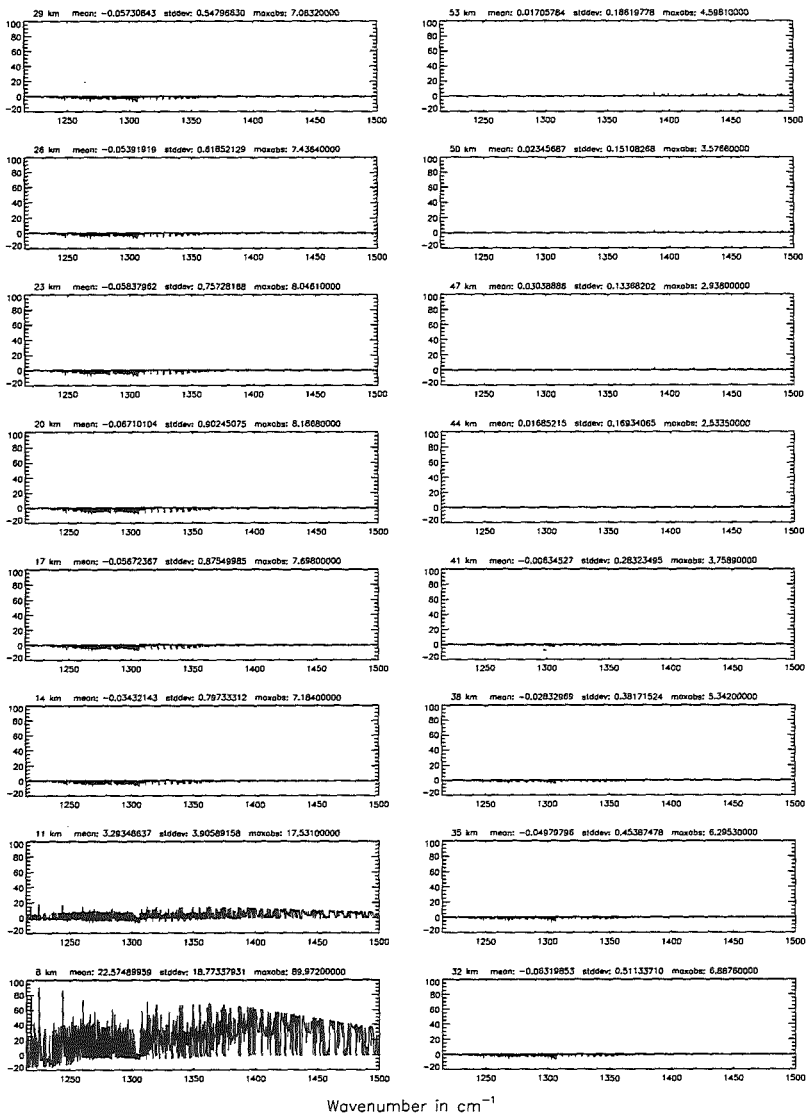


Figure 176: Relative error [%] with respect to MIPAS NESR



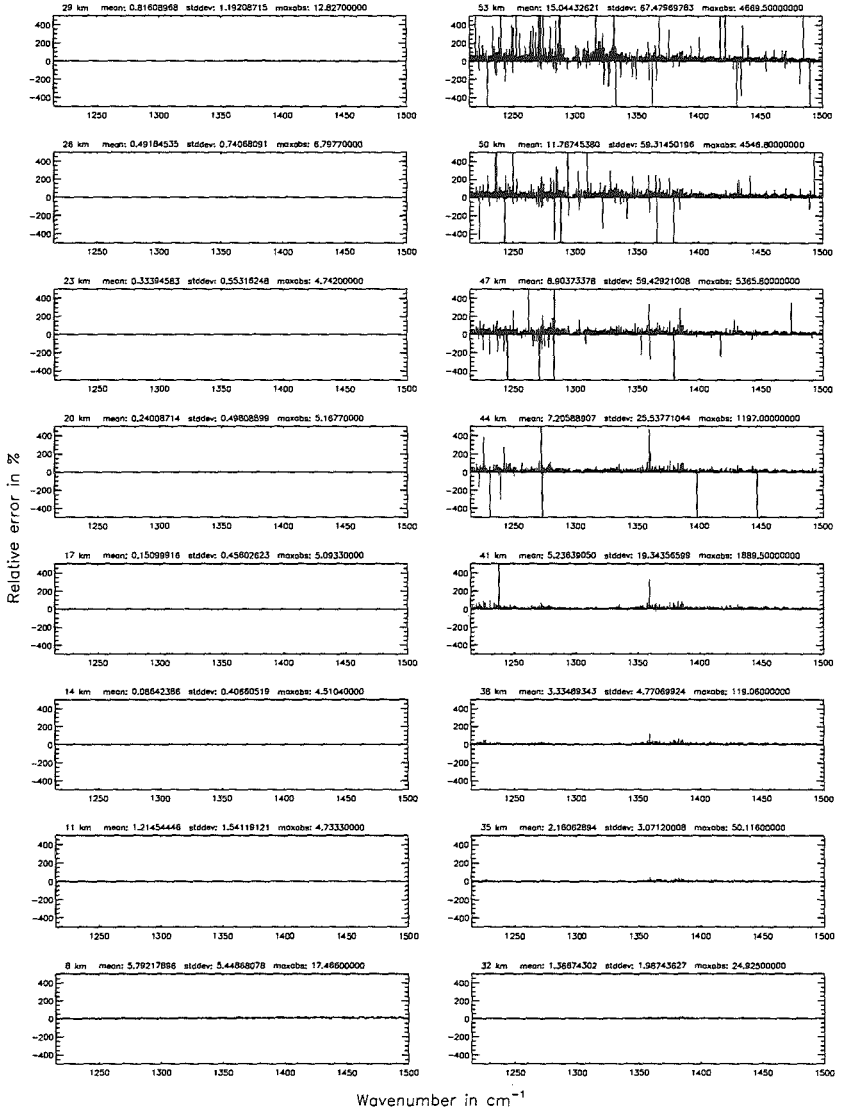


Figure 177: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

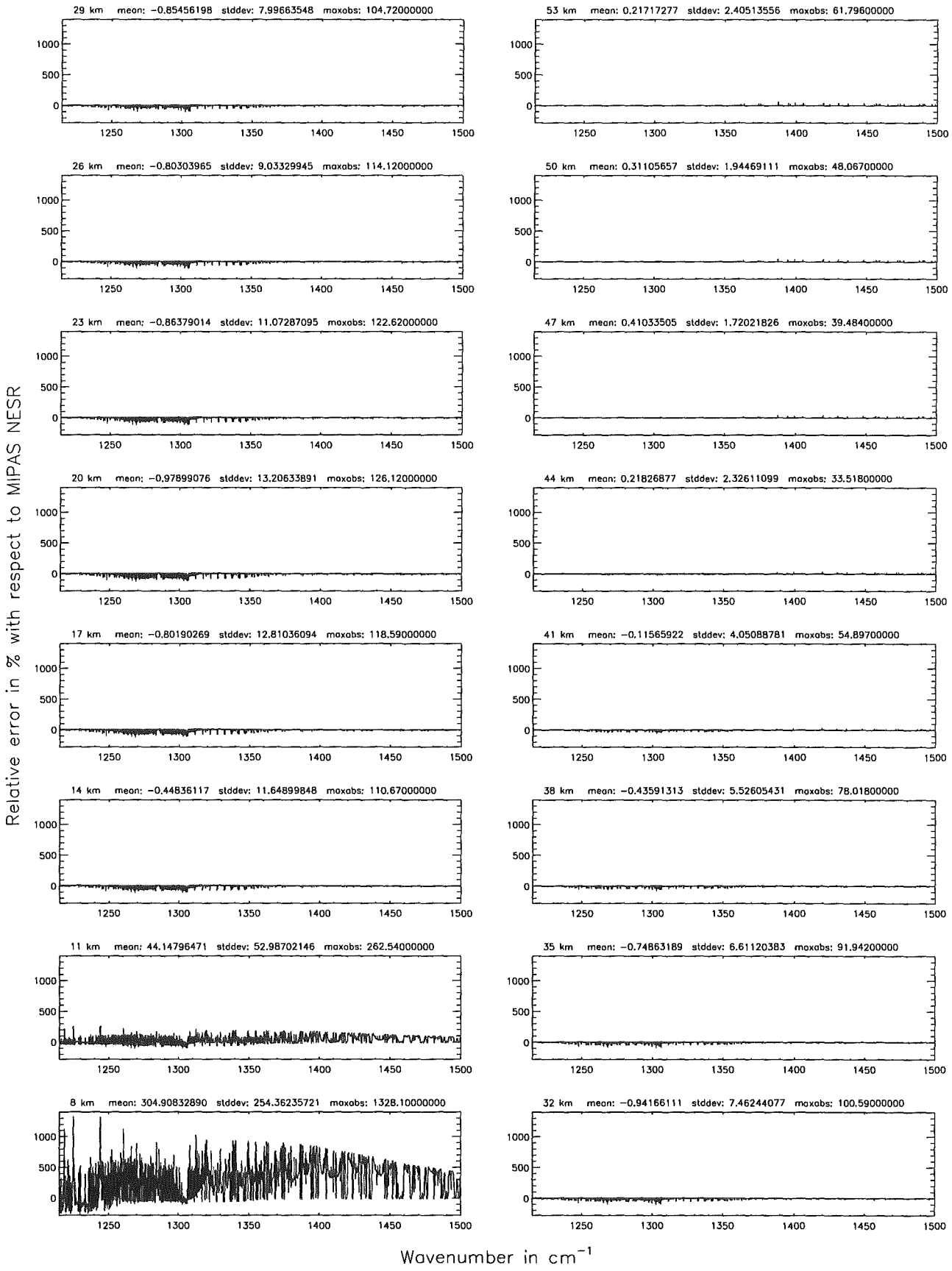
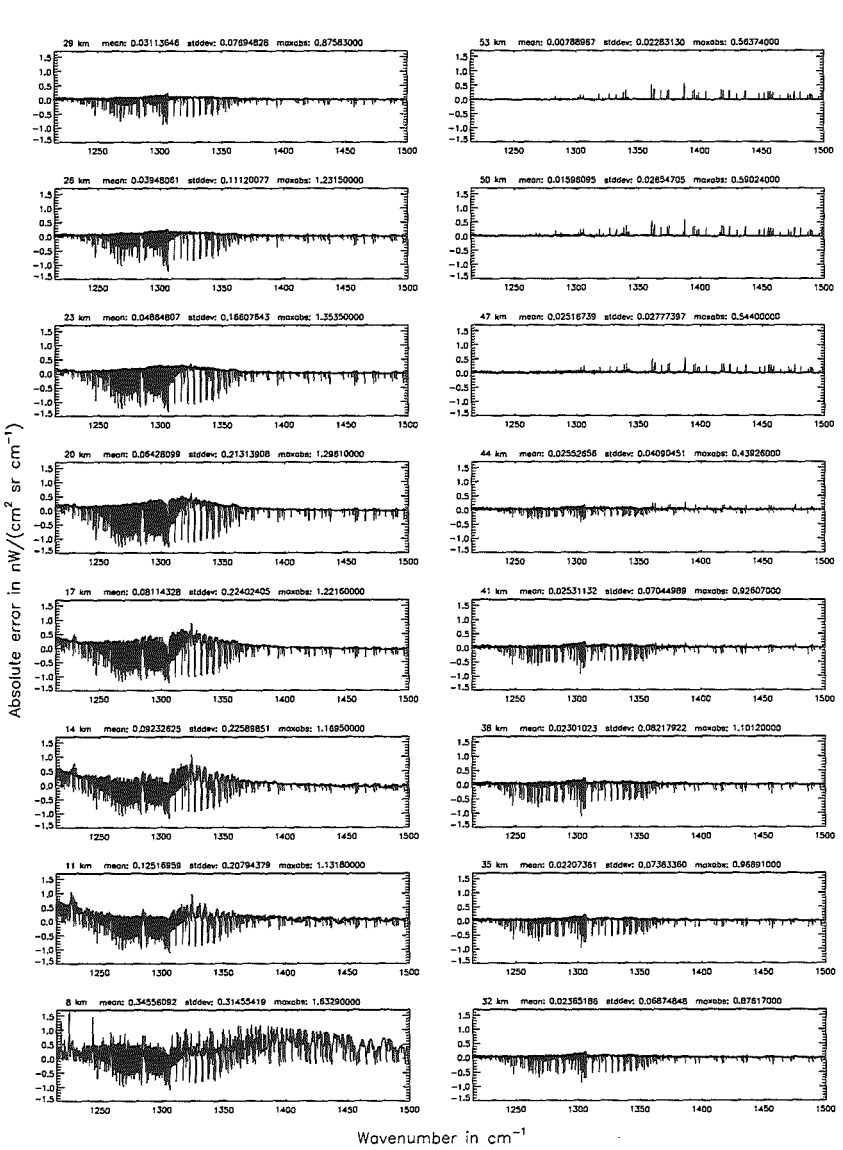


Figure 178: Relative error [%] with respect to MIPAS NESR



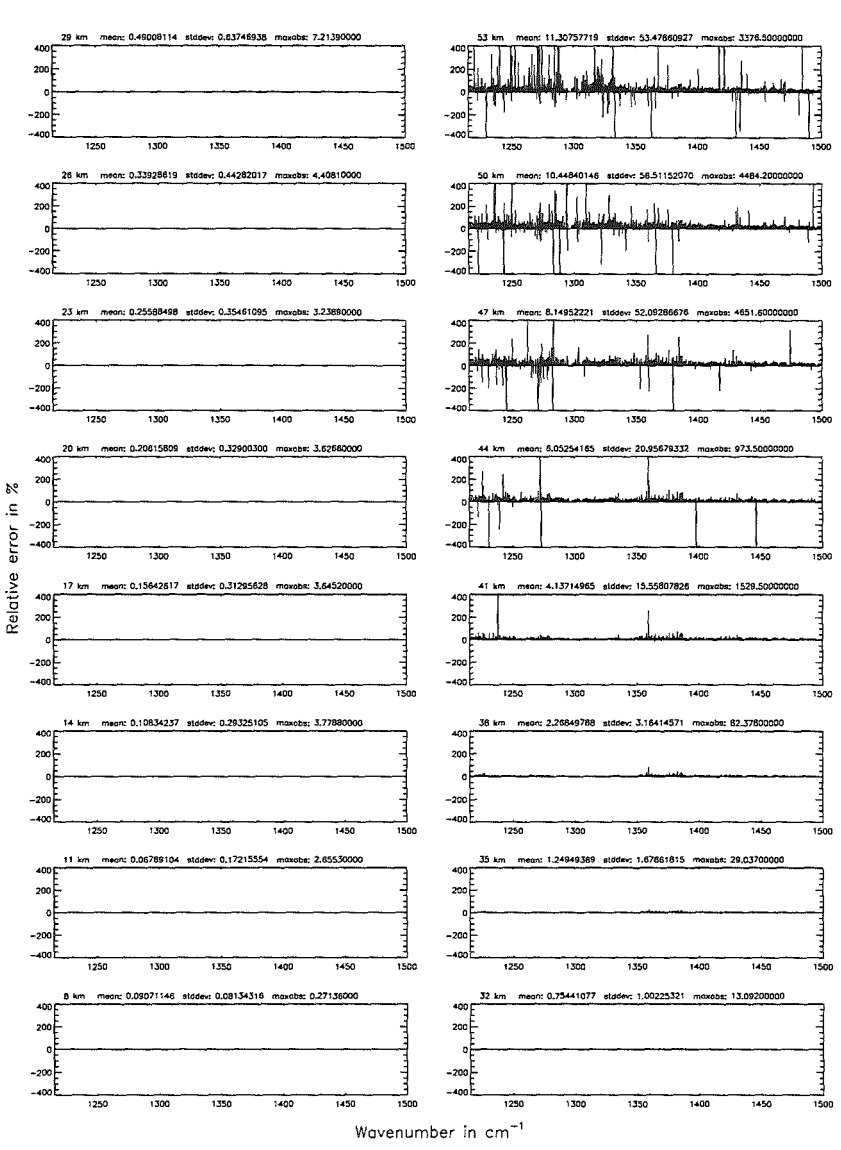


Figure 179: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (e) 77 levels; (Ref.: 156 levels)

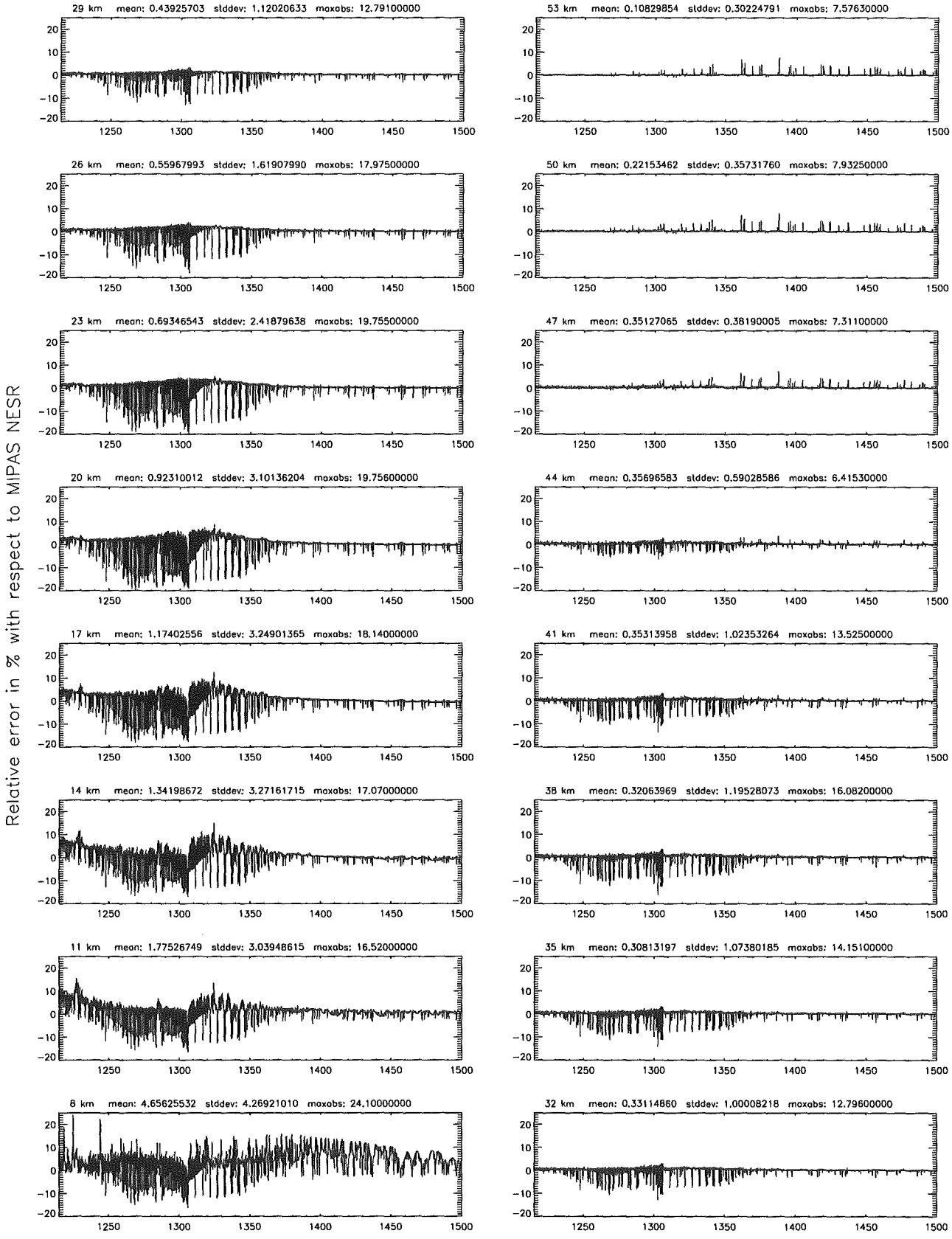
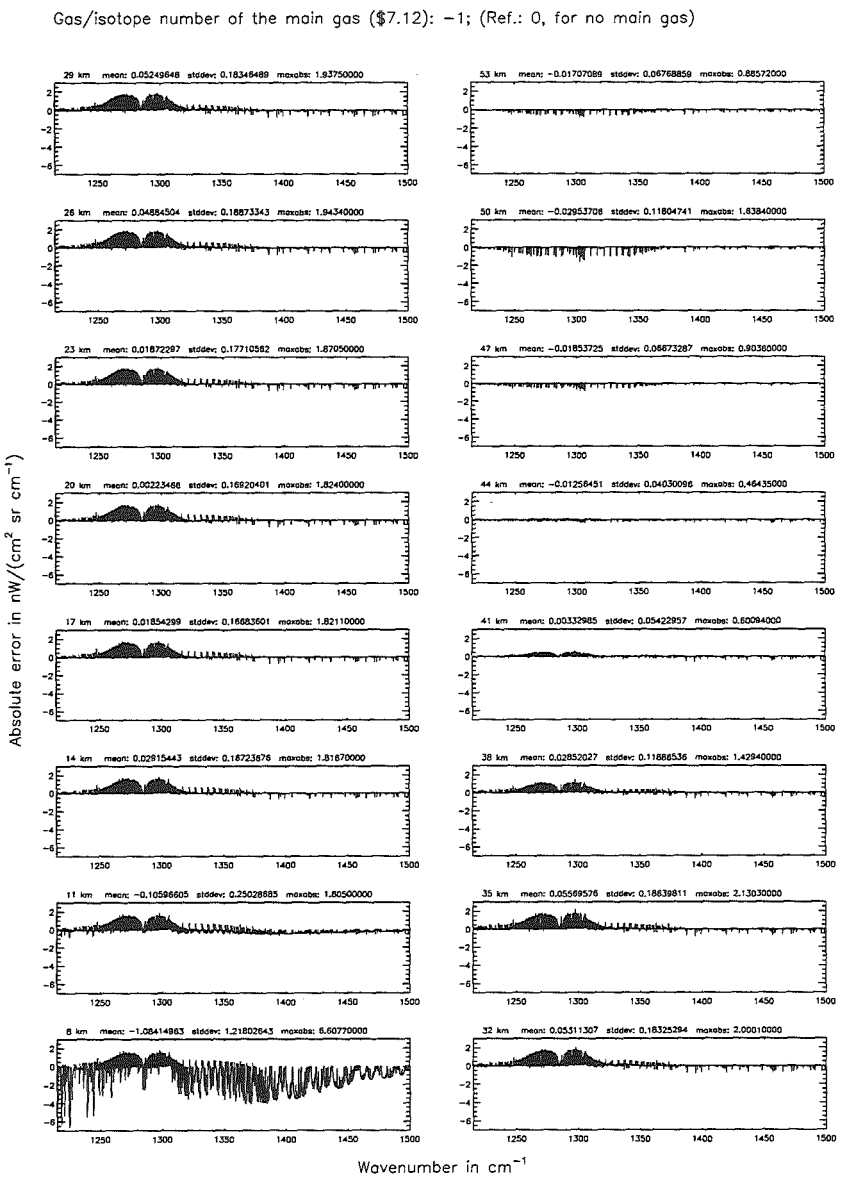


Figure 180: Relative error [%] with respect to MIPAS NESR



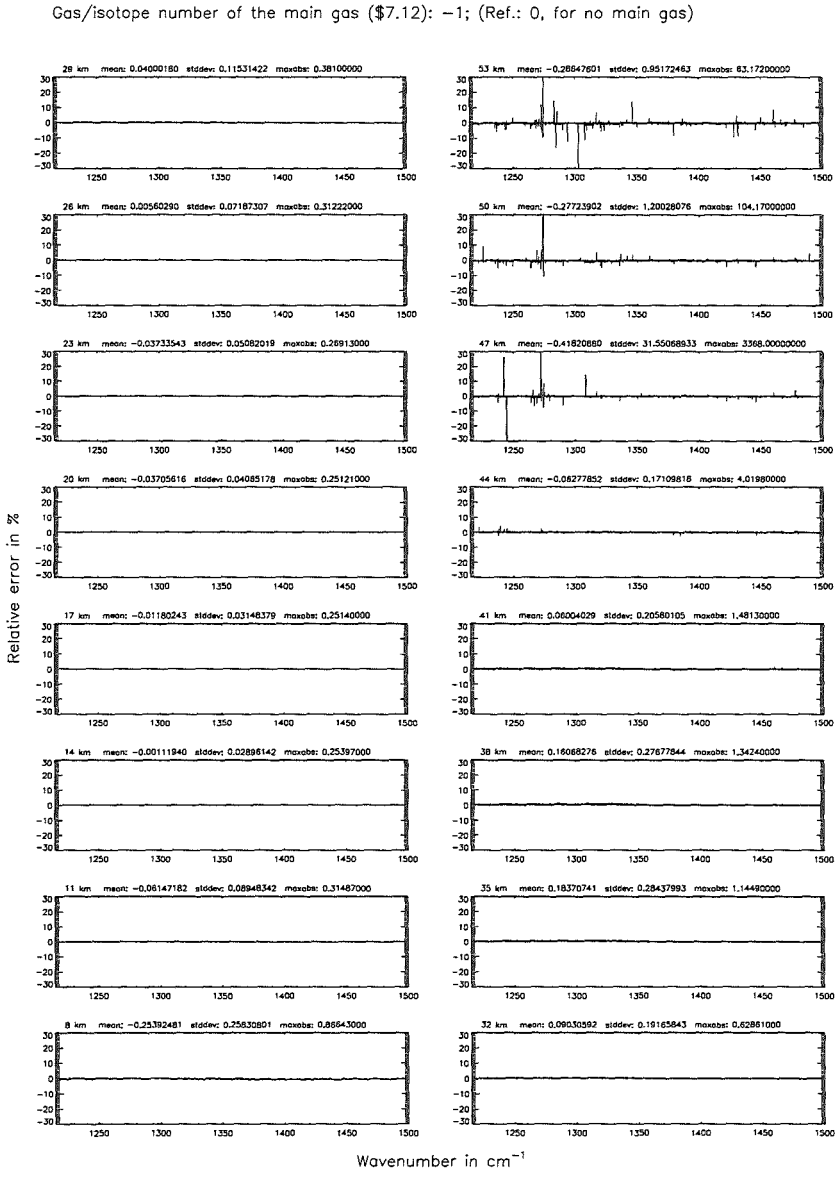


Figure 181: Absolute error [nW/(cm² sr cm⁻¹)] and relative error [%]

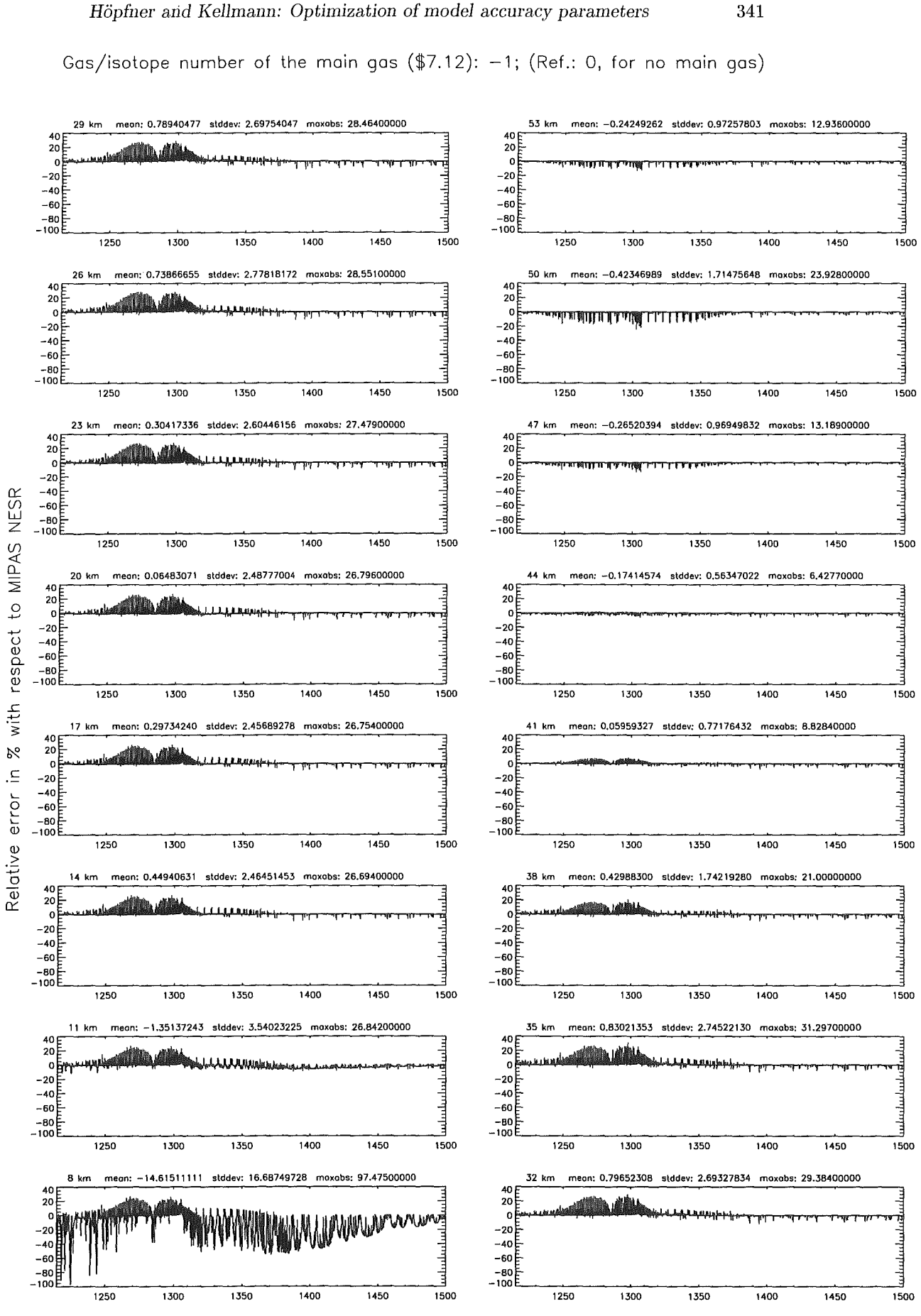
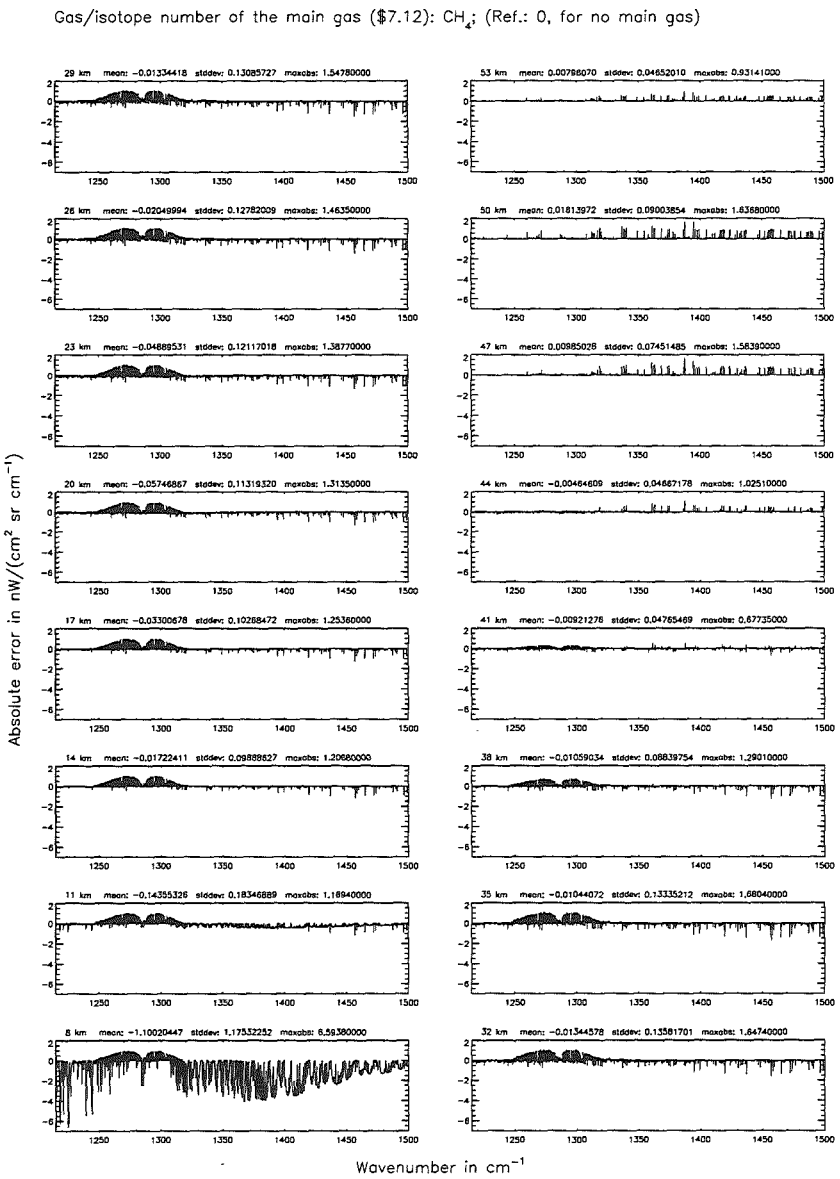


Figure 182: Relative error [%] with respect to MIPAS NESR



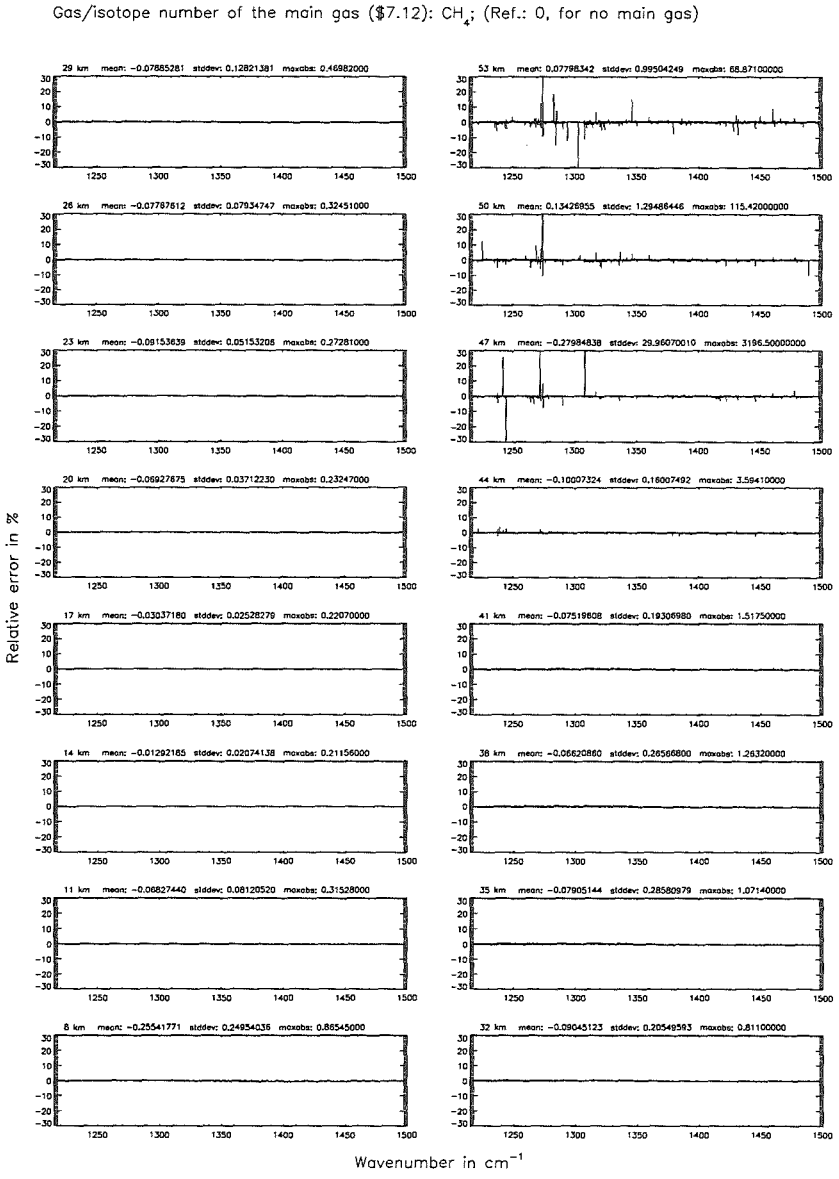


Figure 183: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

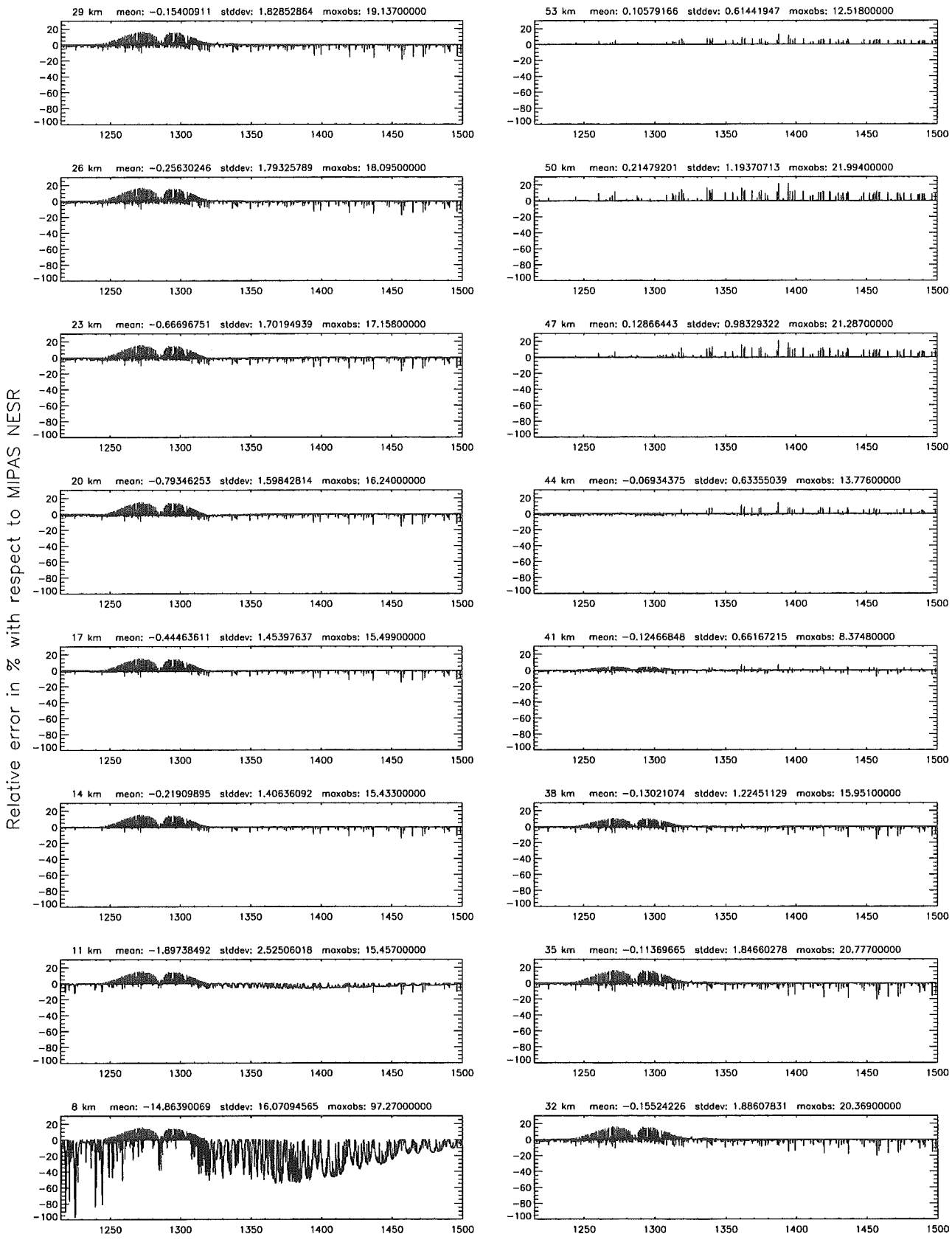
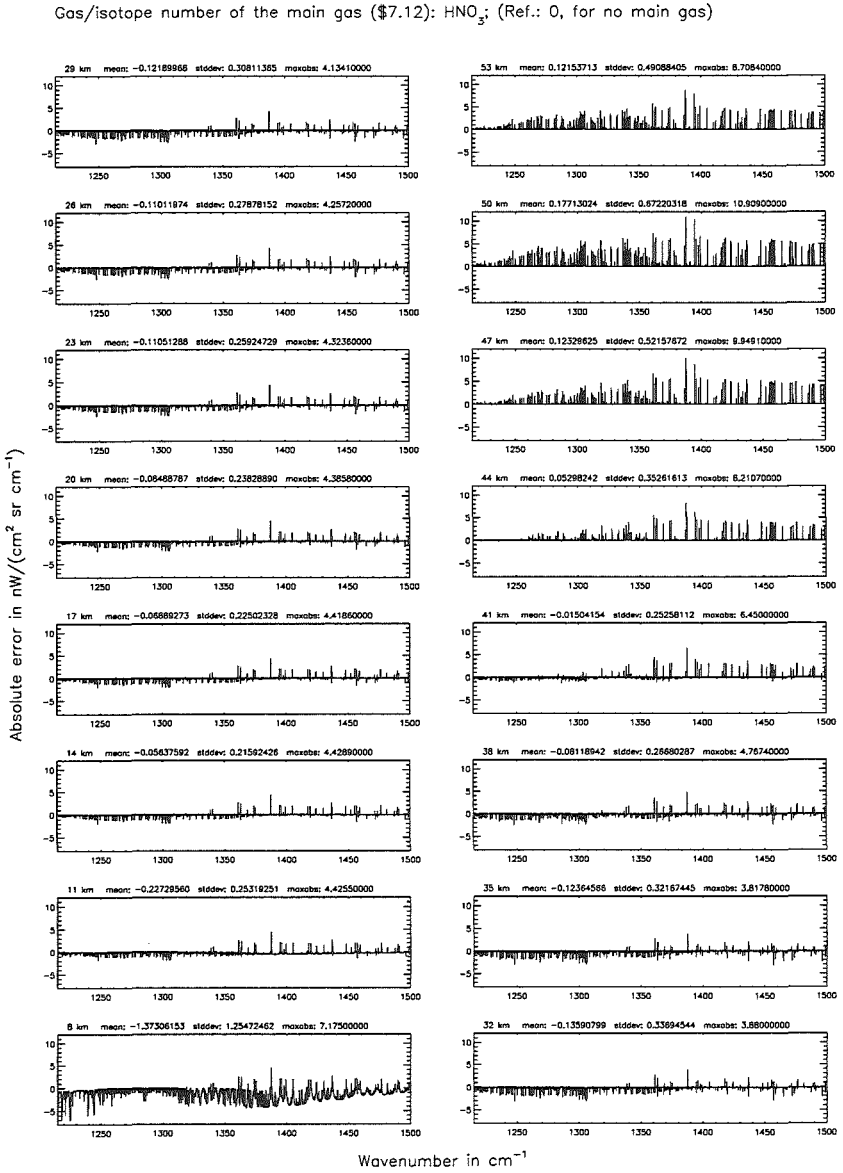


Figure 184: Relative error [%] with respect to MIPAS NESR



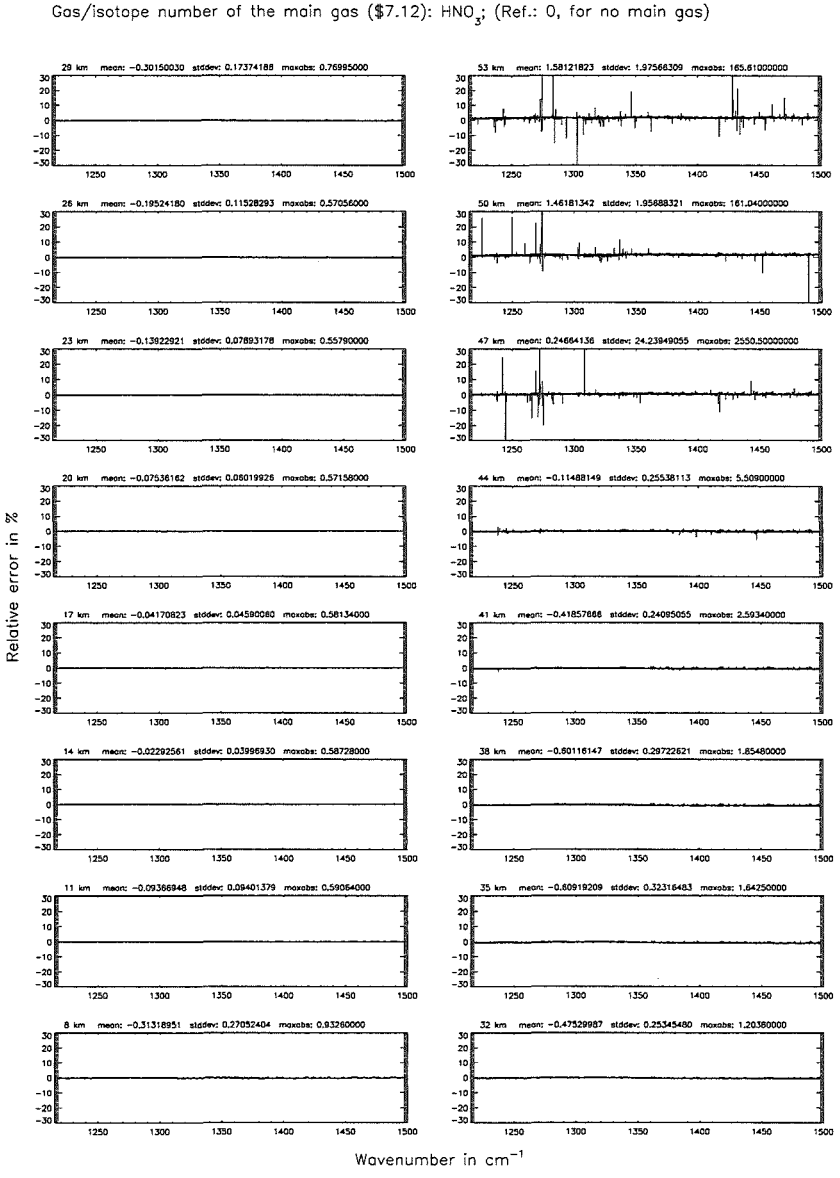


Figure 185: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

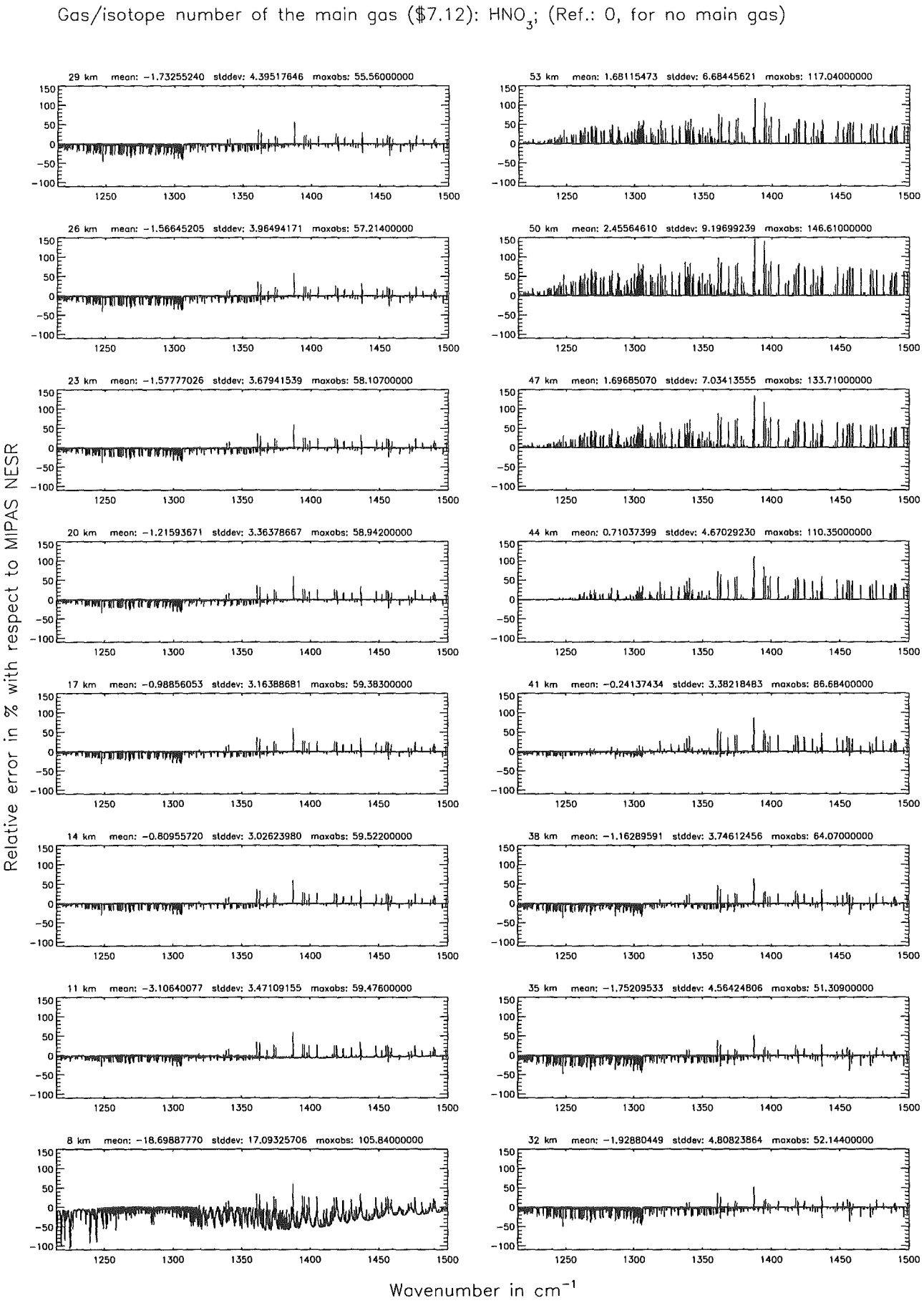
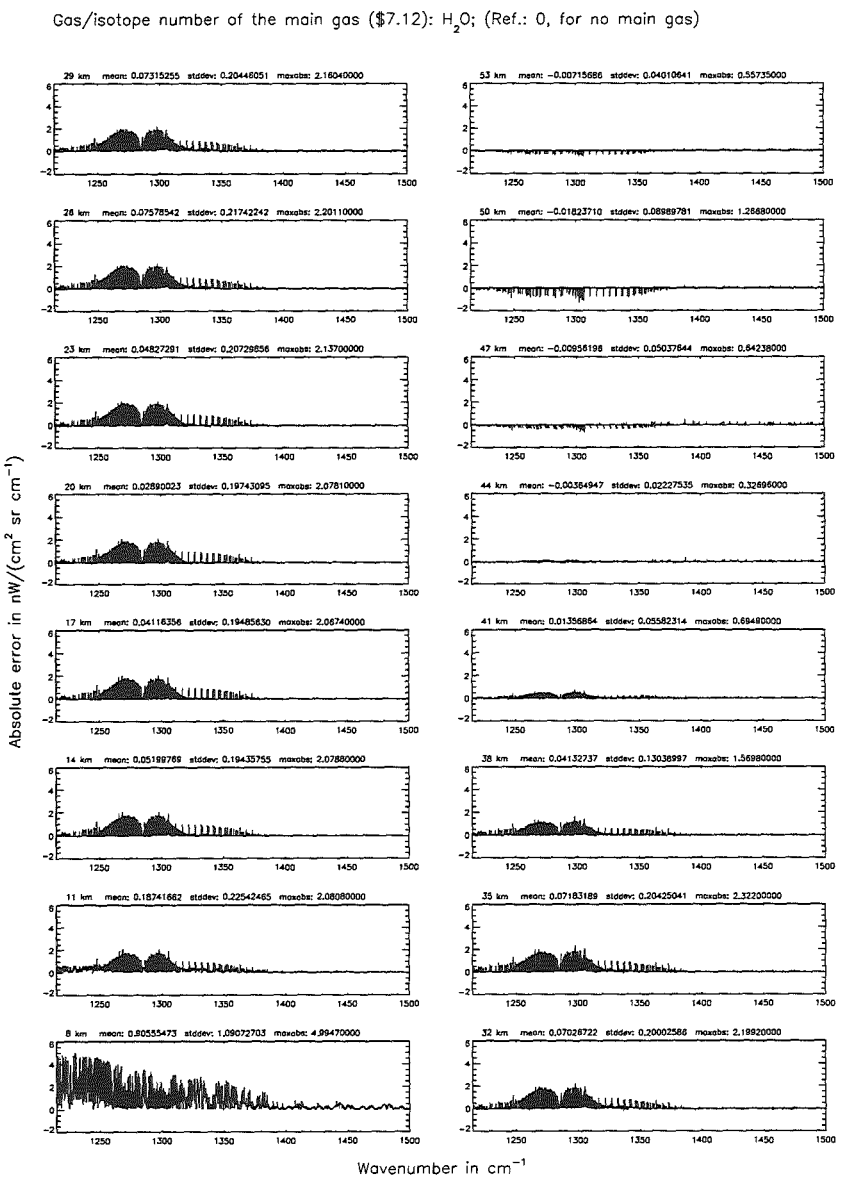


Figure 186: Relative error [%] with respect to MIPAS NESR



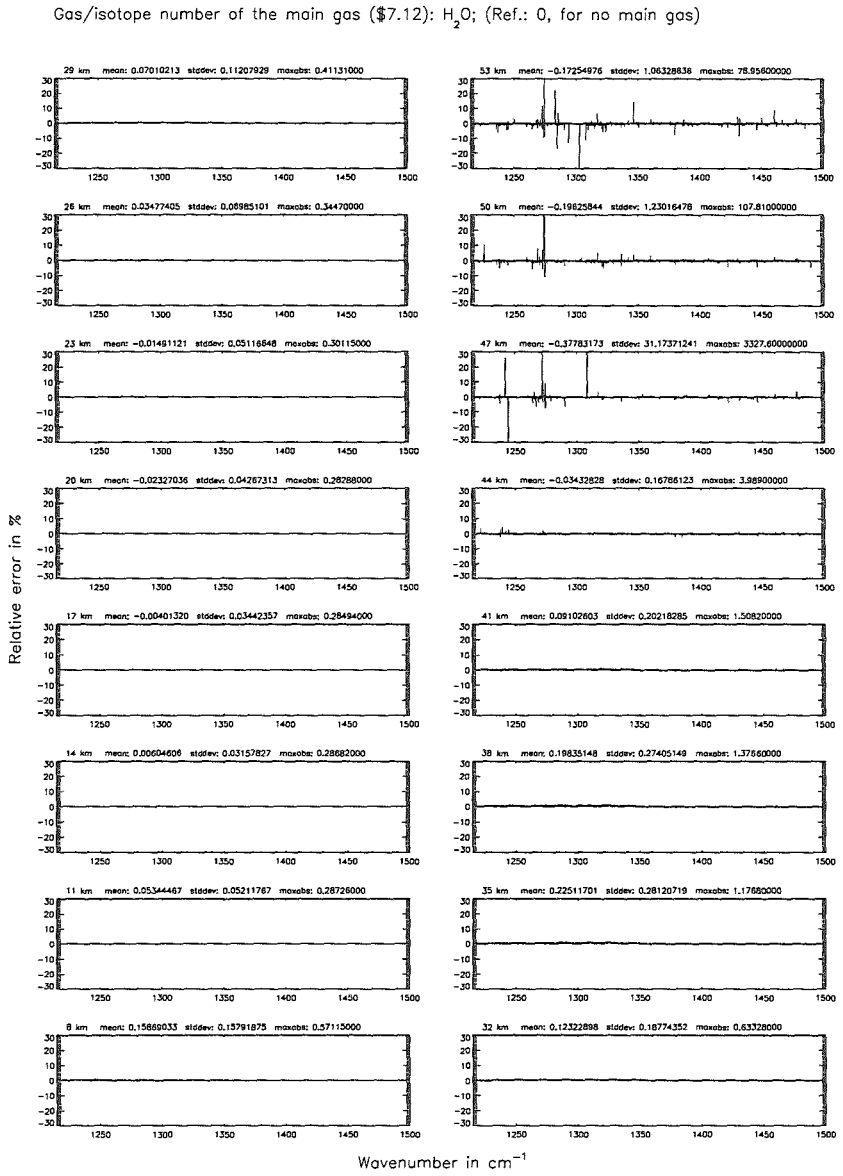


Figure 187: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

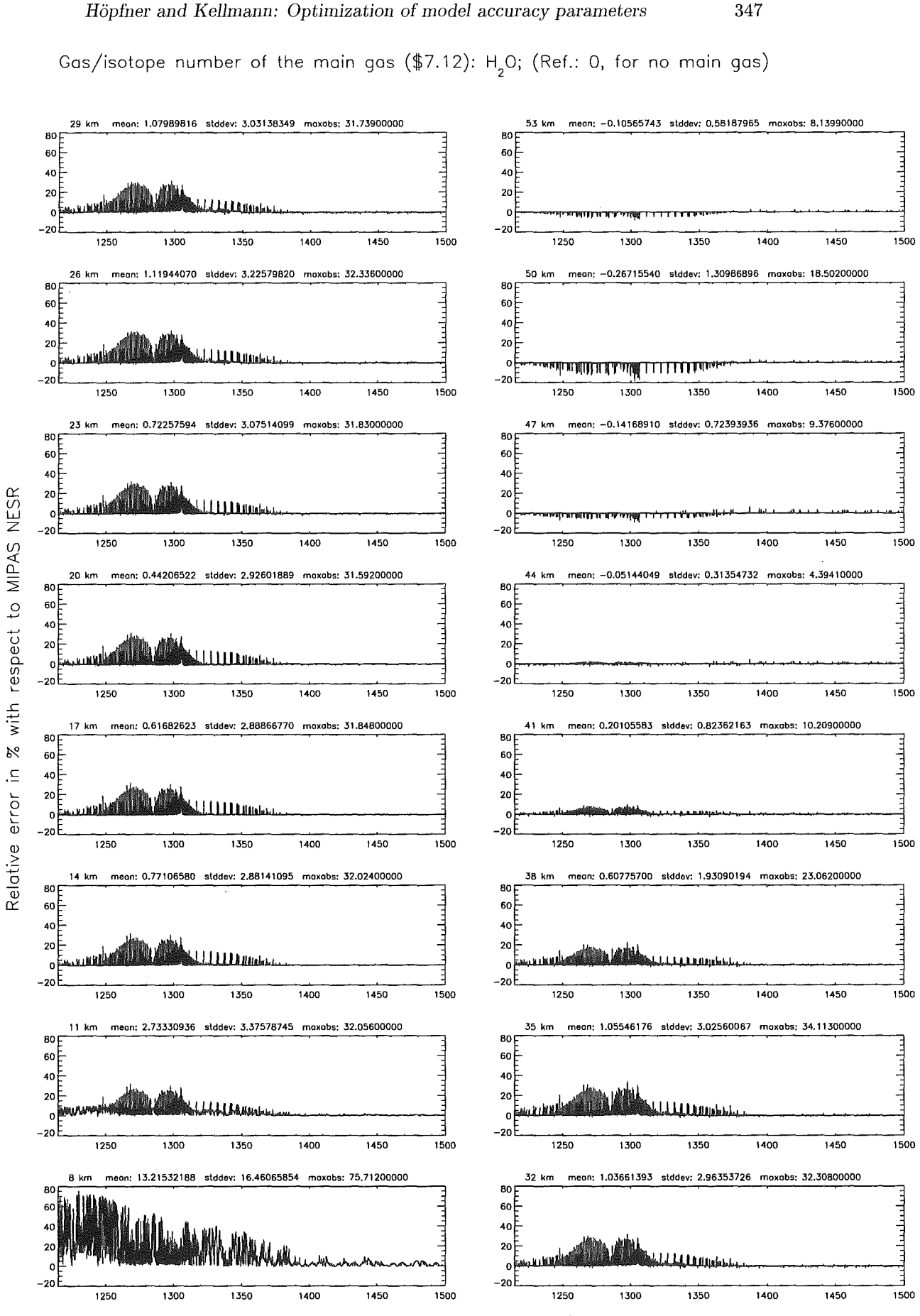


Figure 188: Relative error [%] with respect to MIPAS NESR

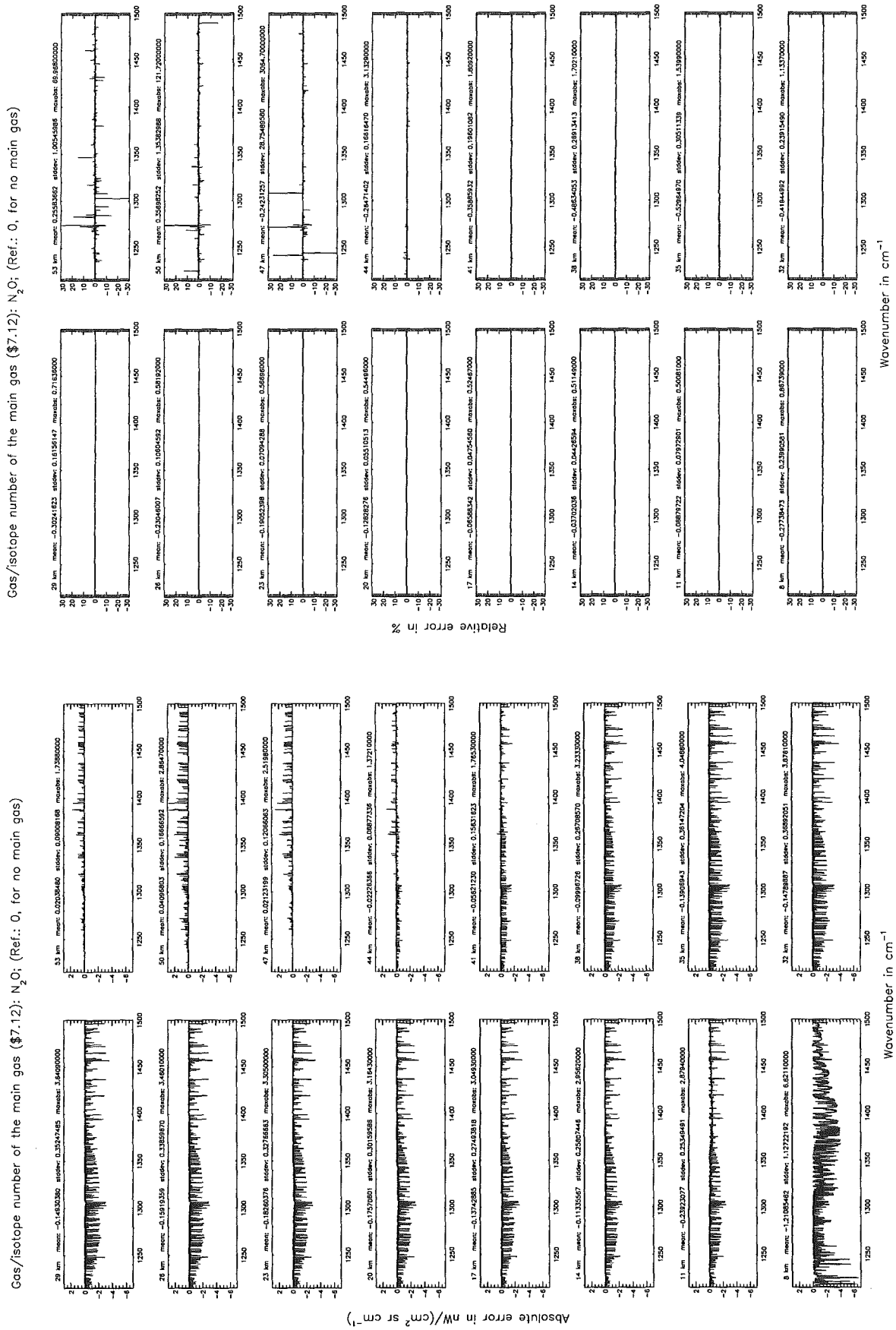


Figure 189: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

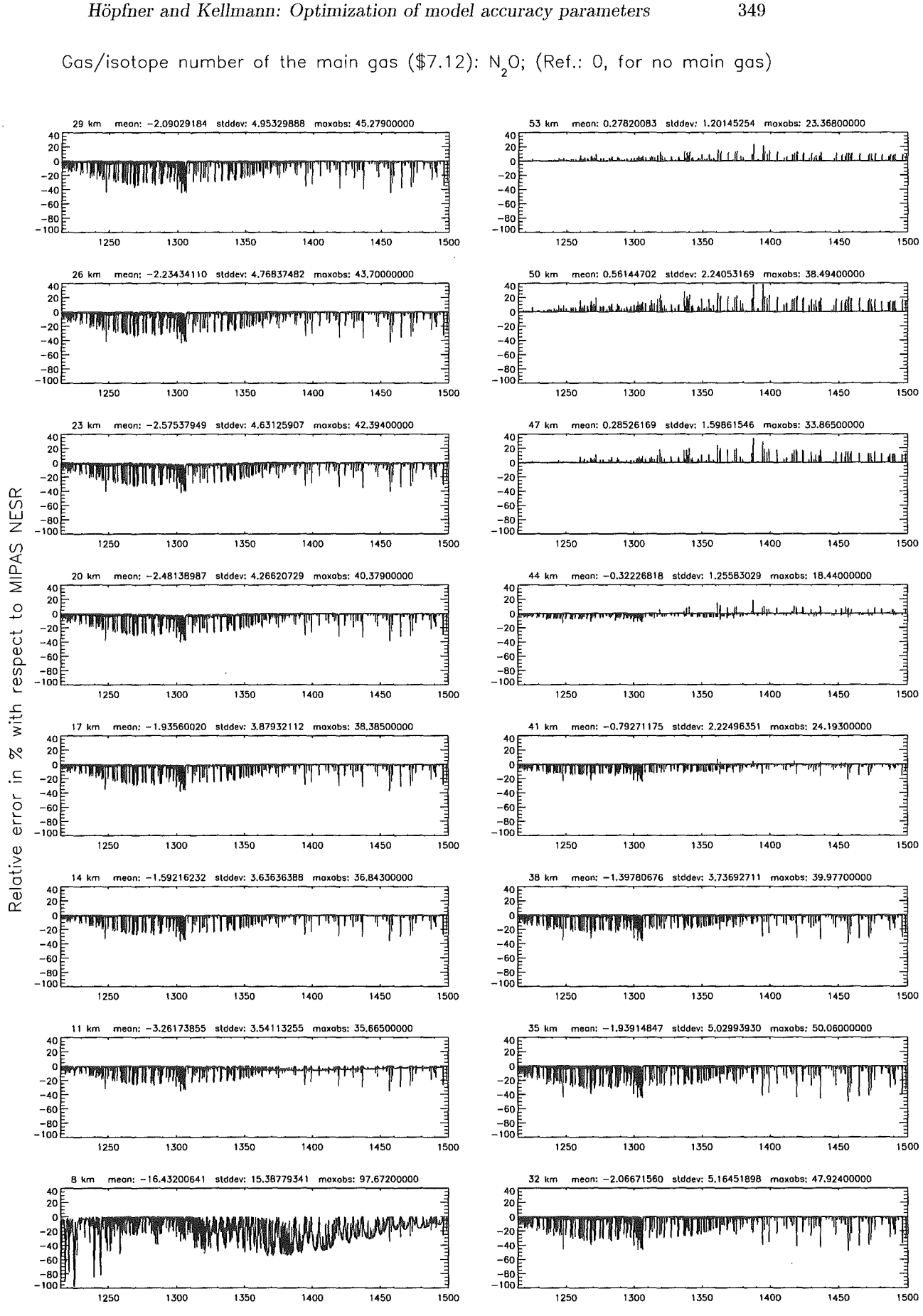
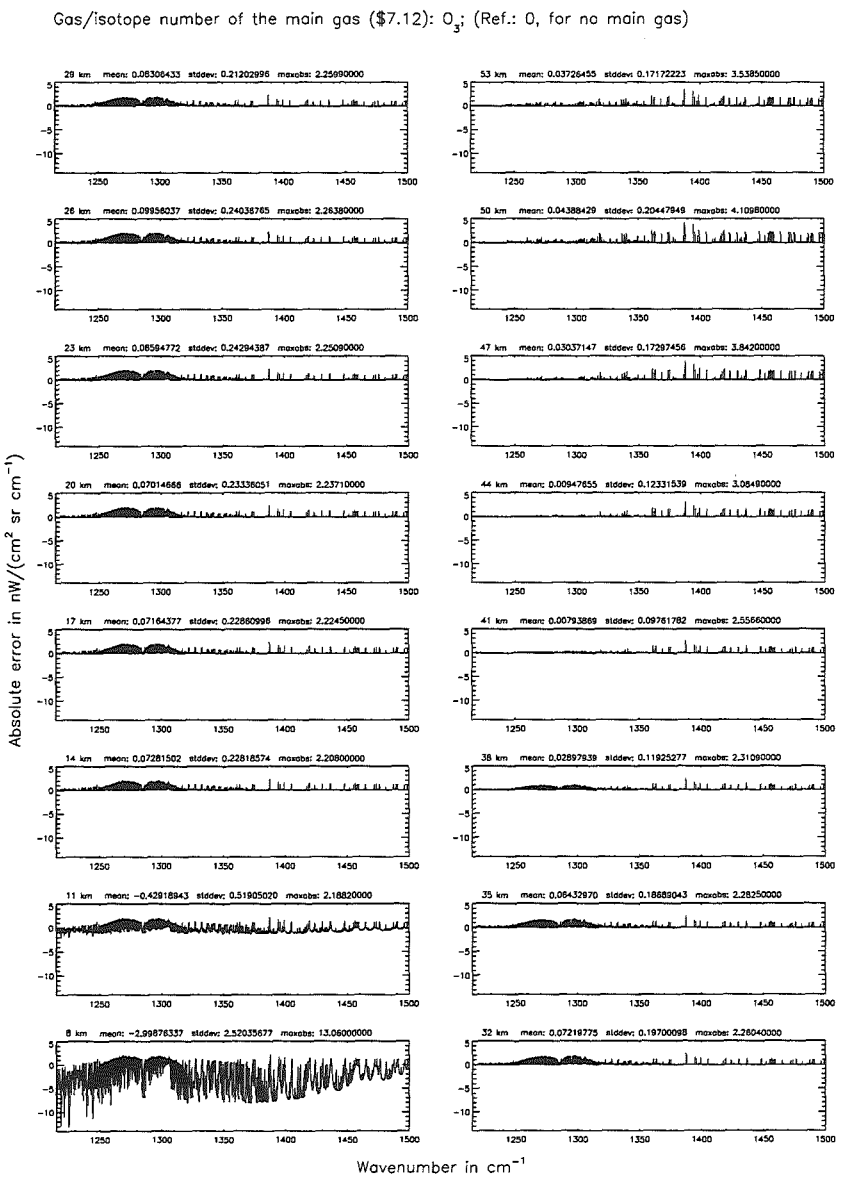


Figure 190: Relative error [%] with respect to MIPAS NESR



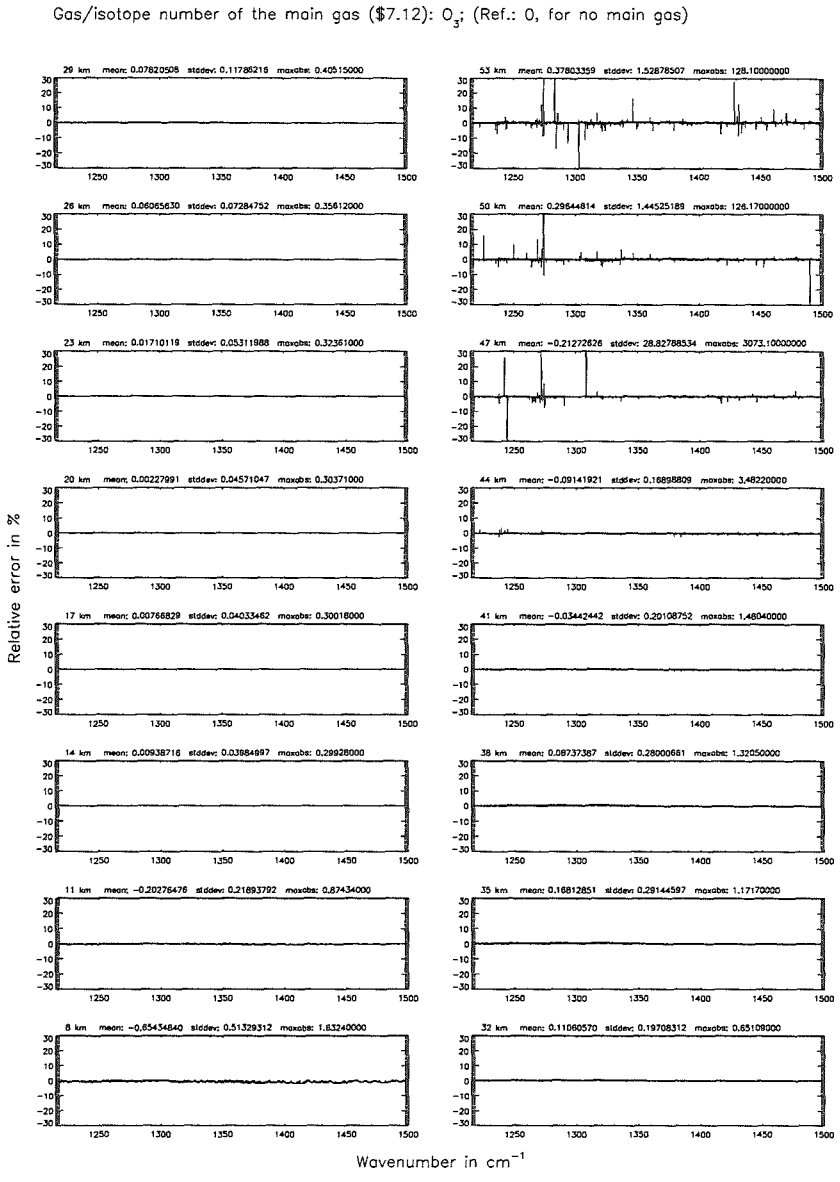


Figure 191: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

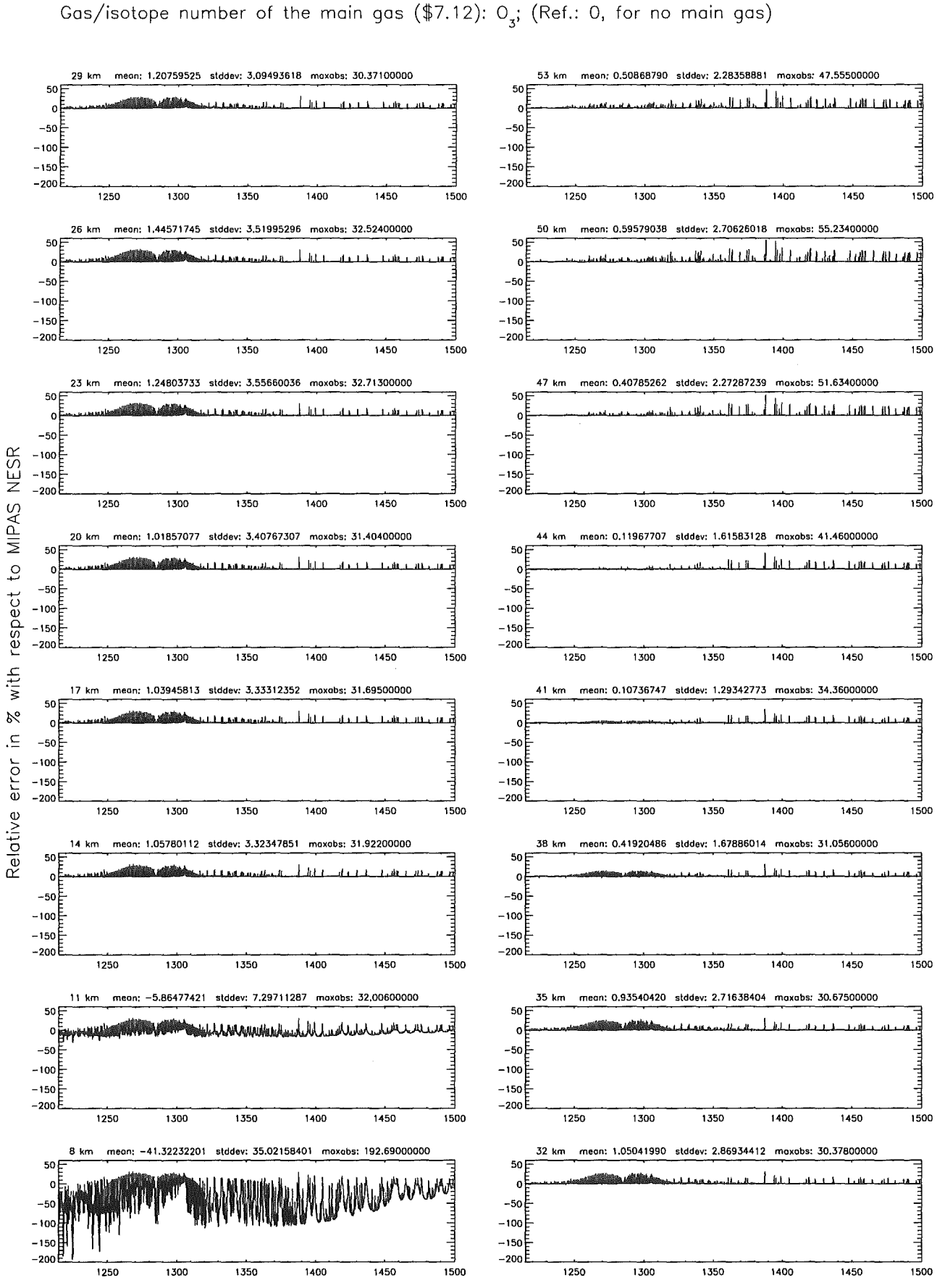
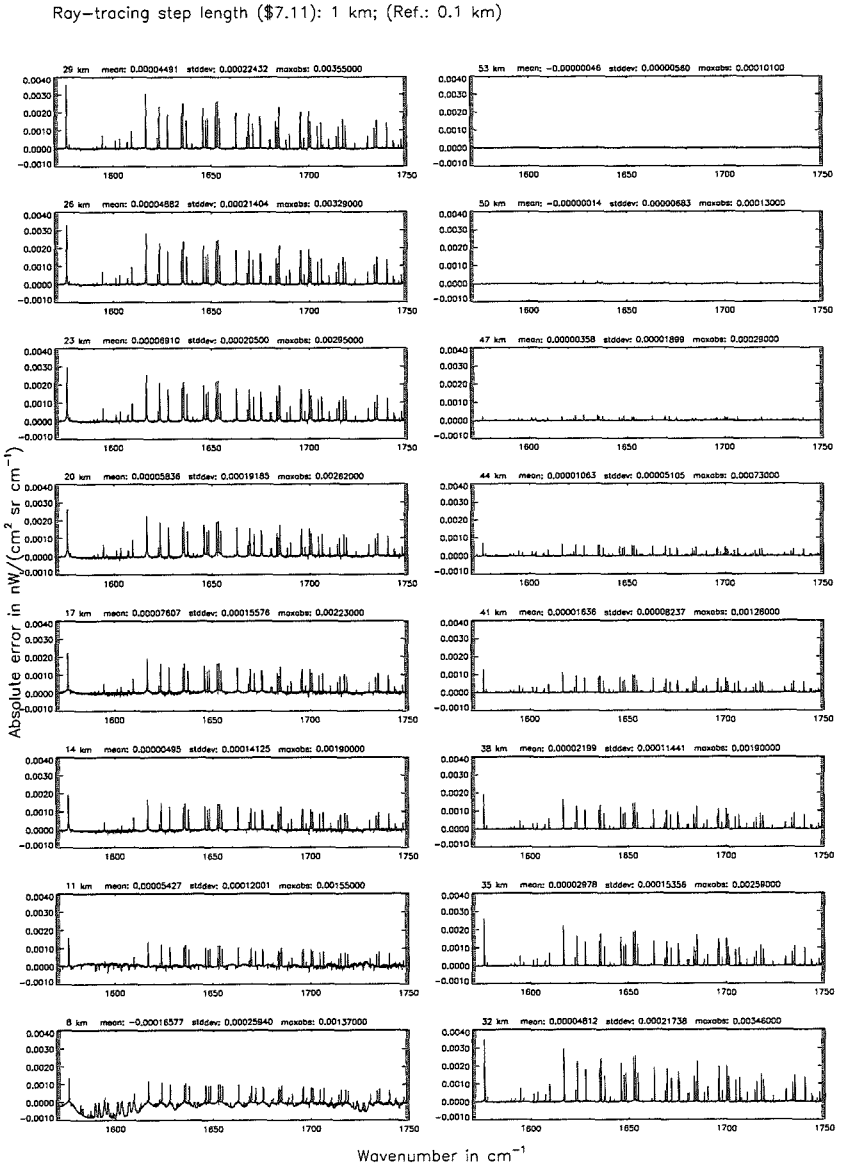


Figure 192: Relative error [%] with respect to MIPAS NESR

Band C: 1570 – 1750 cm^{-1}



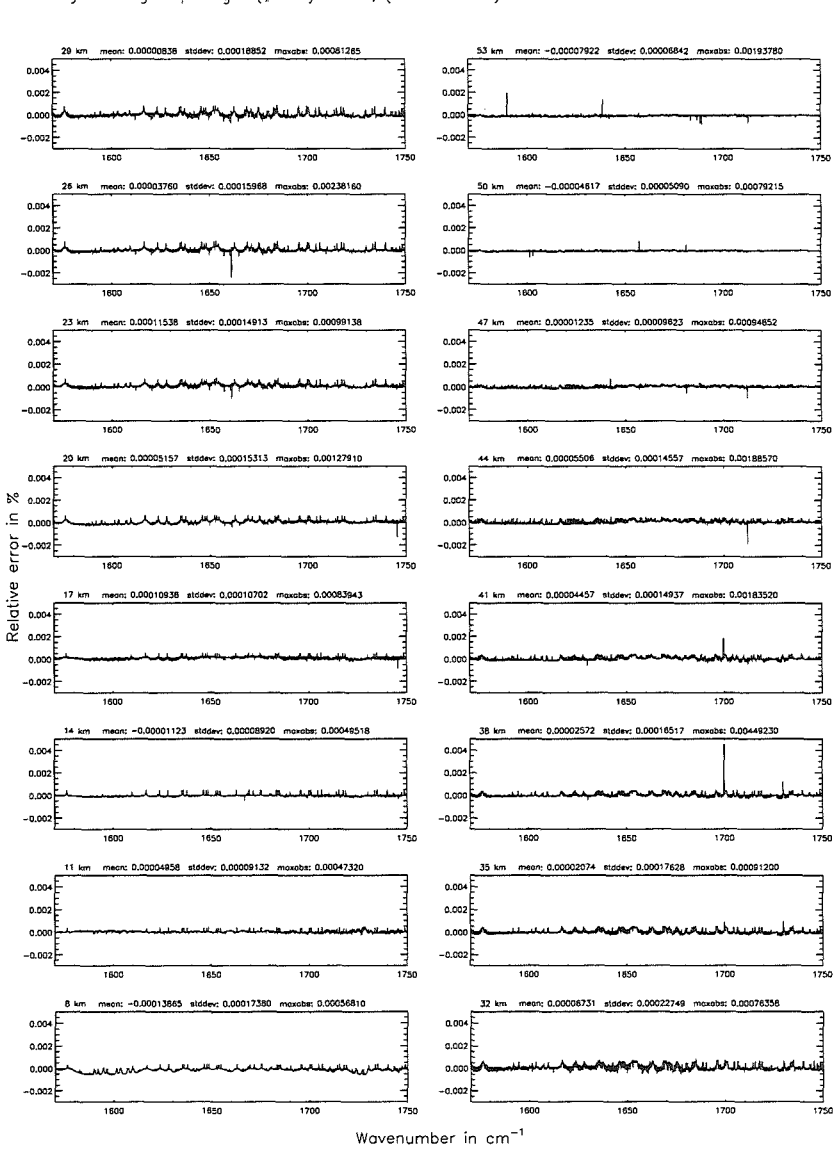


Figure 193: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

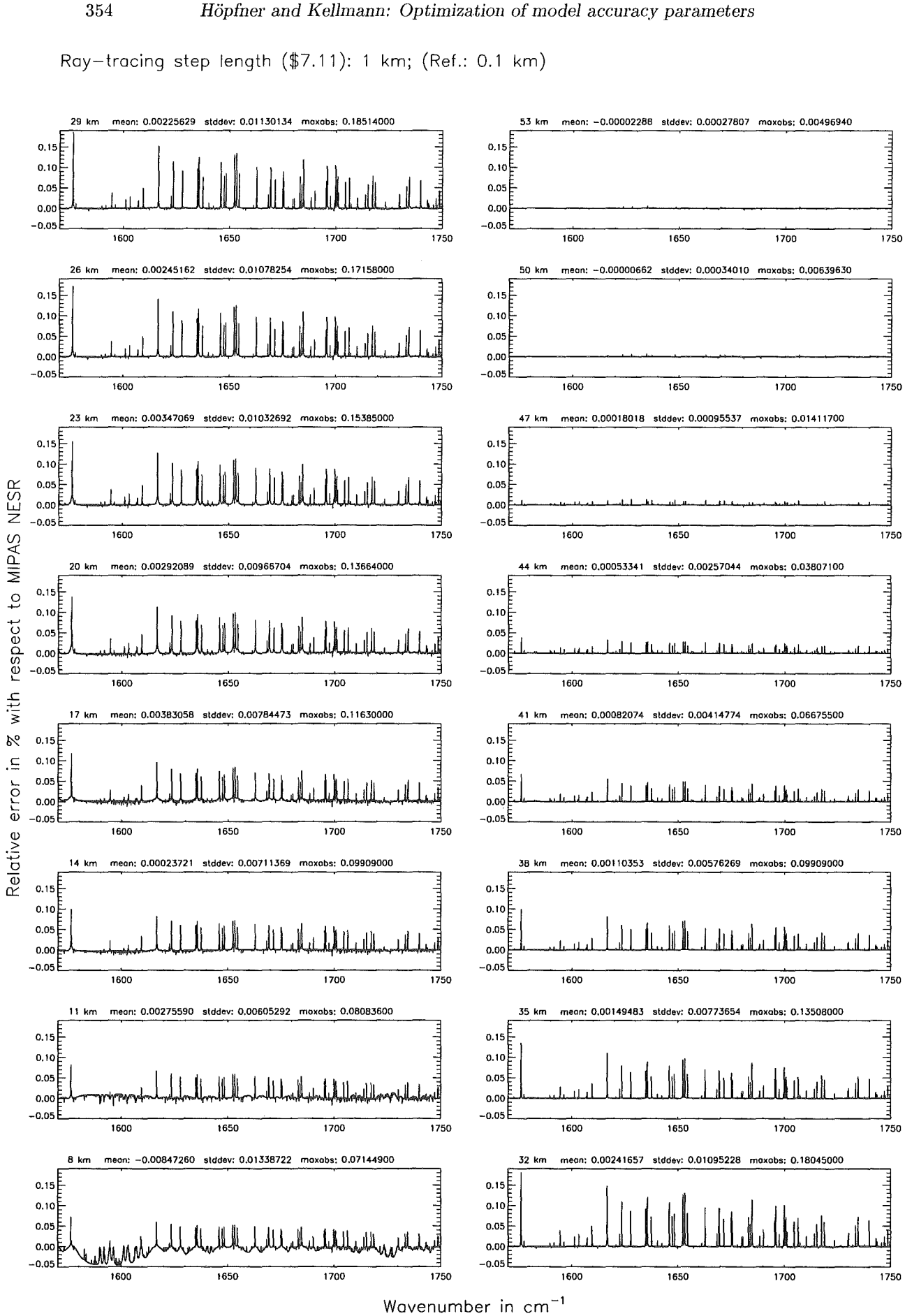


Figure 194: Relative error [%] with respect to MIPAS NESR

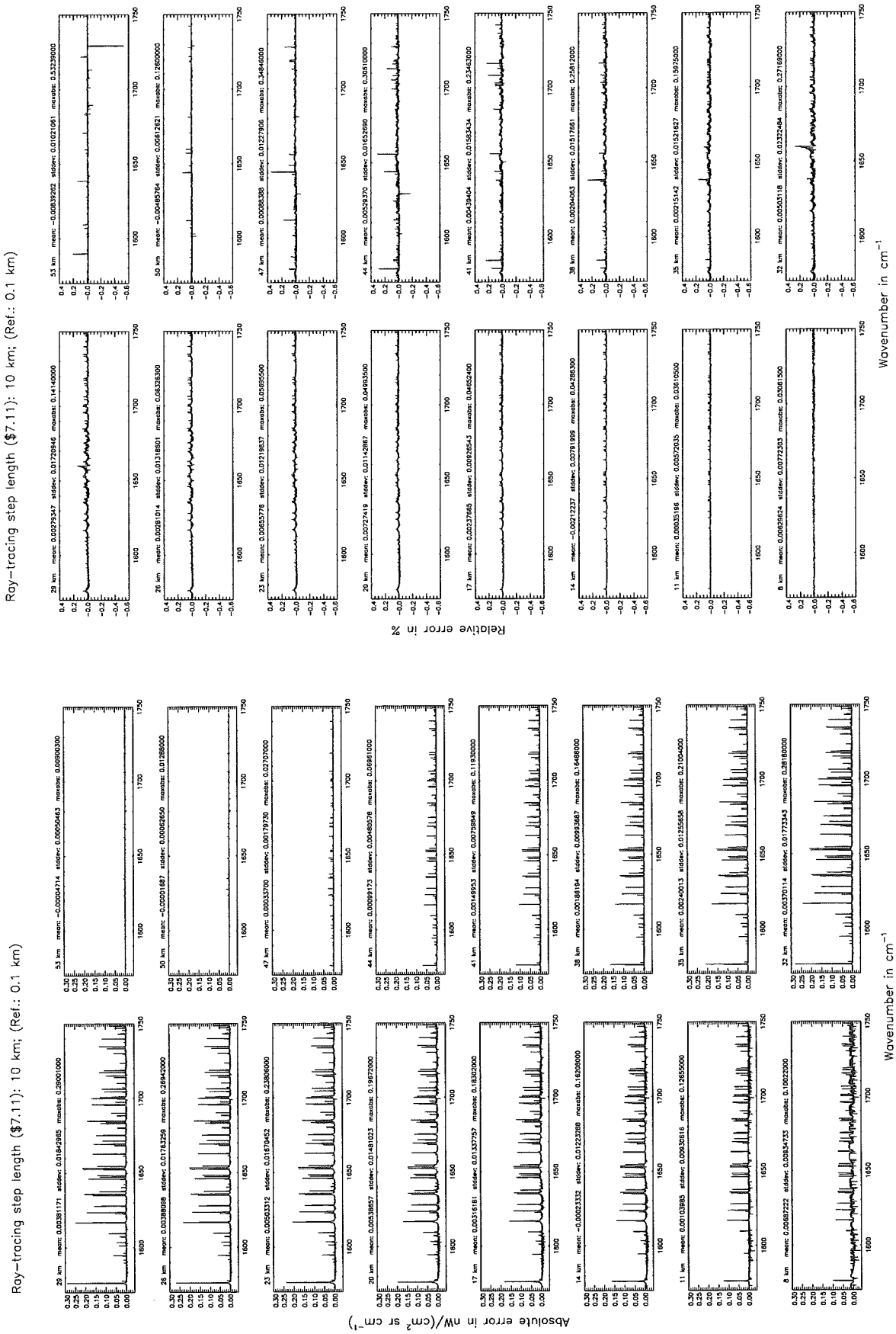


Figure 195: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

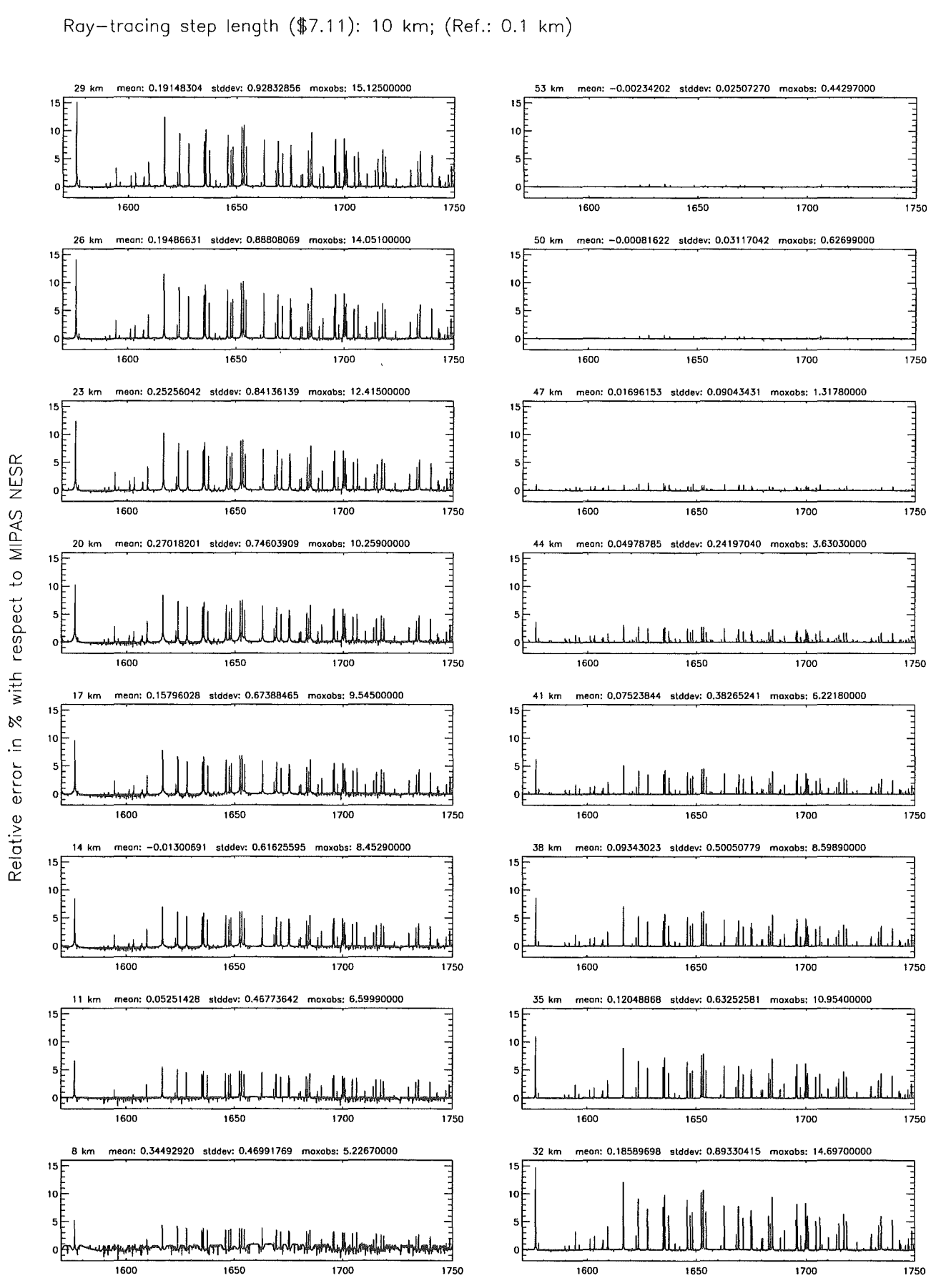


Figure 196: Relative error [%] with respect to MIPAS NESR

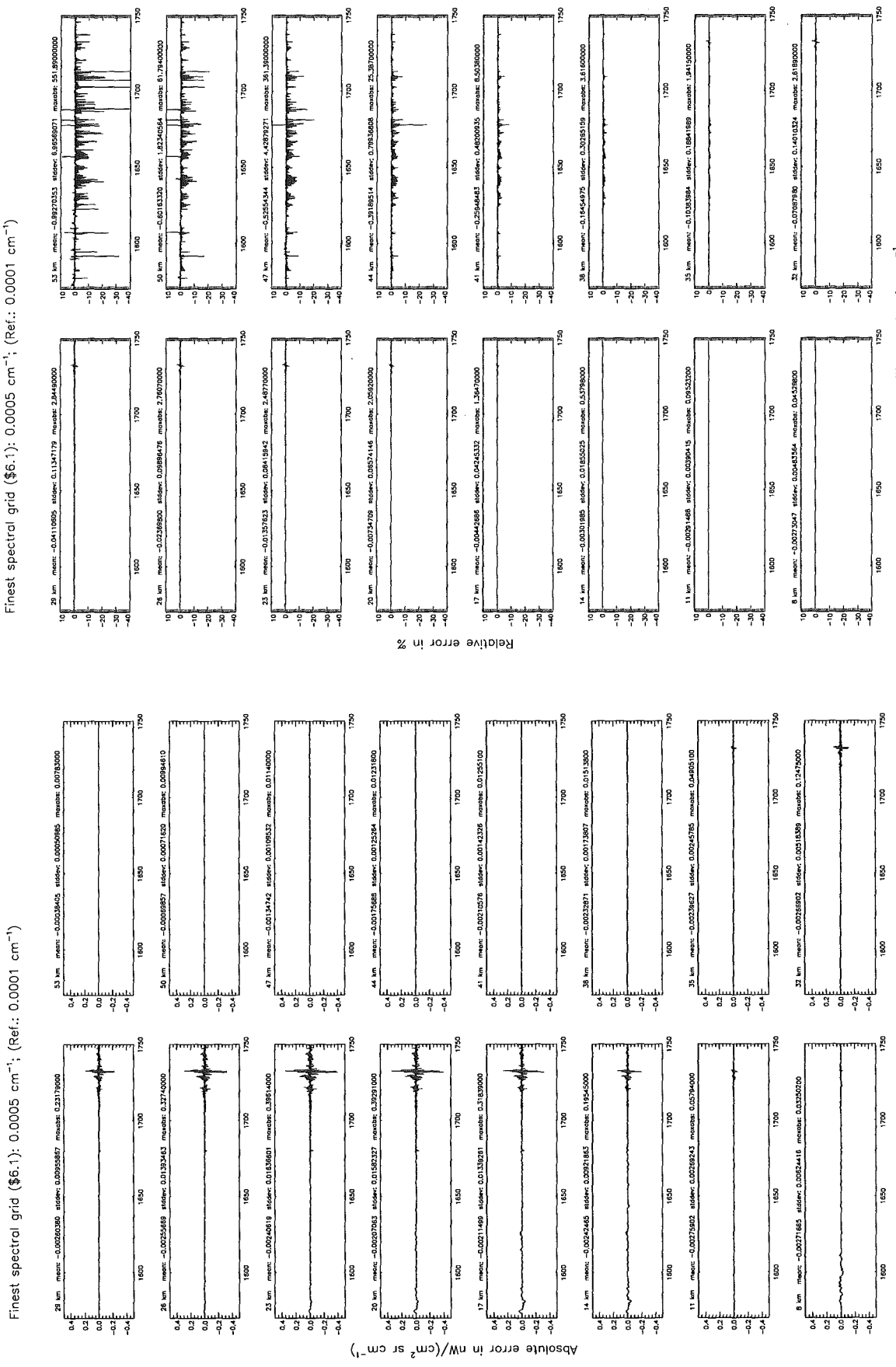


Figure 197: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

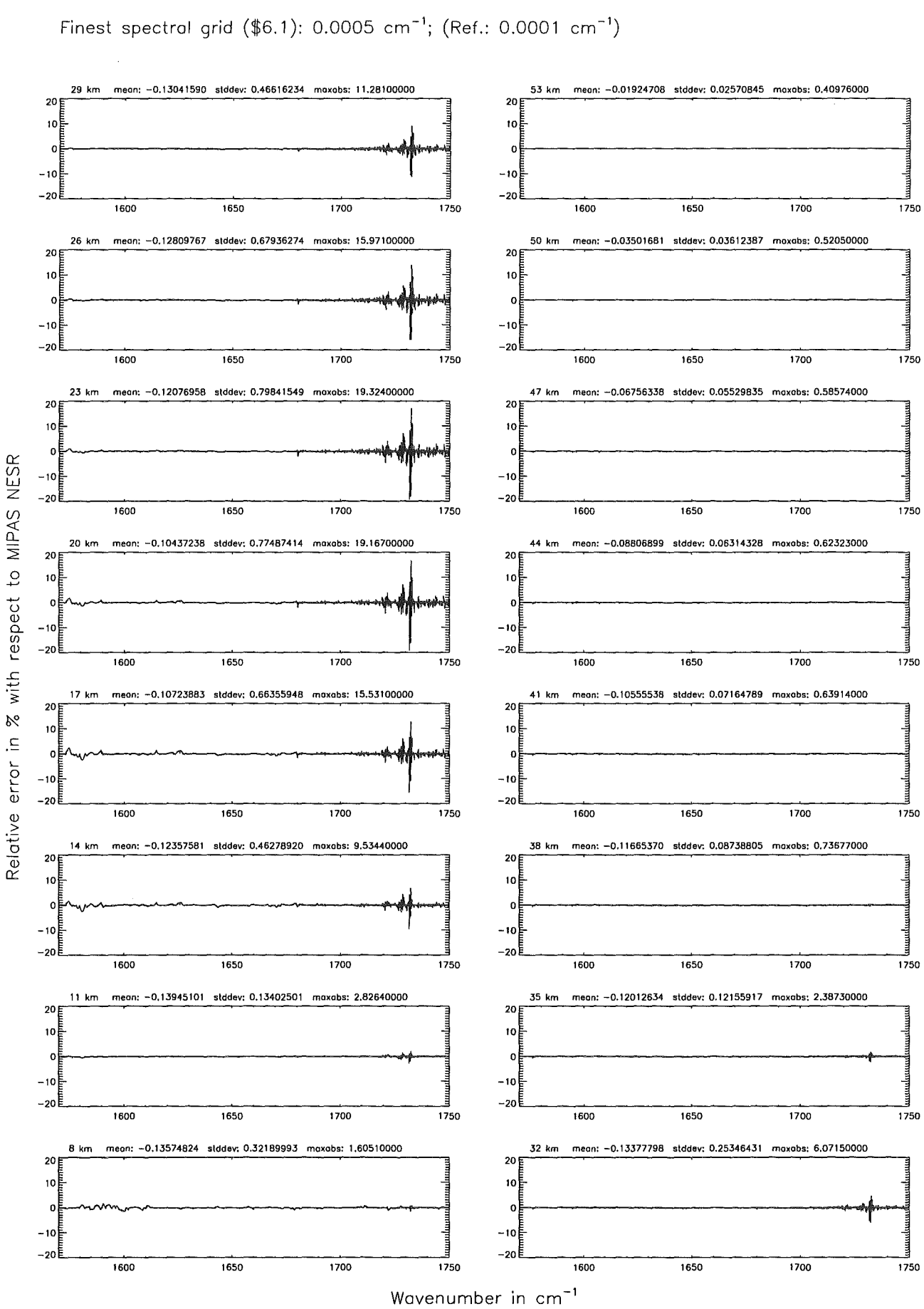


Figure 198: Relative error [%] with respect to MIPAS NESR

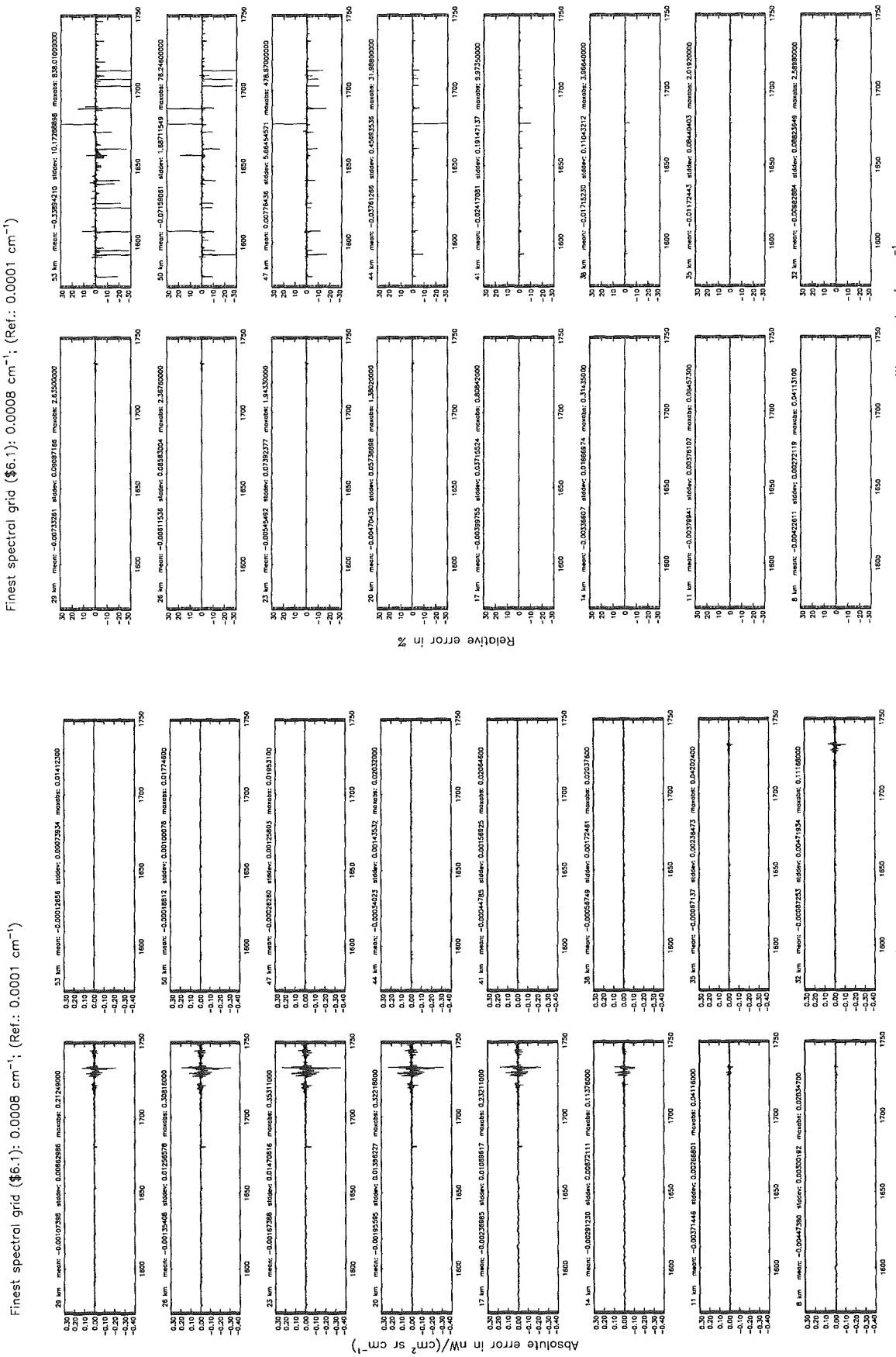


Figure 199: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

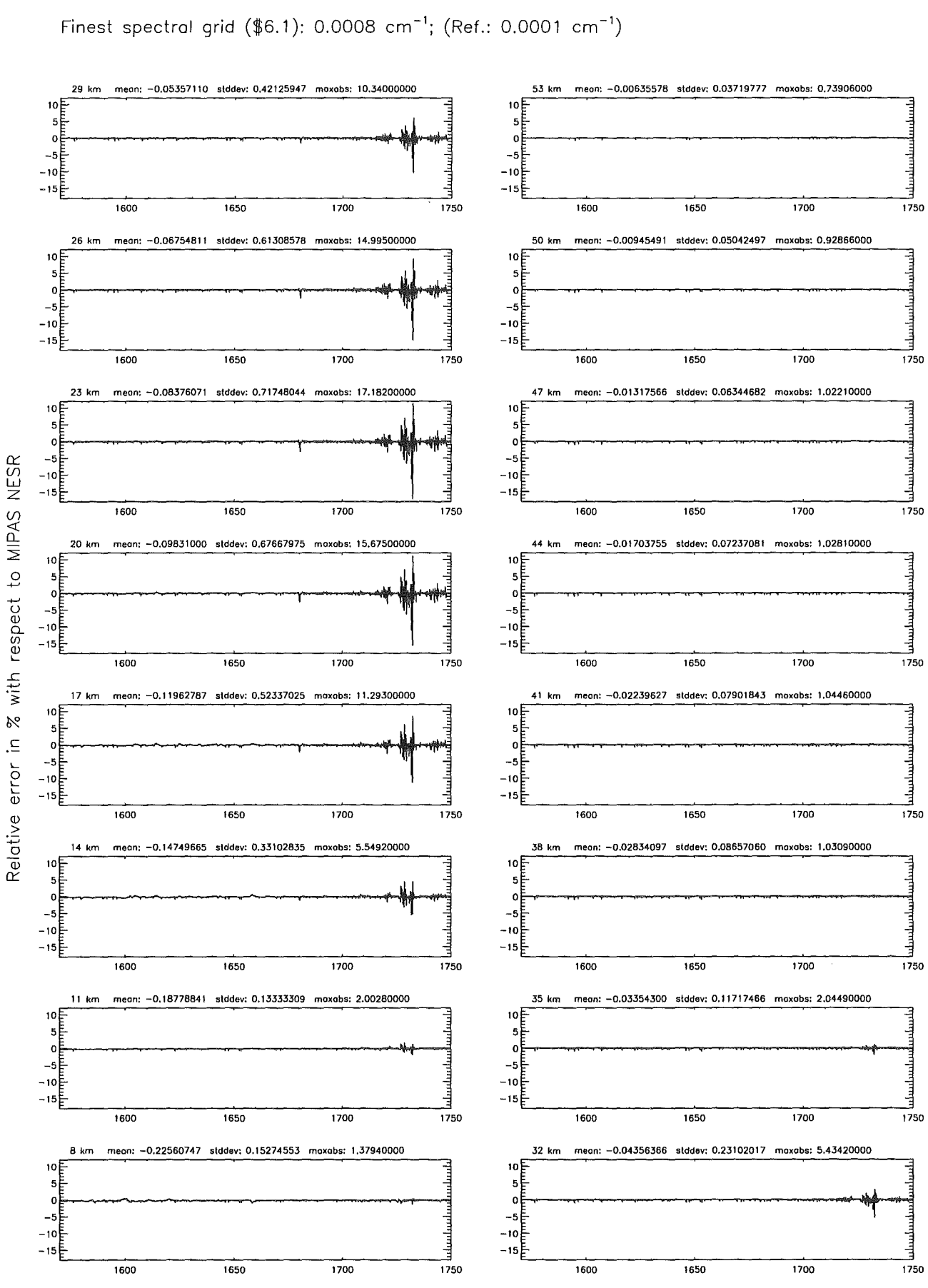


Figure 200: Relative error [%] with respect to MIPAS NESR

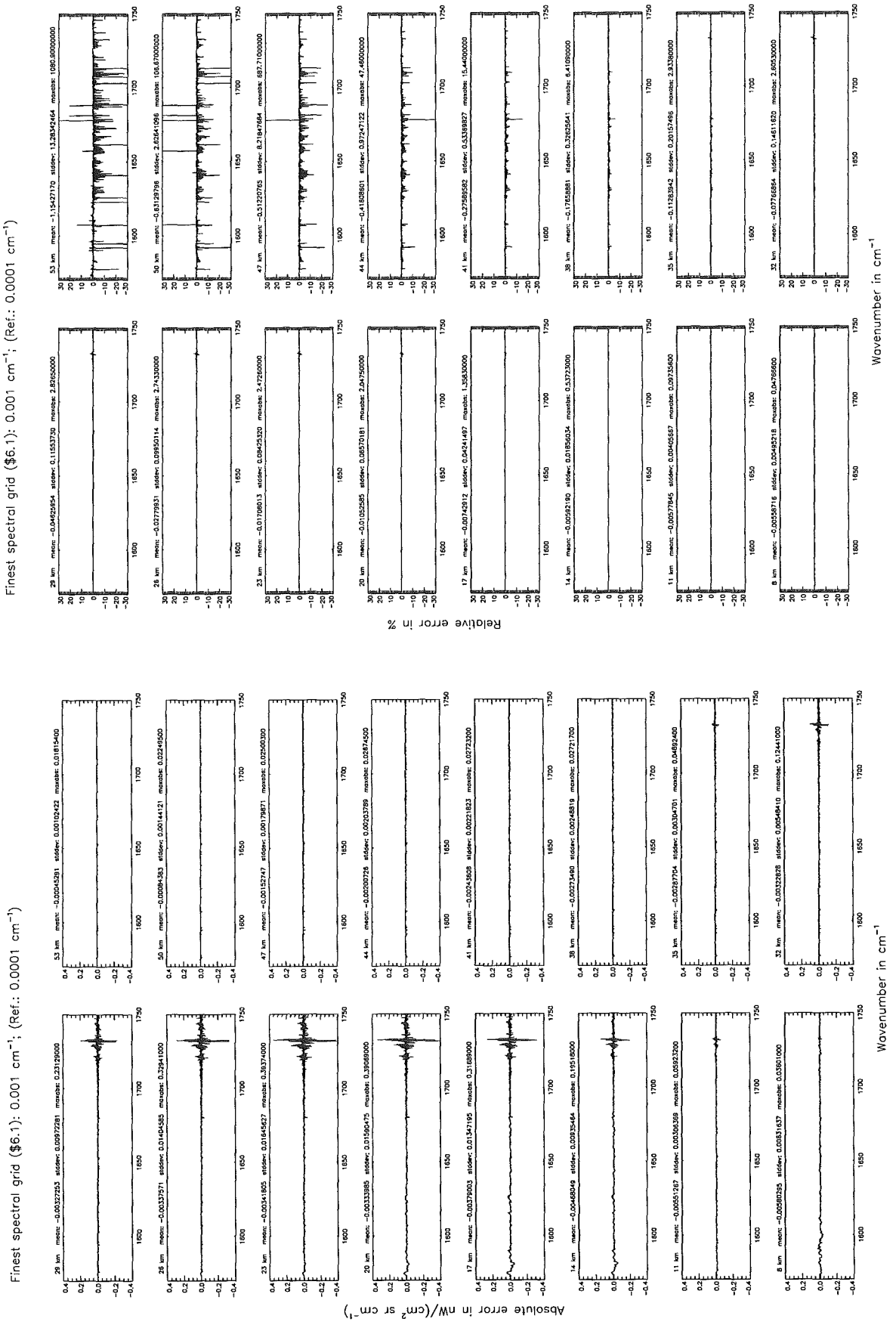


Figure 201: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

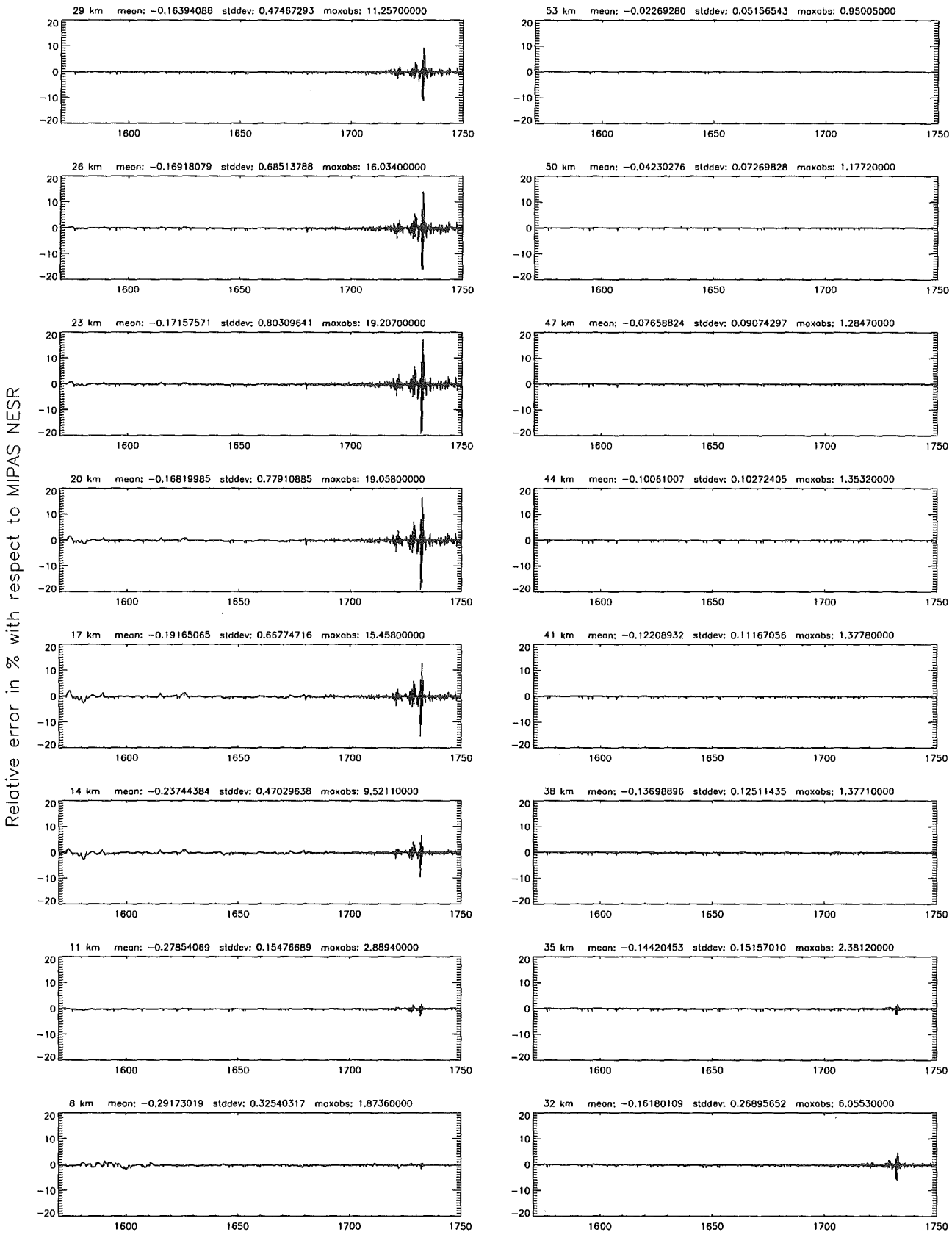


Figure 202: Relative error [%] with respect to MIPAS NESR

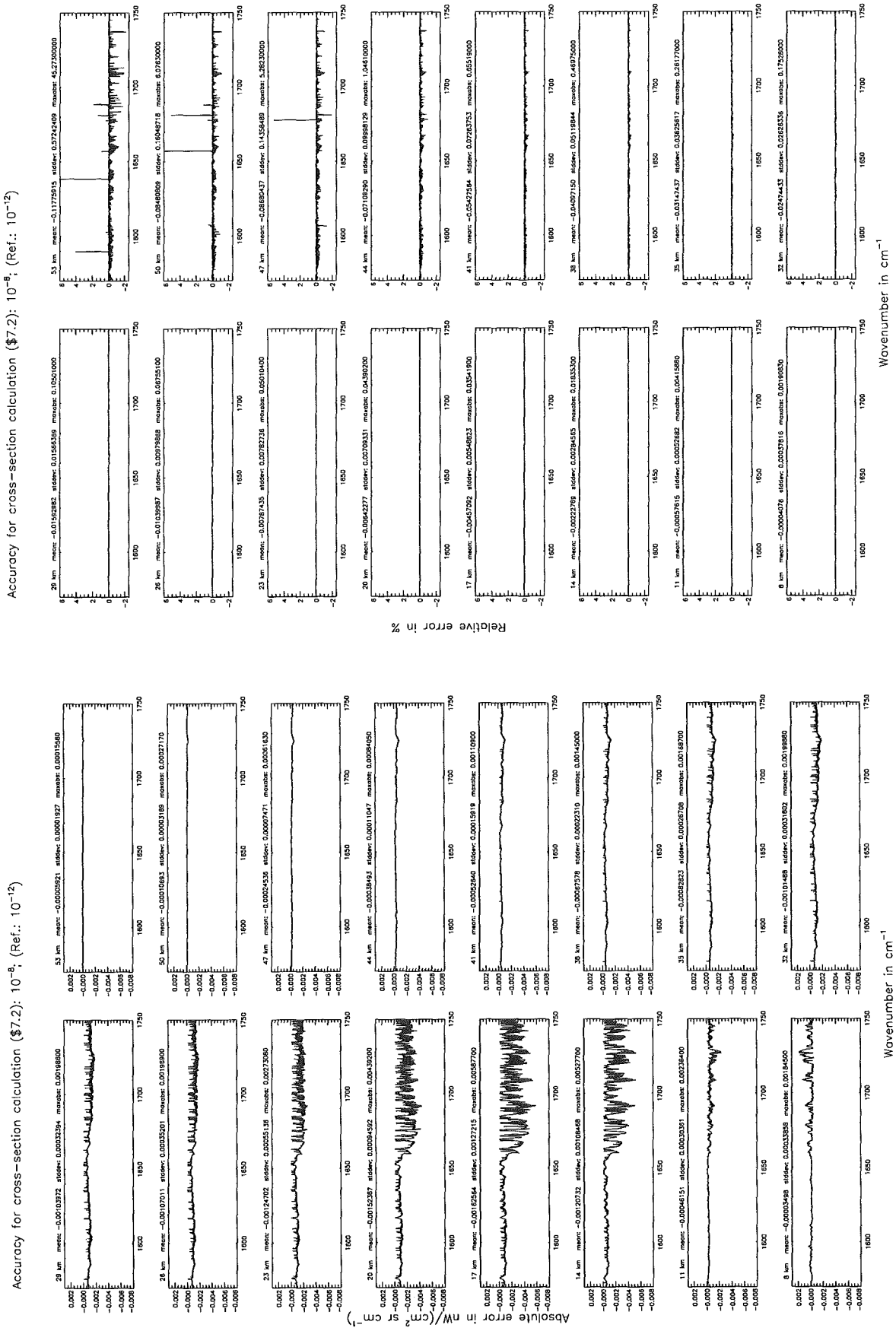


Figure 203: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

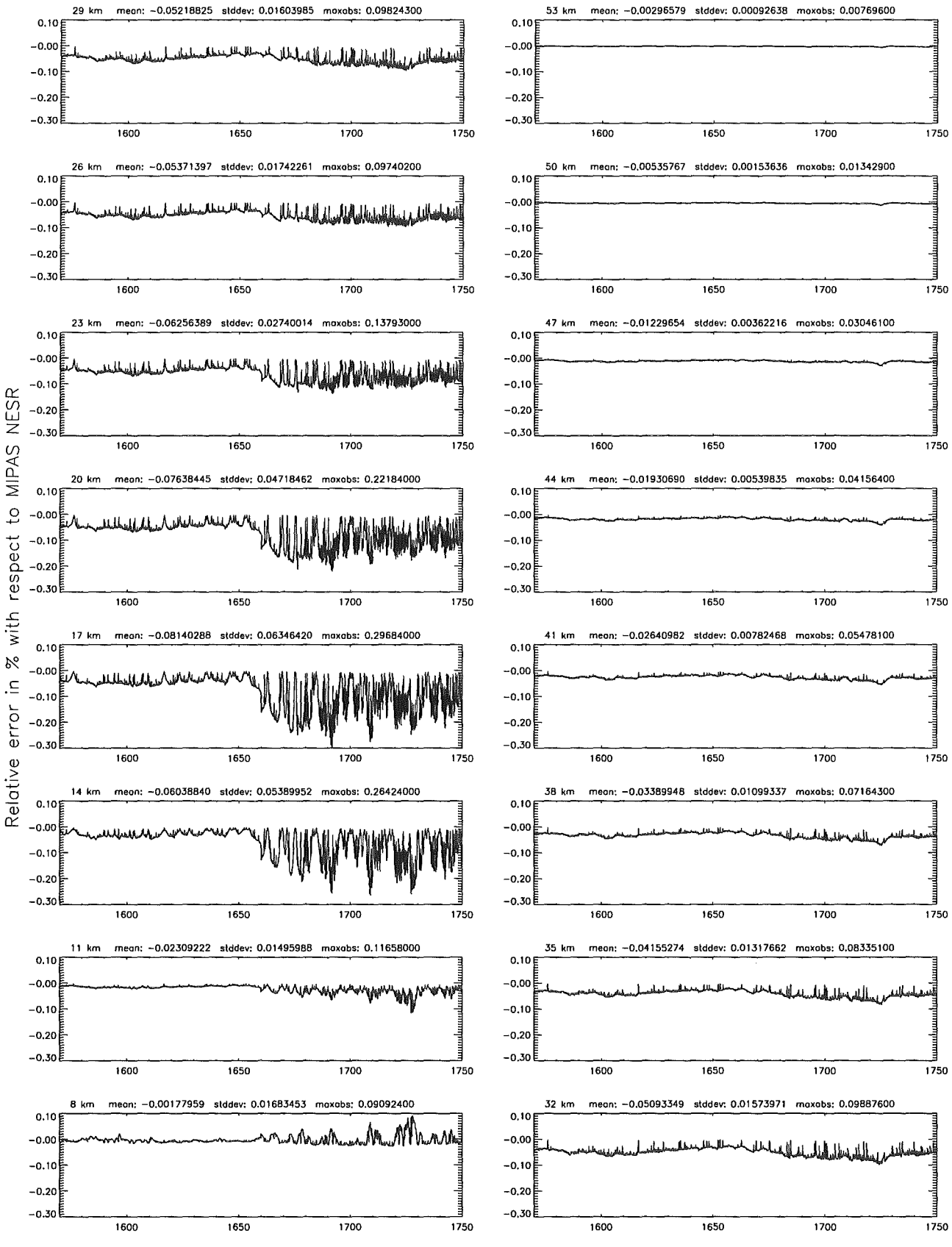


Figure 204: Relative error [%] with respect to MIPAS NESR

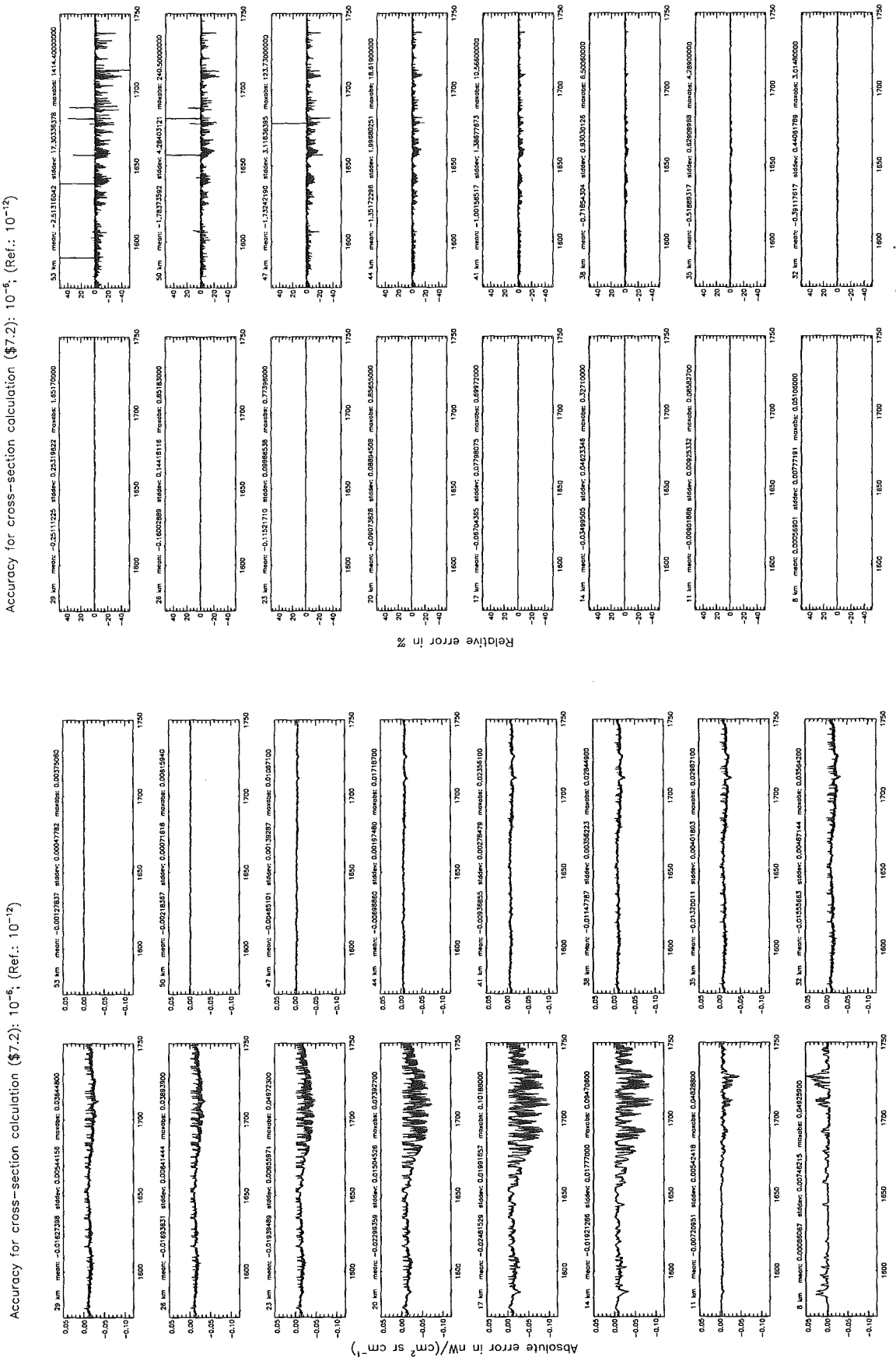


Figure 205: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

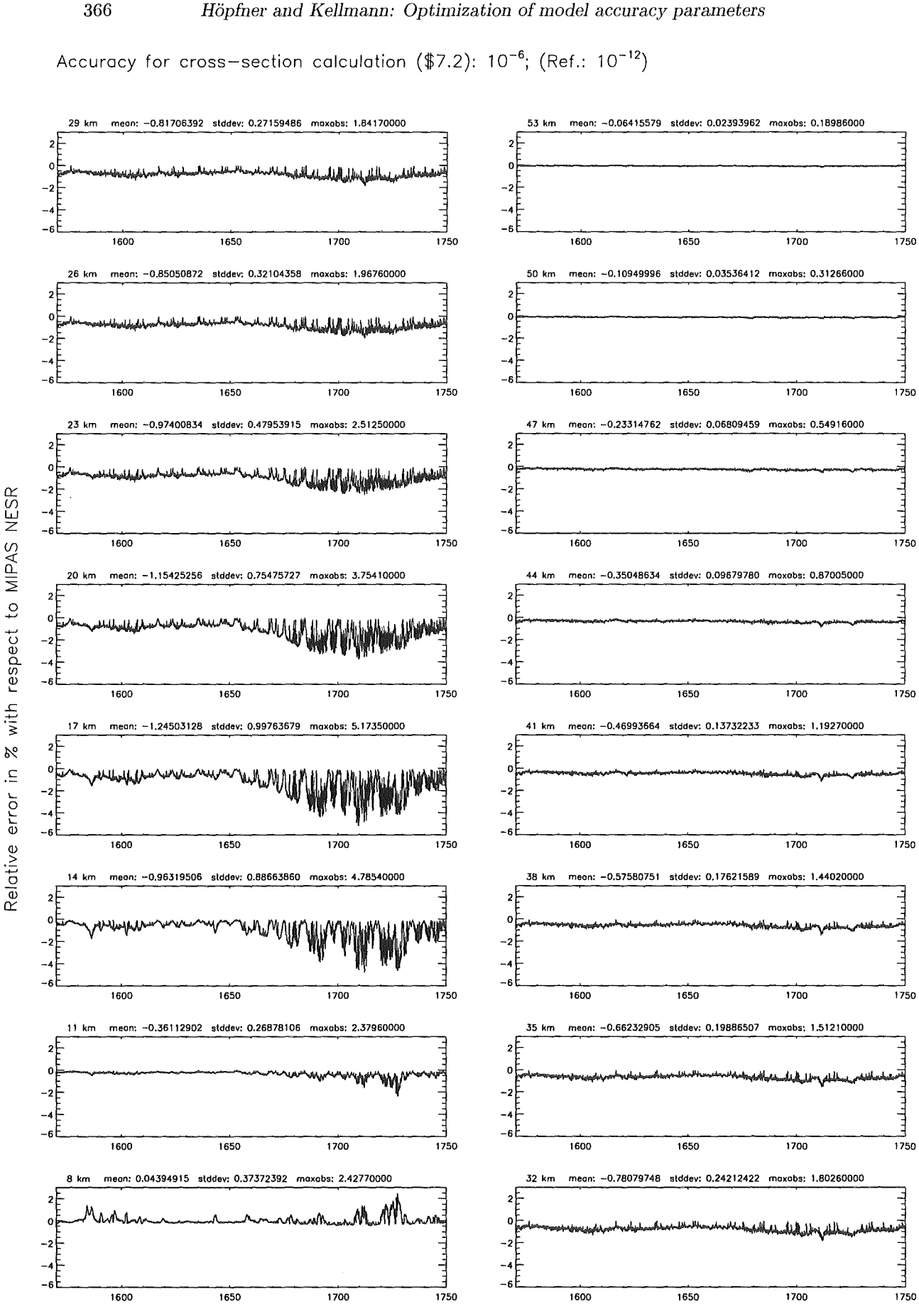


Figure 206: Relative error [%] with respect to MIPAS NESR

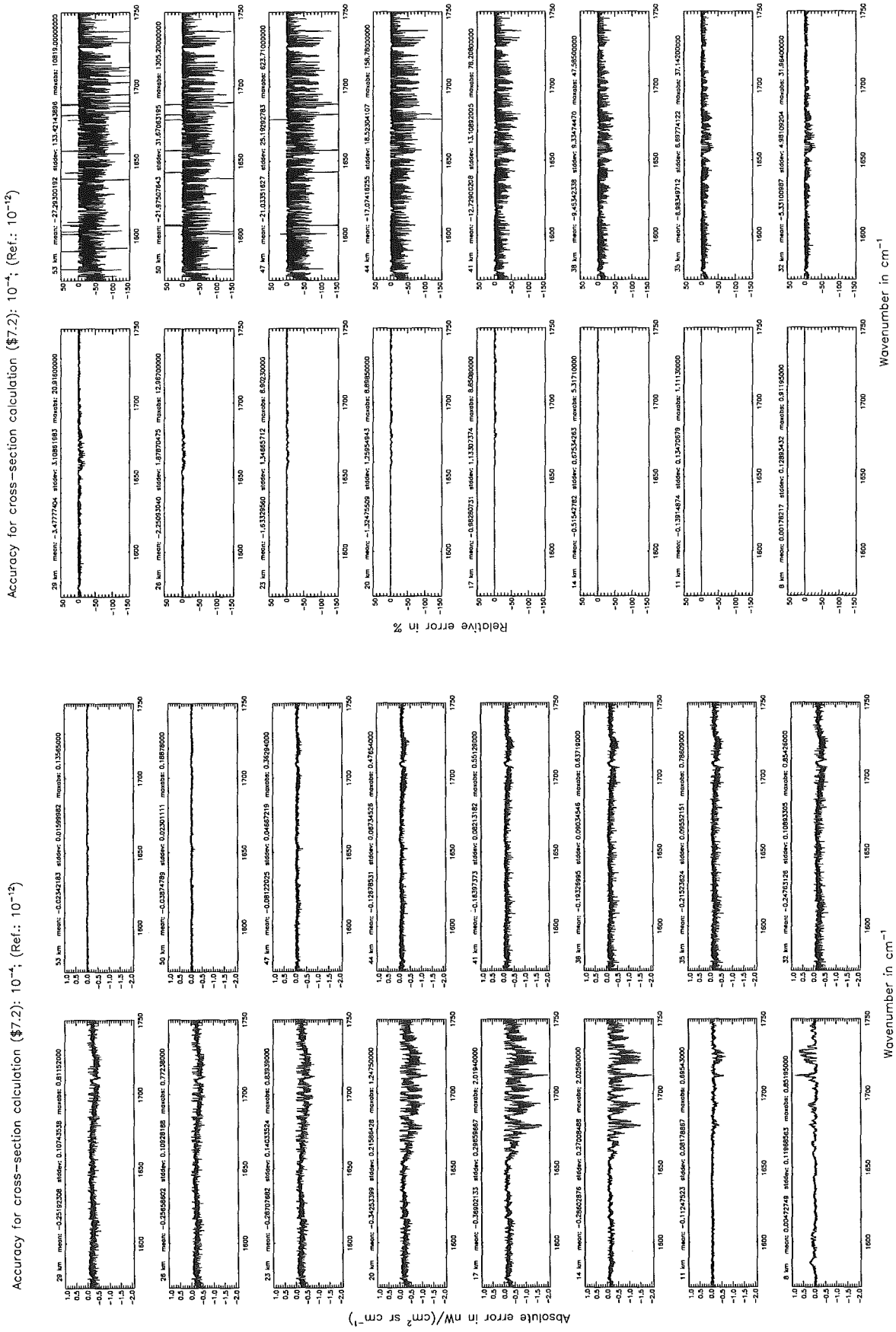


Figure 207: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

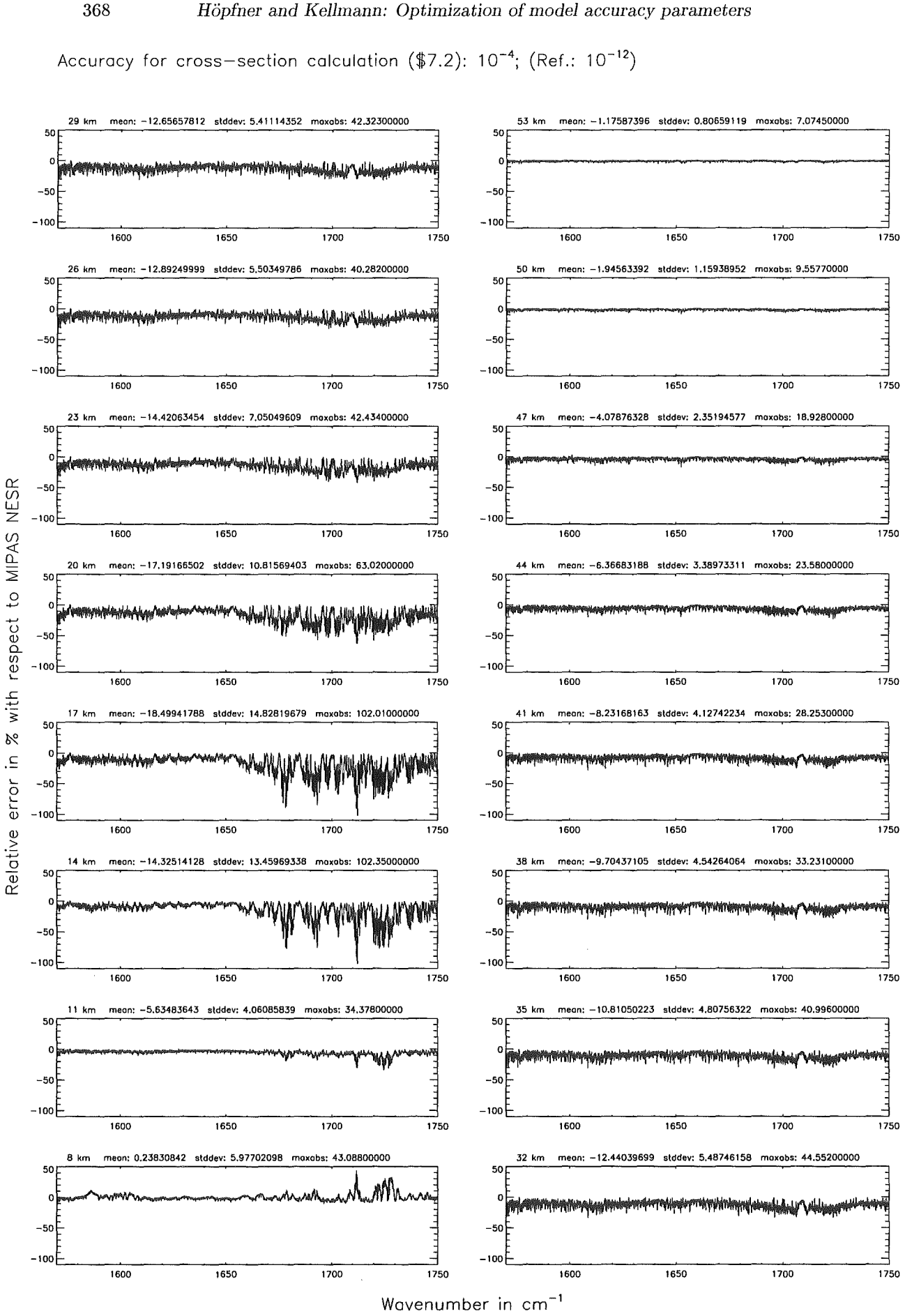


Figure 208: Relative error [%] with respect to MIPAS NESR

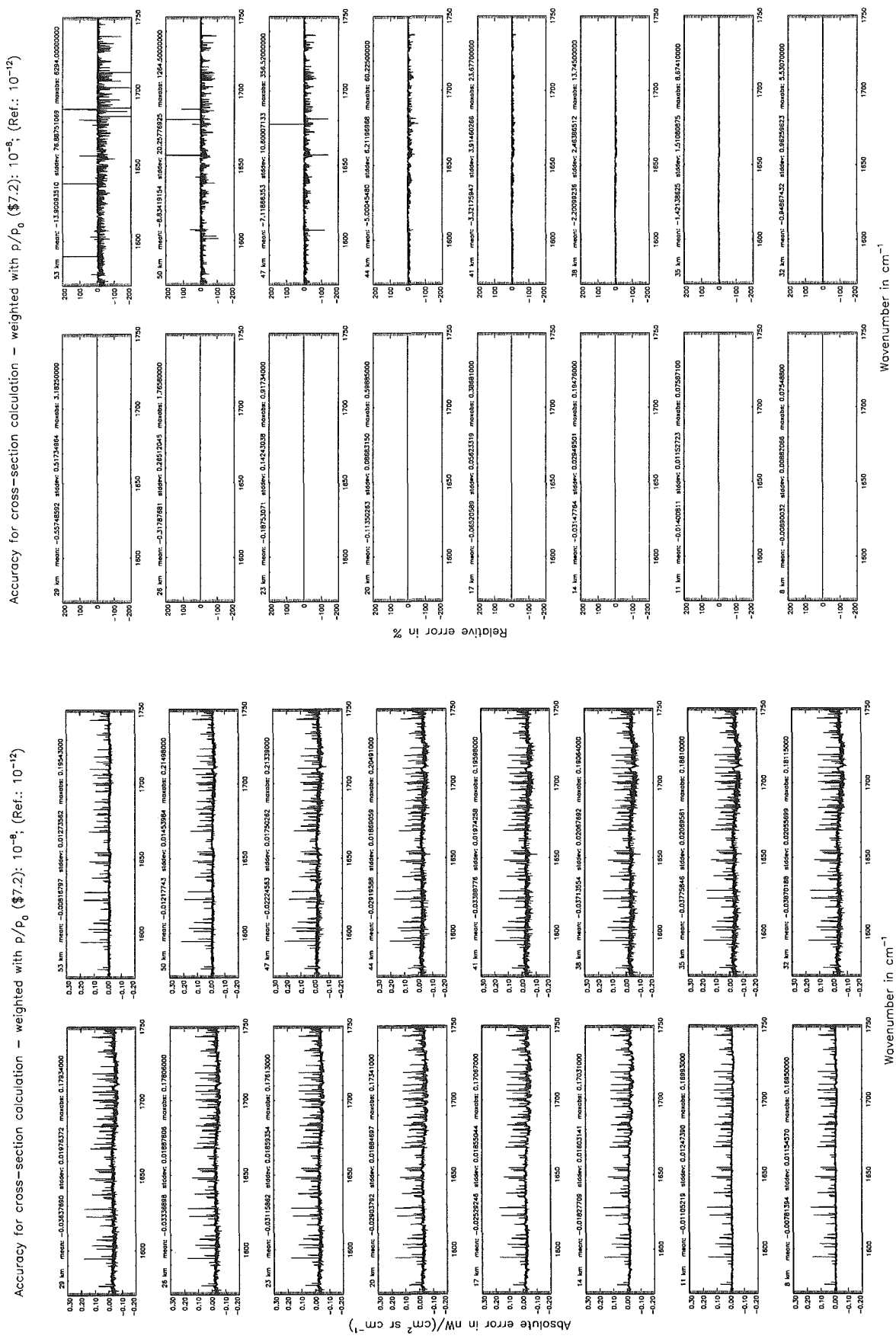


Figure 209: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

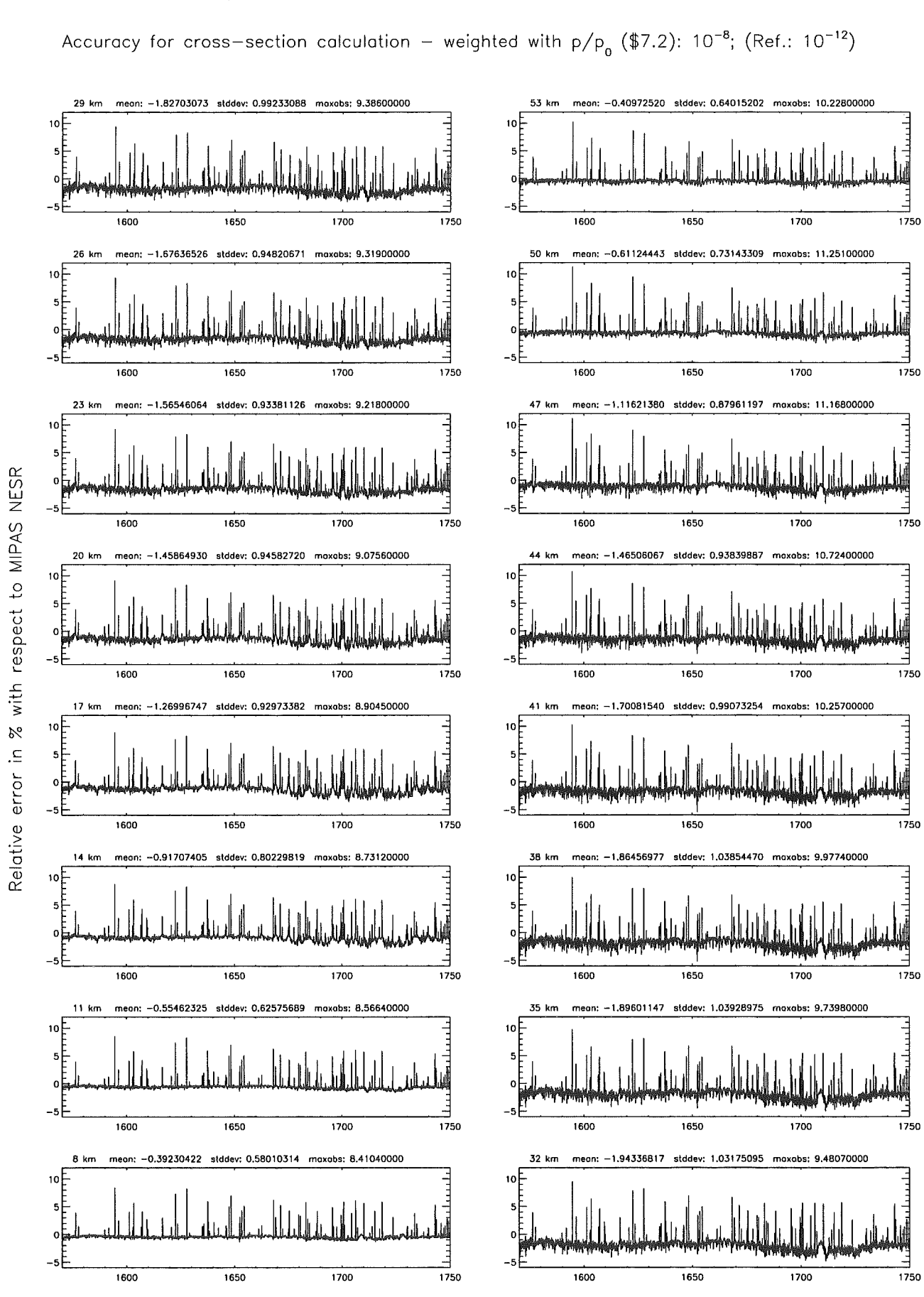


Figure 210: Relative error [%] with respect to MIPAS NESR

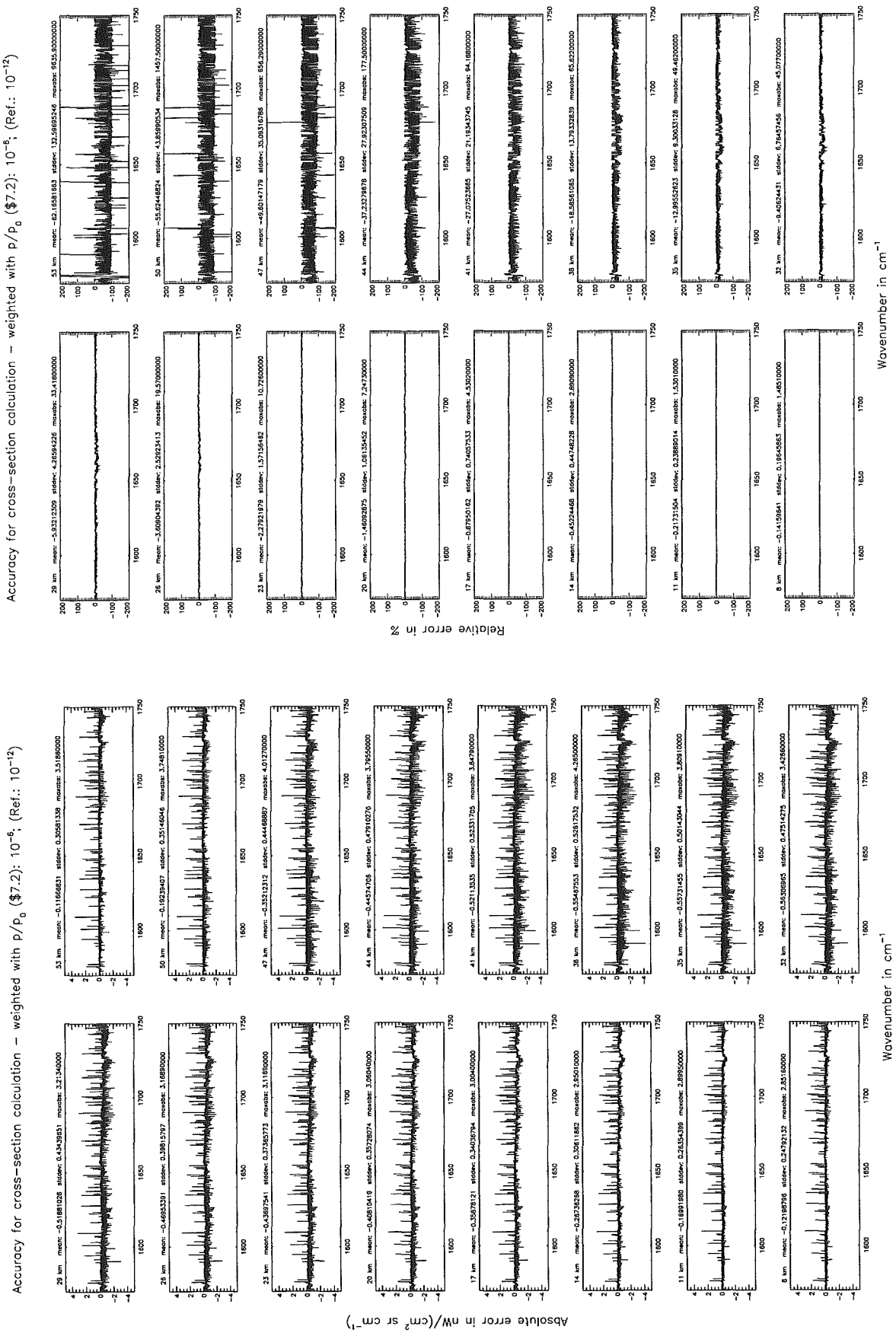


Figure 211: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

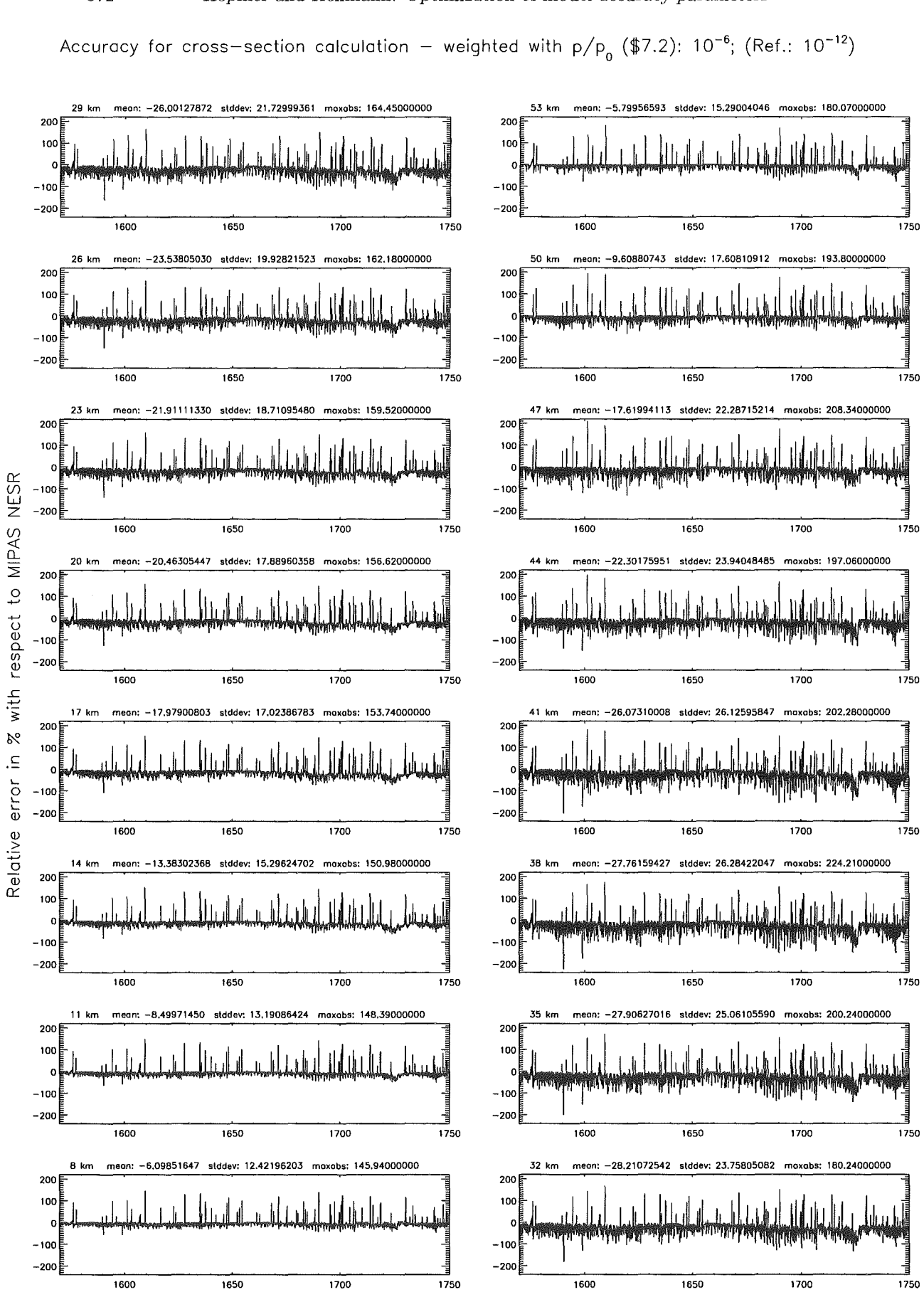
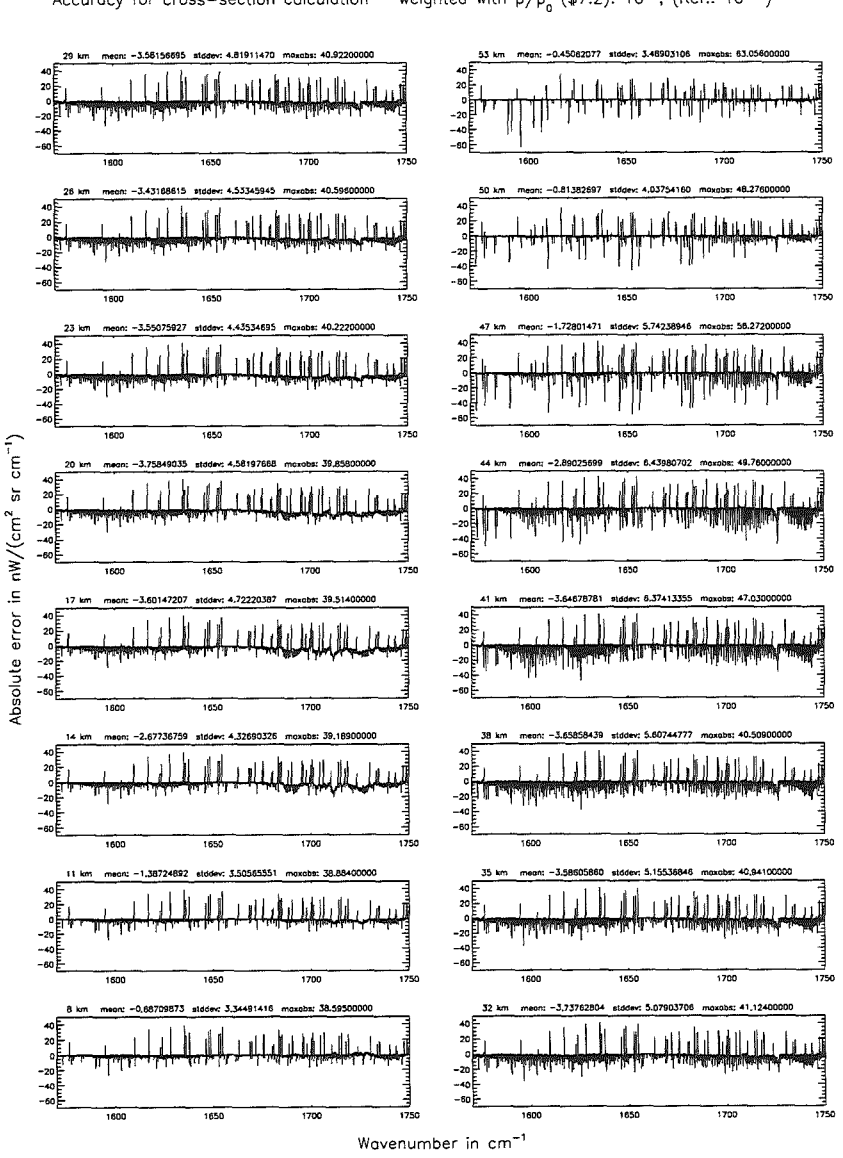


Figure 212: Relative error [%] with respect to MIPAS NESR



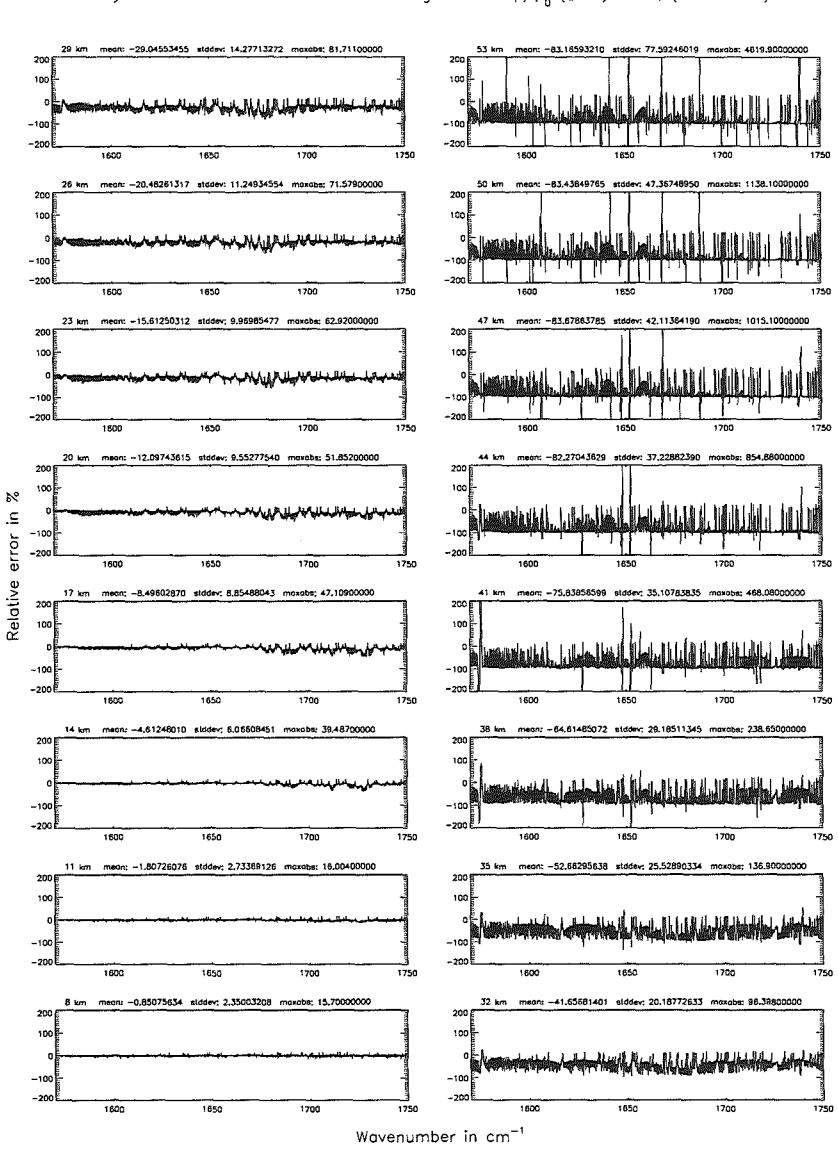


Figure 213: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

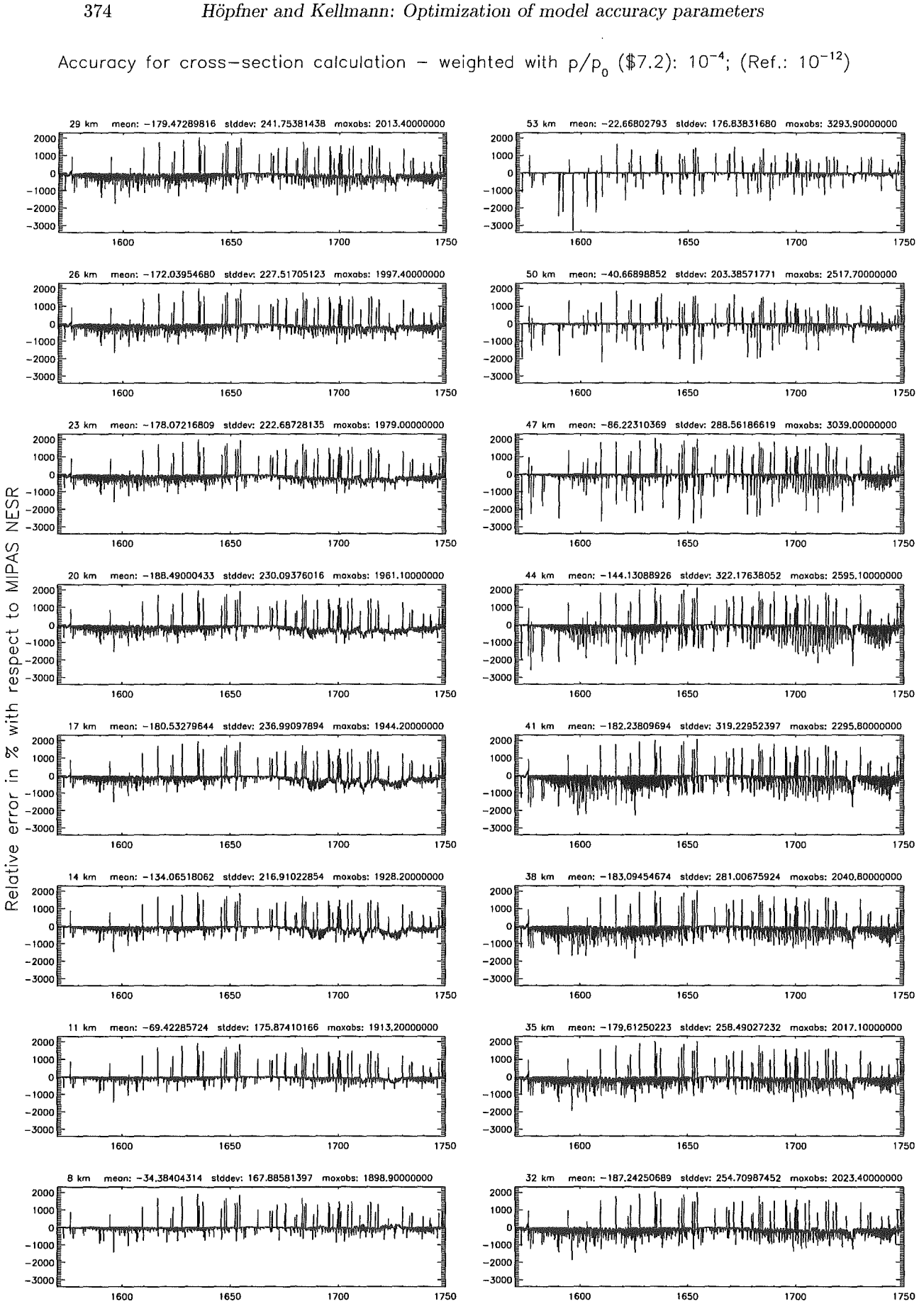
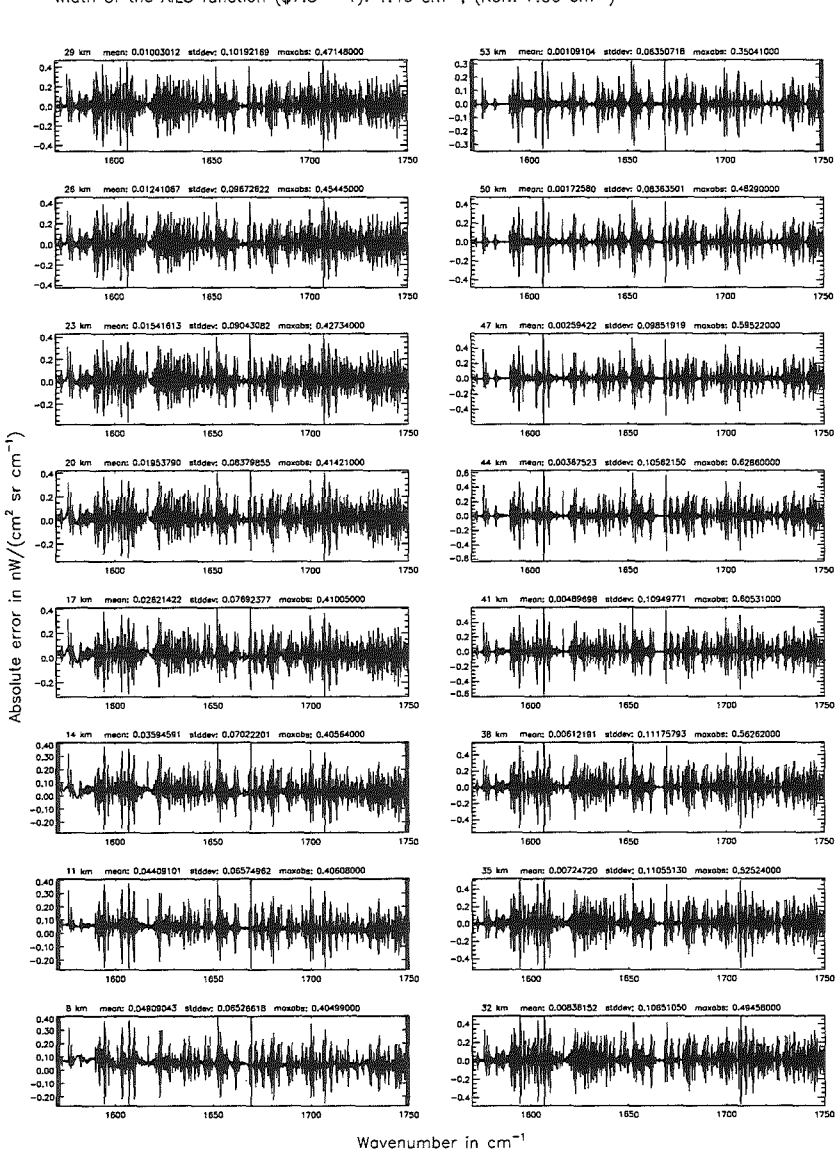


Figure 214: Relative error [%] with respect to MIPAS NESR



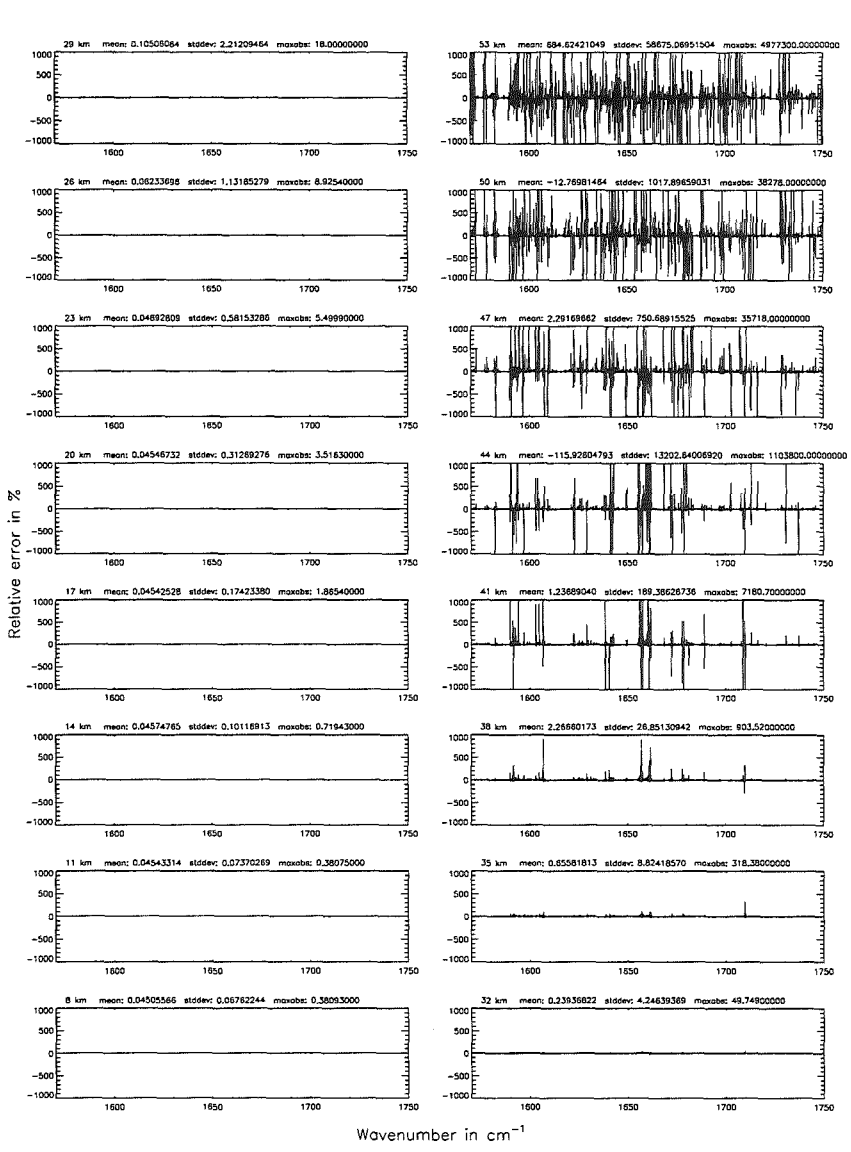


Figure 215: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

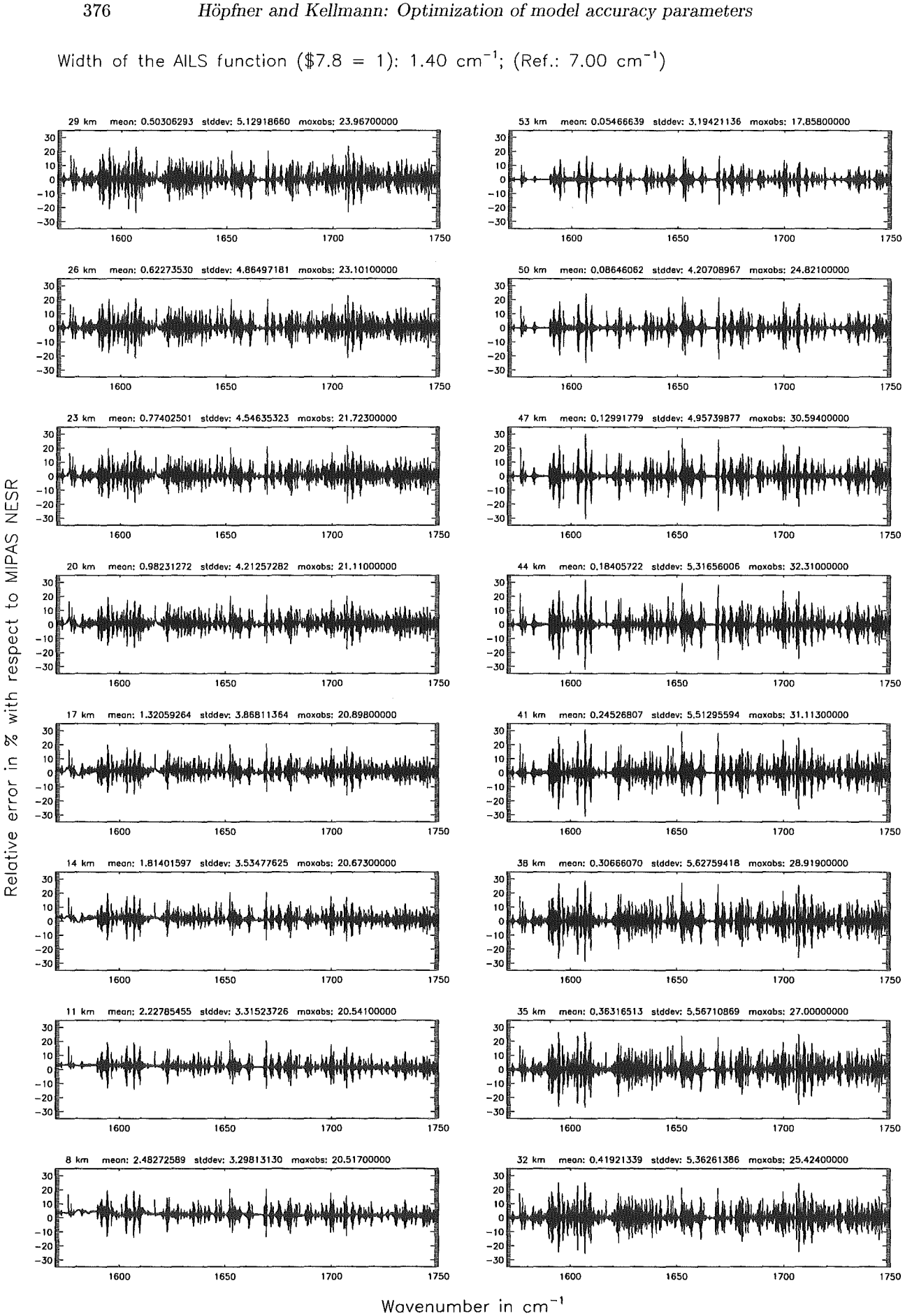


Figure 216: Relative error [%] with respect to MIPAS NESR

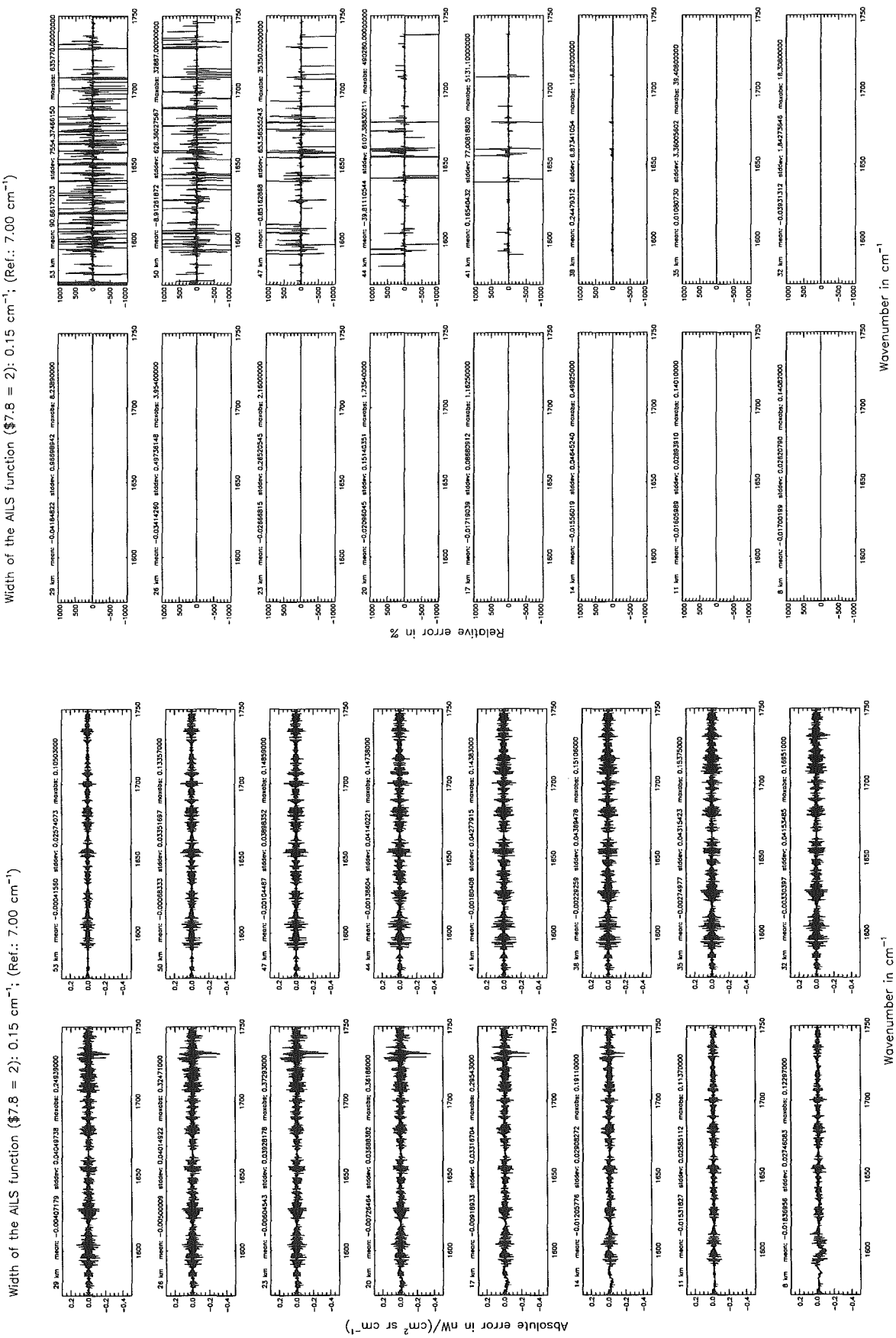


Figure 217: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

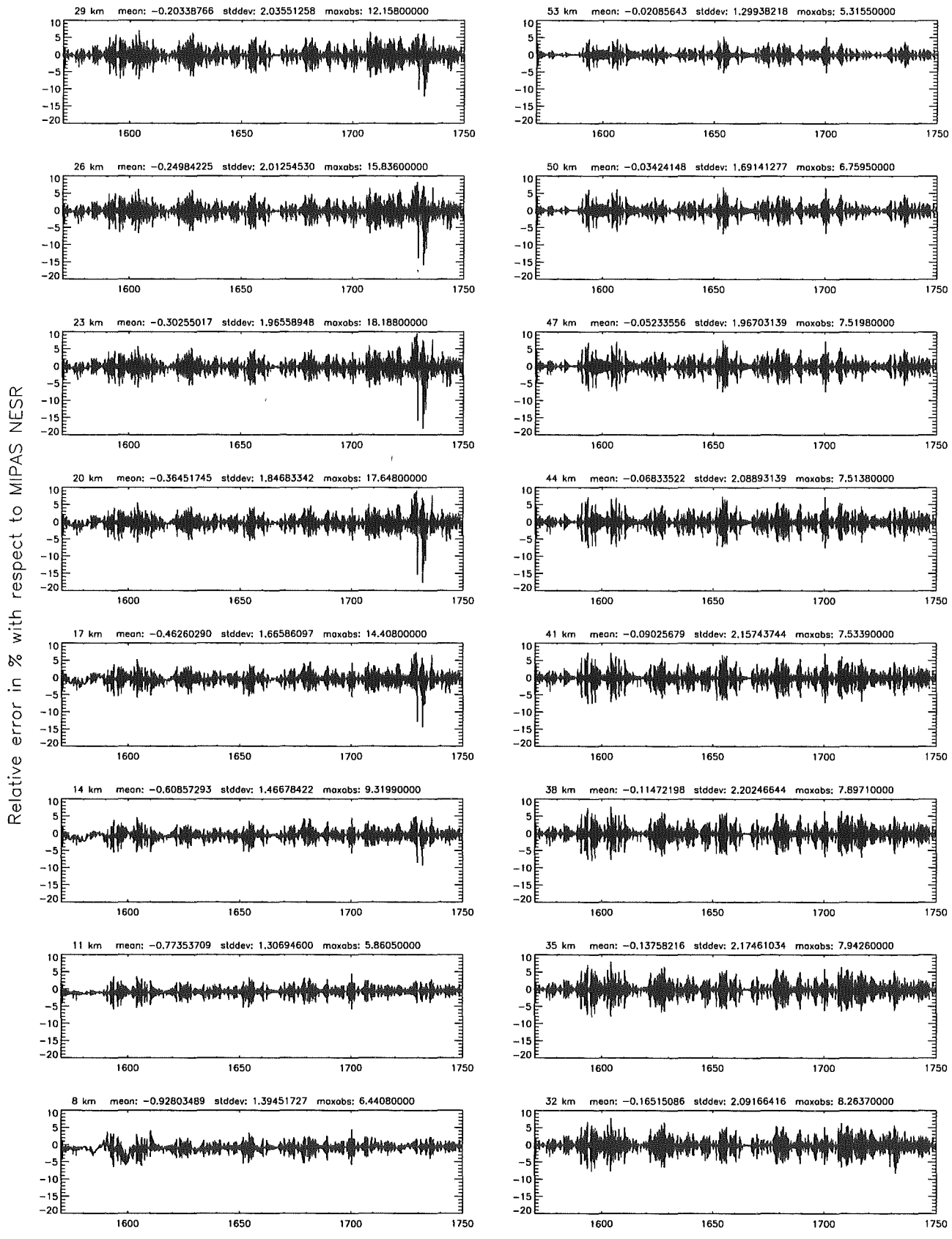
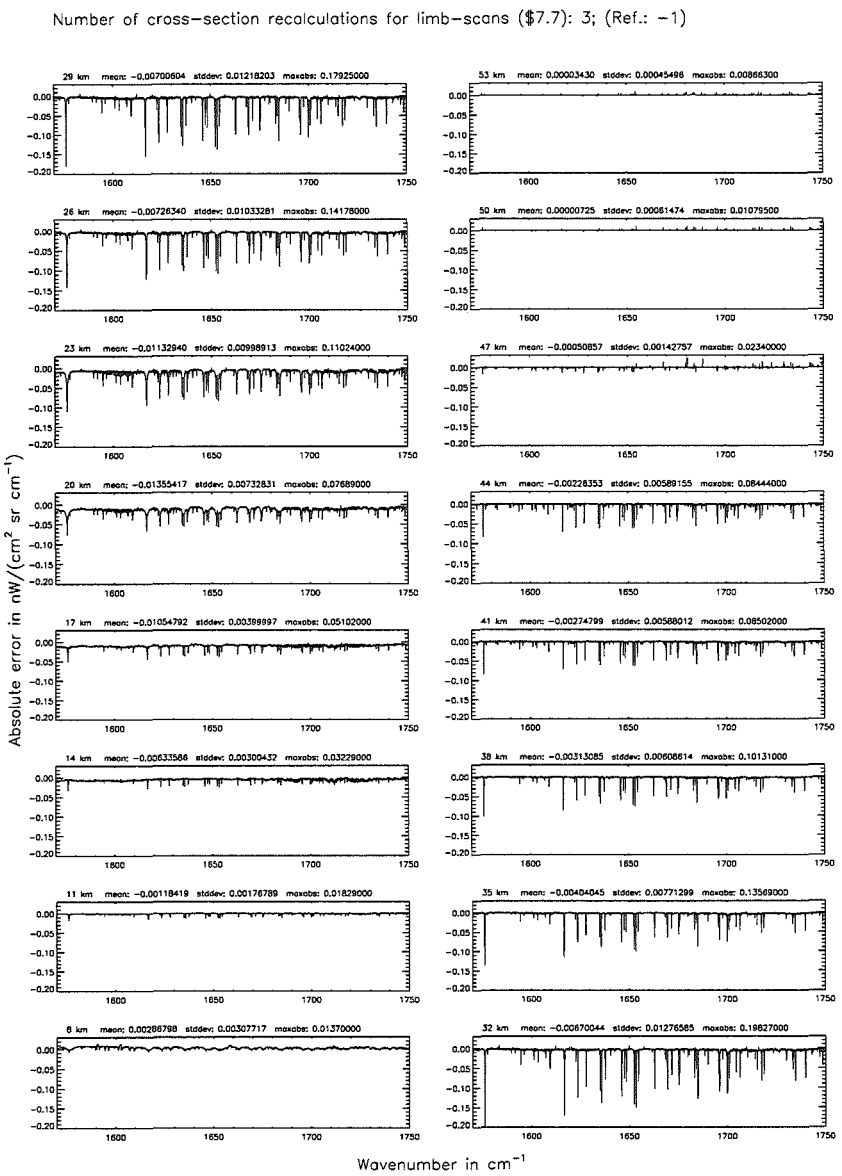


Figure 218: Relative error [%] with respect to MIPAS NESR



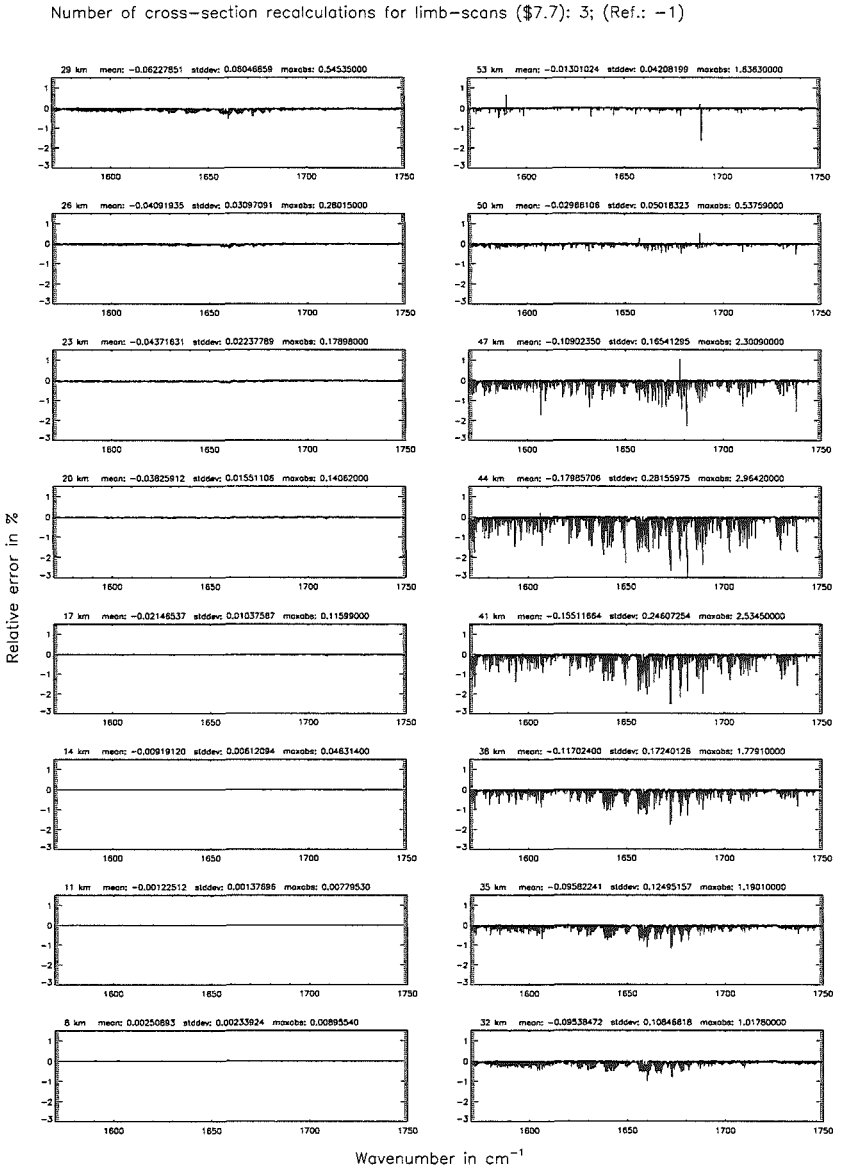


Figure 219: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

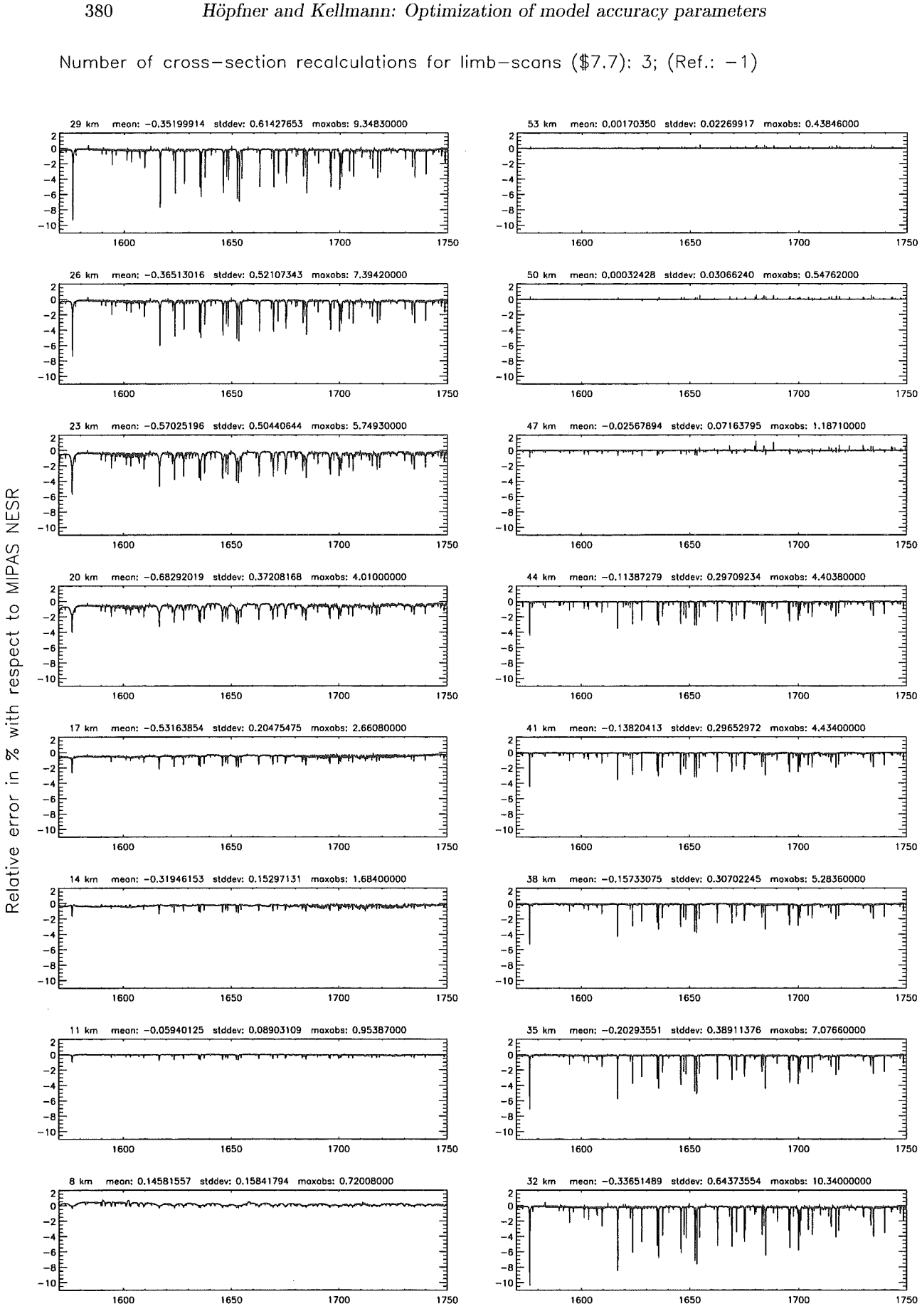
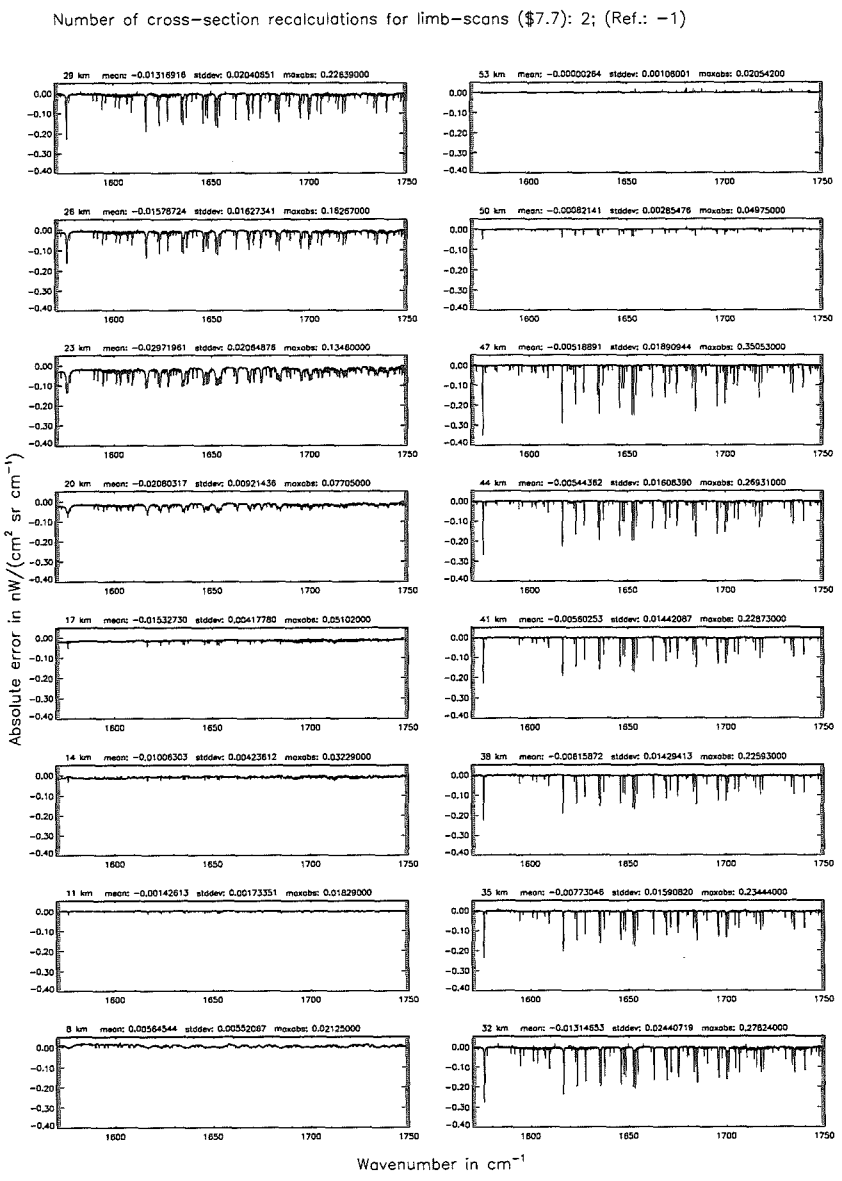


Figure 220: Relative error [%] with respect to MIPAS NESR



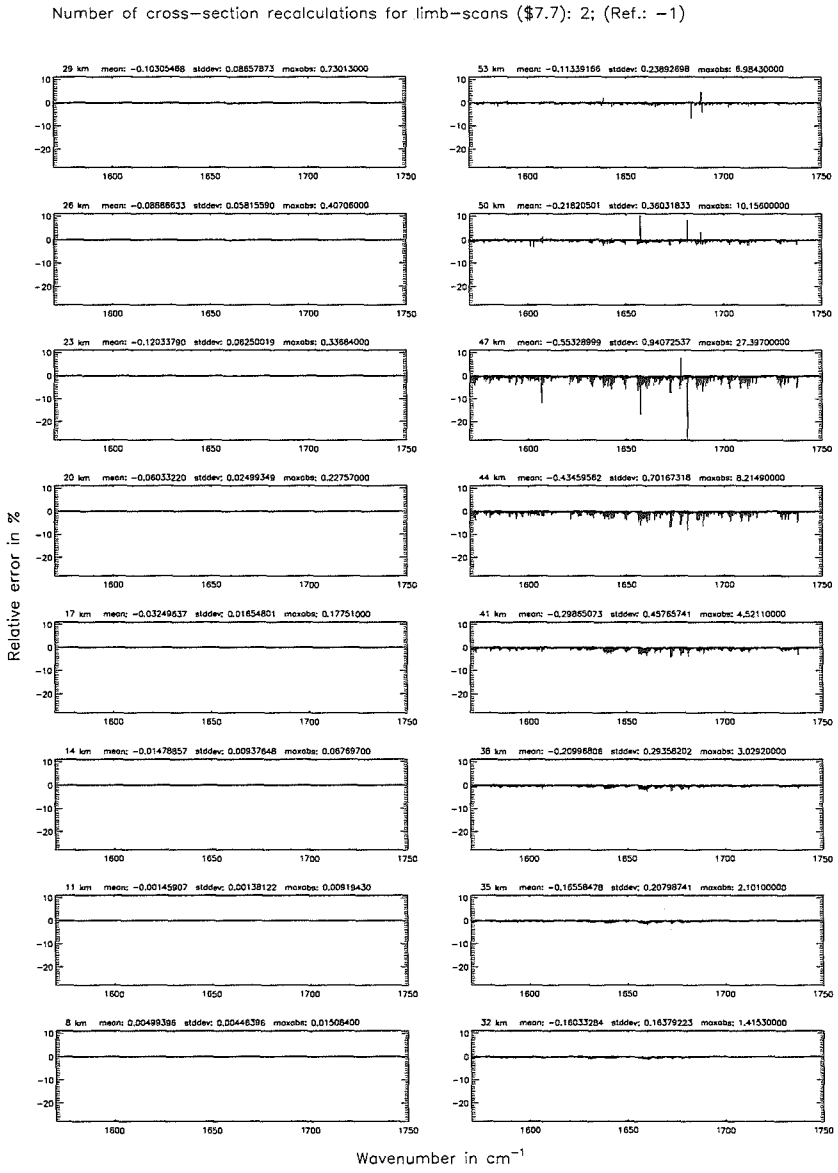


Figure 221: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

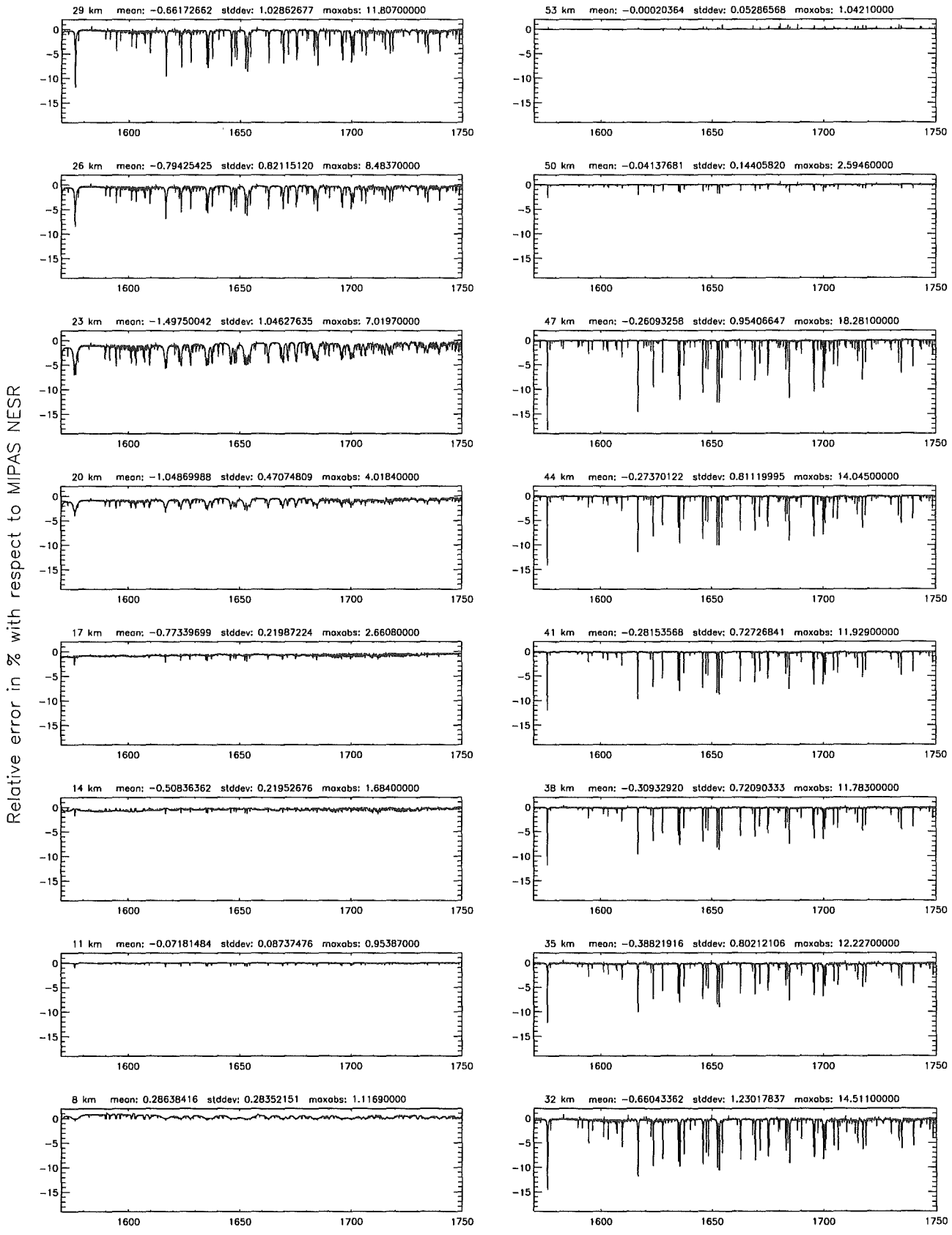


Figure 222: Relative error [%] with respect to MIPAS NESR

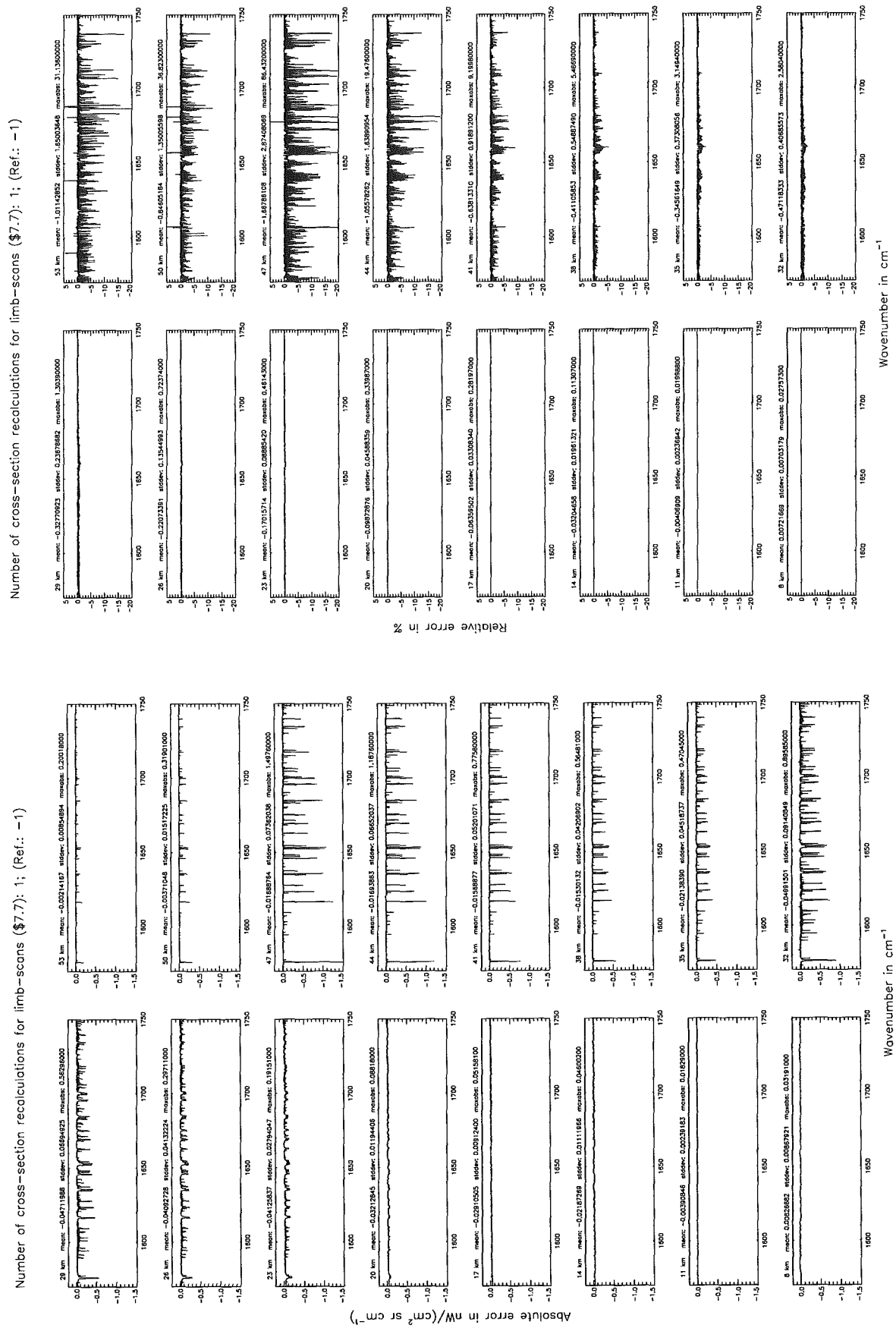


Figure 223: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

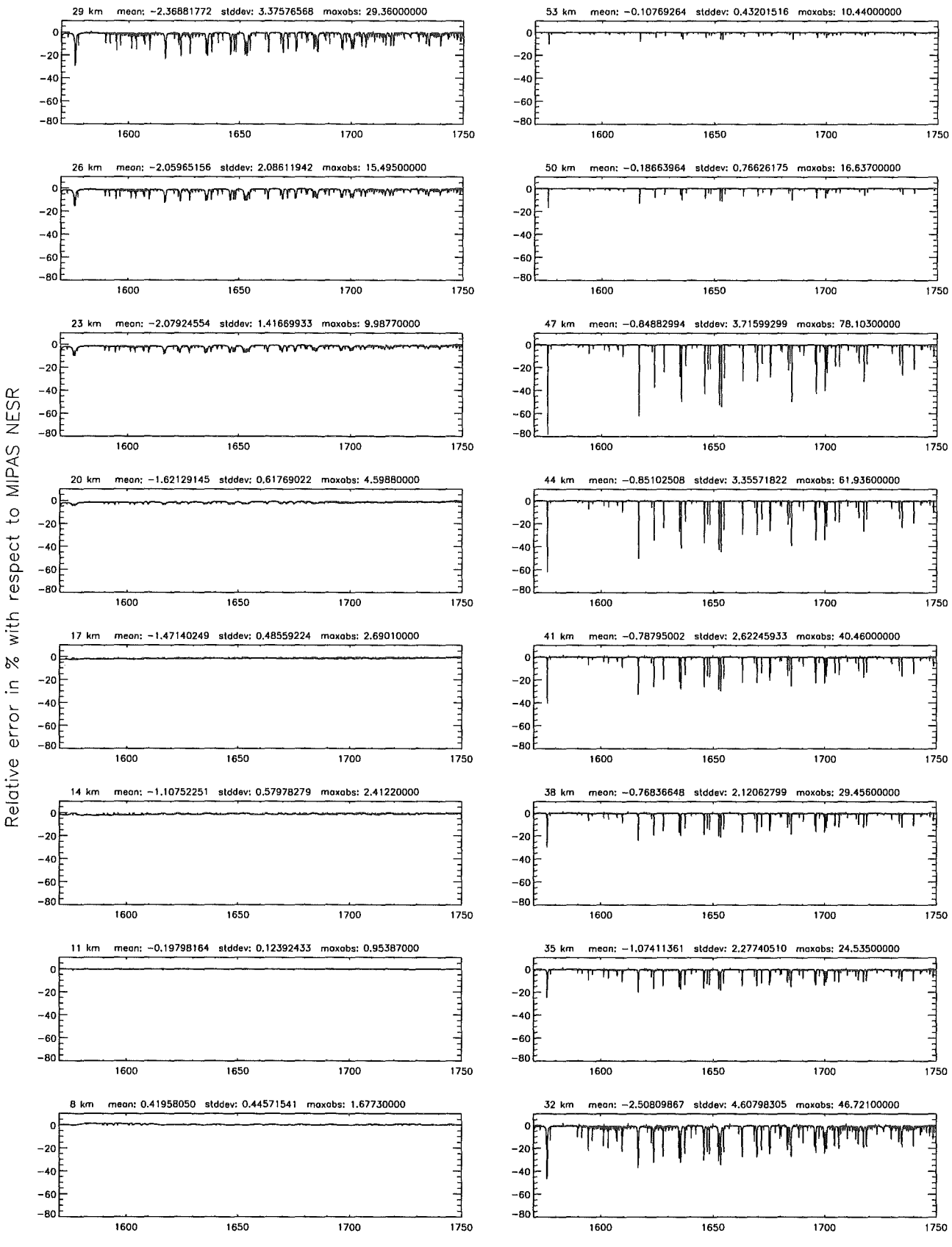


Figure 224: Relative error [%] with respect to MIPAS NESR

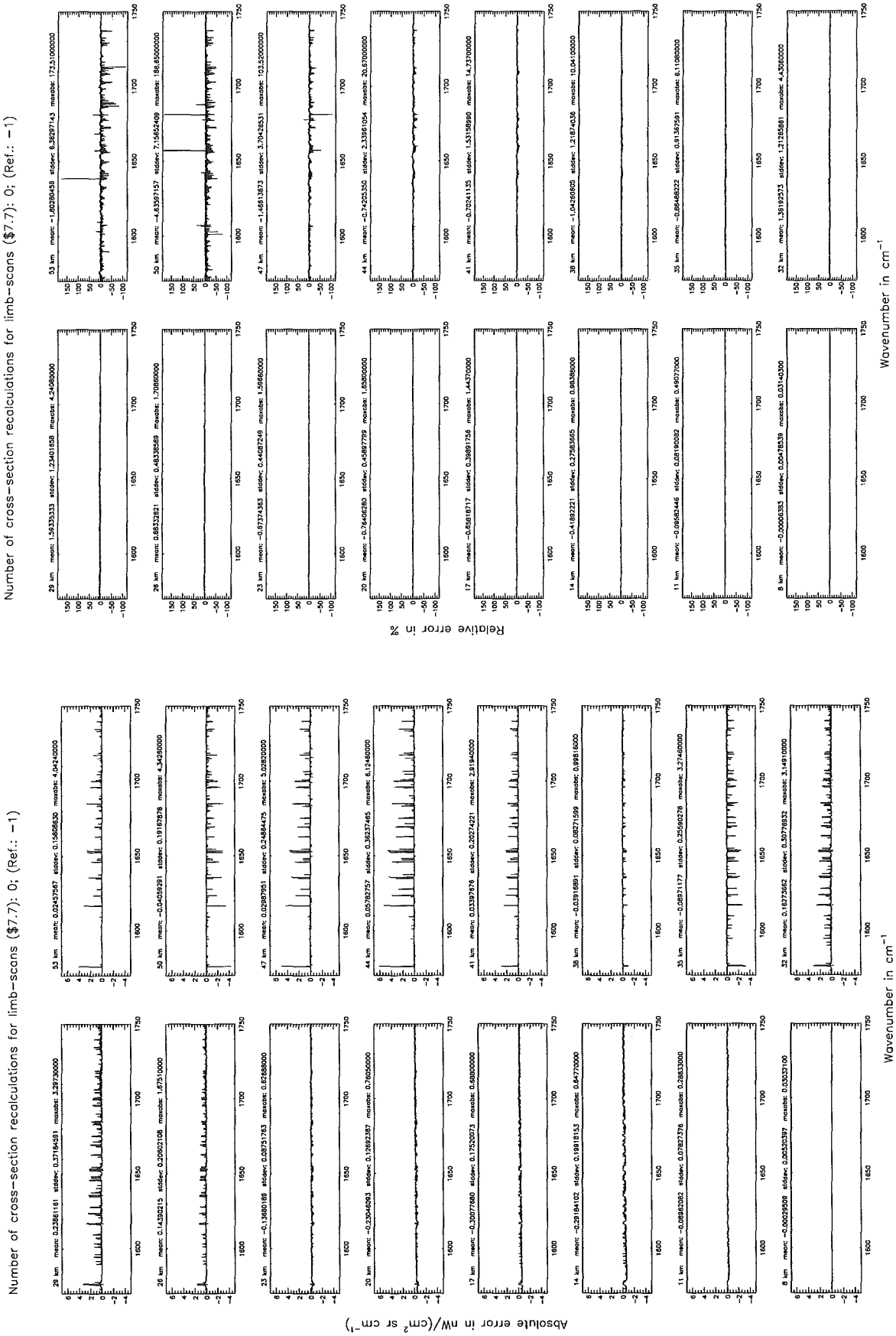


Figure 225: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

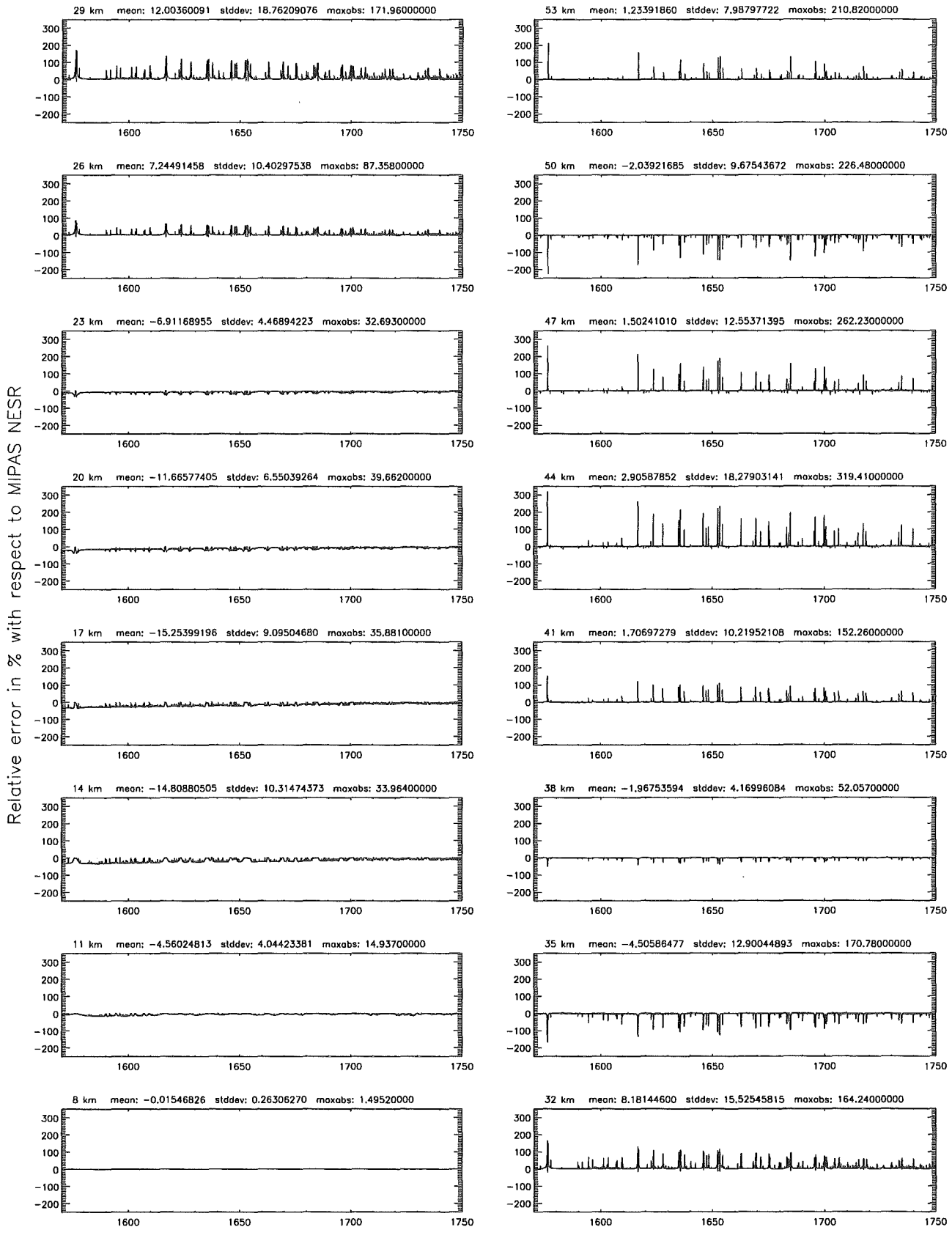
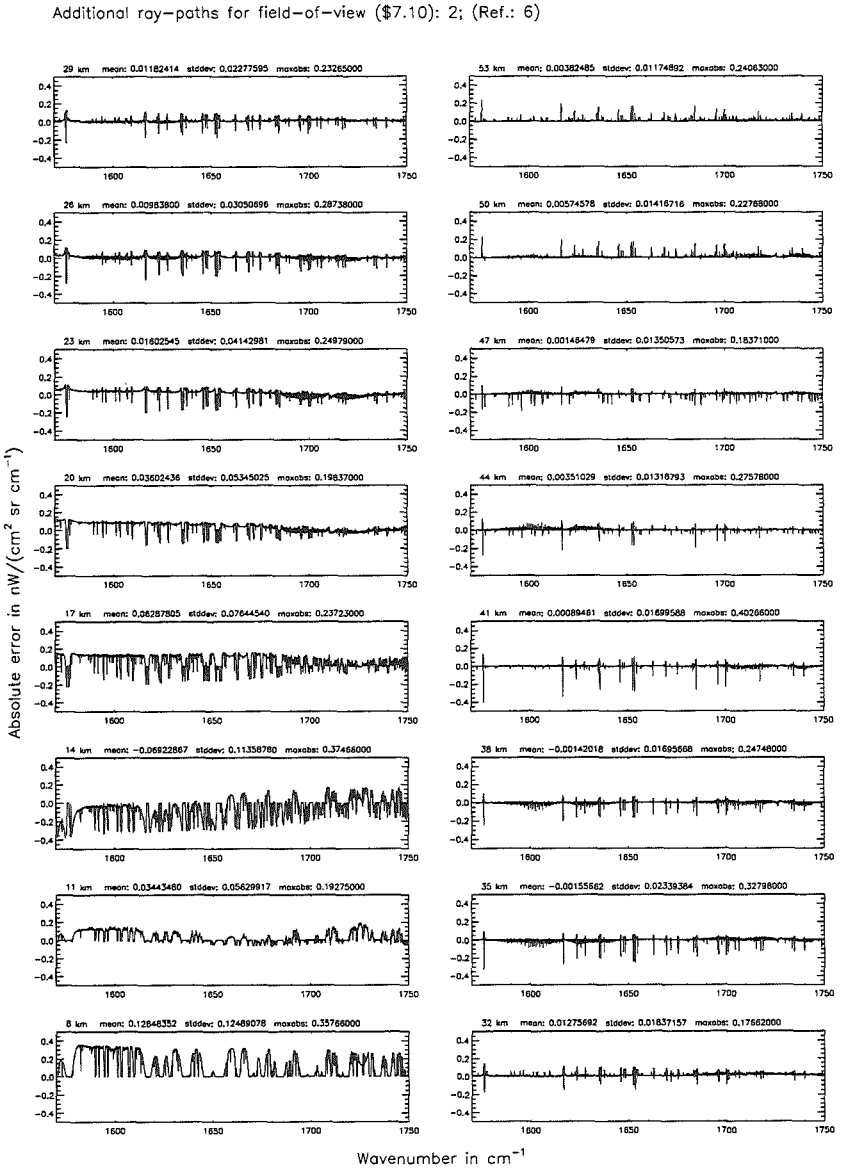


Figure 226: Relative error [%] with respect to MIPAS NESR



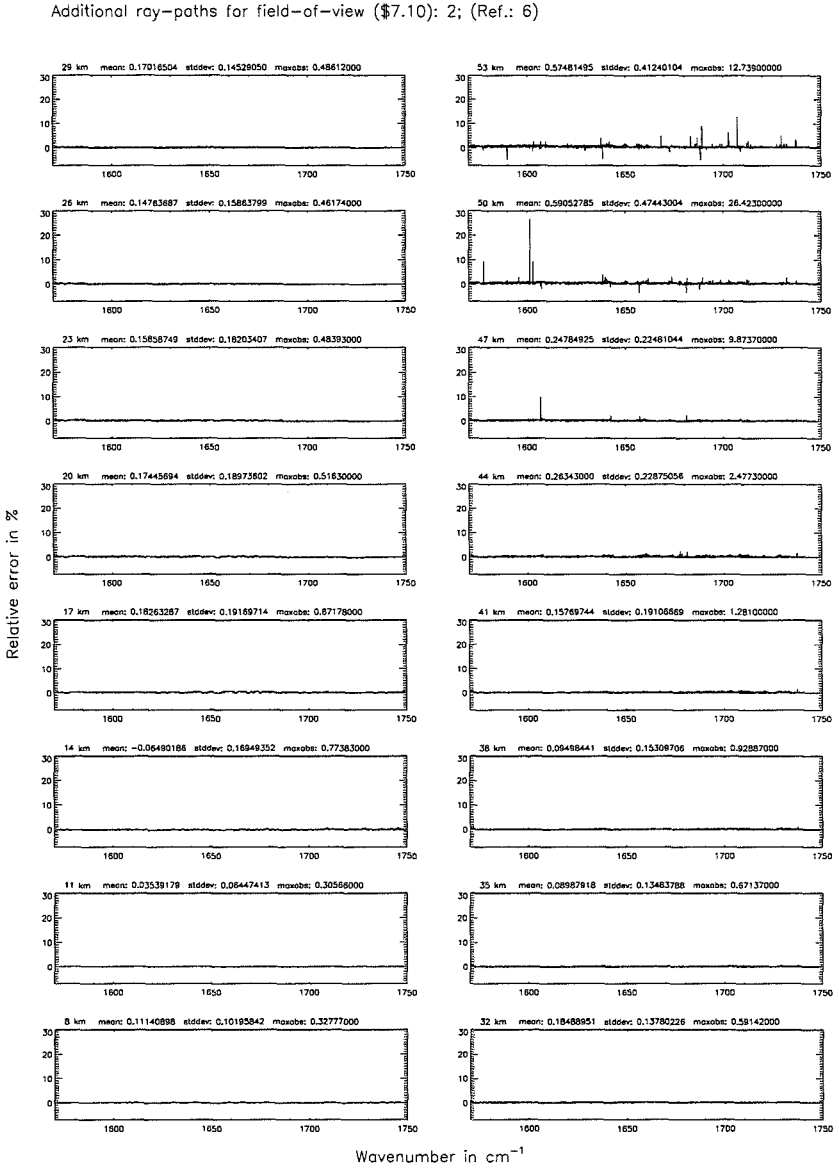


Figure 227: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Additional ray-paths for field-of-view ($7.10): 2; (Ref.: 6)

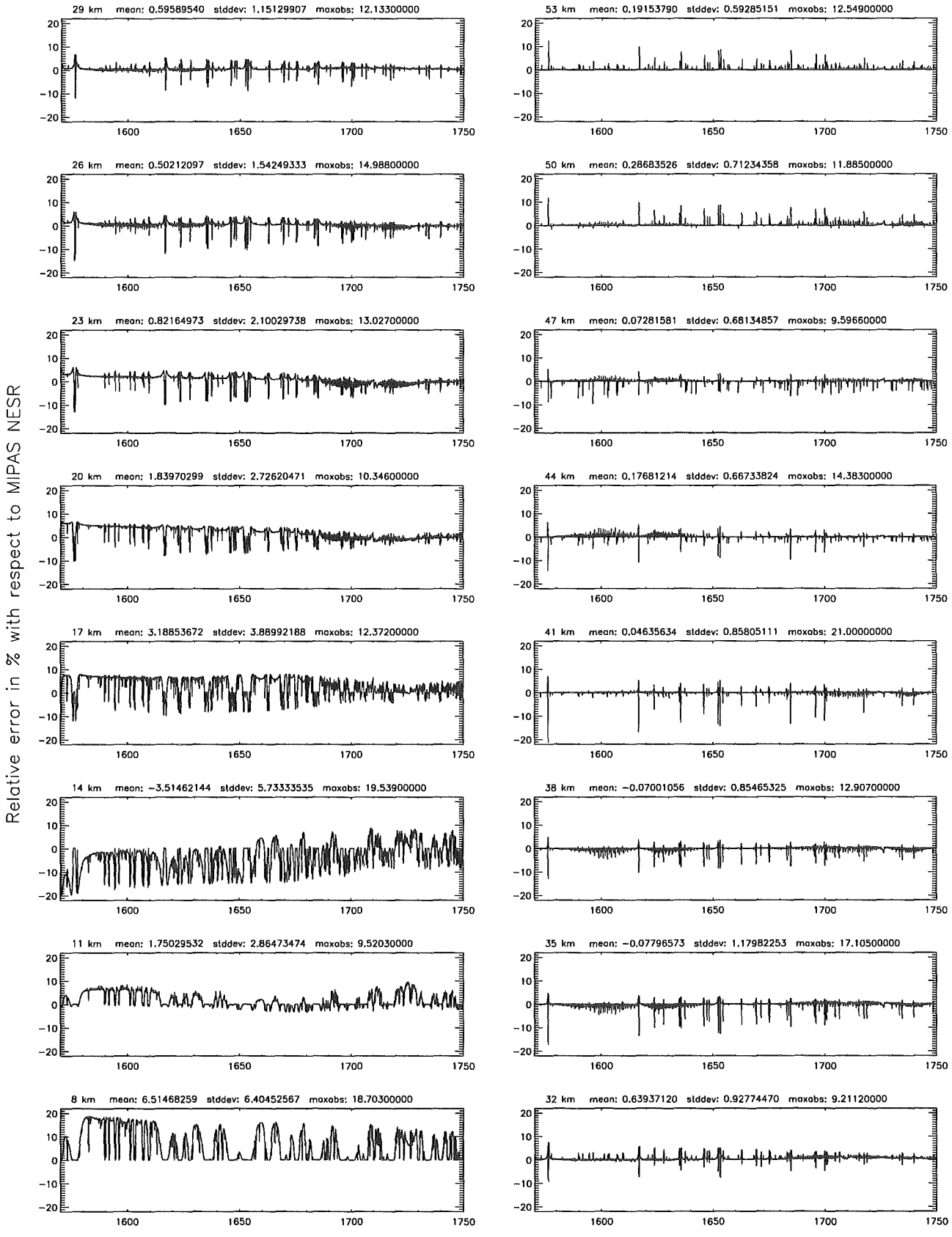


Figure 228: Relative error [%] with respect to MIPAS NESR

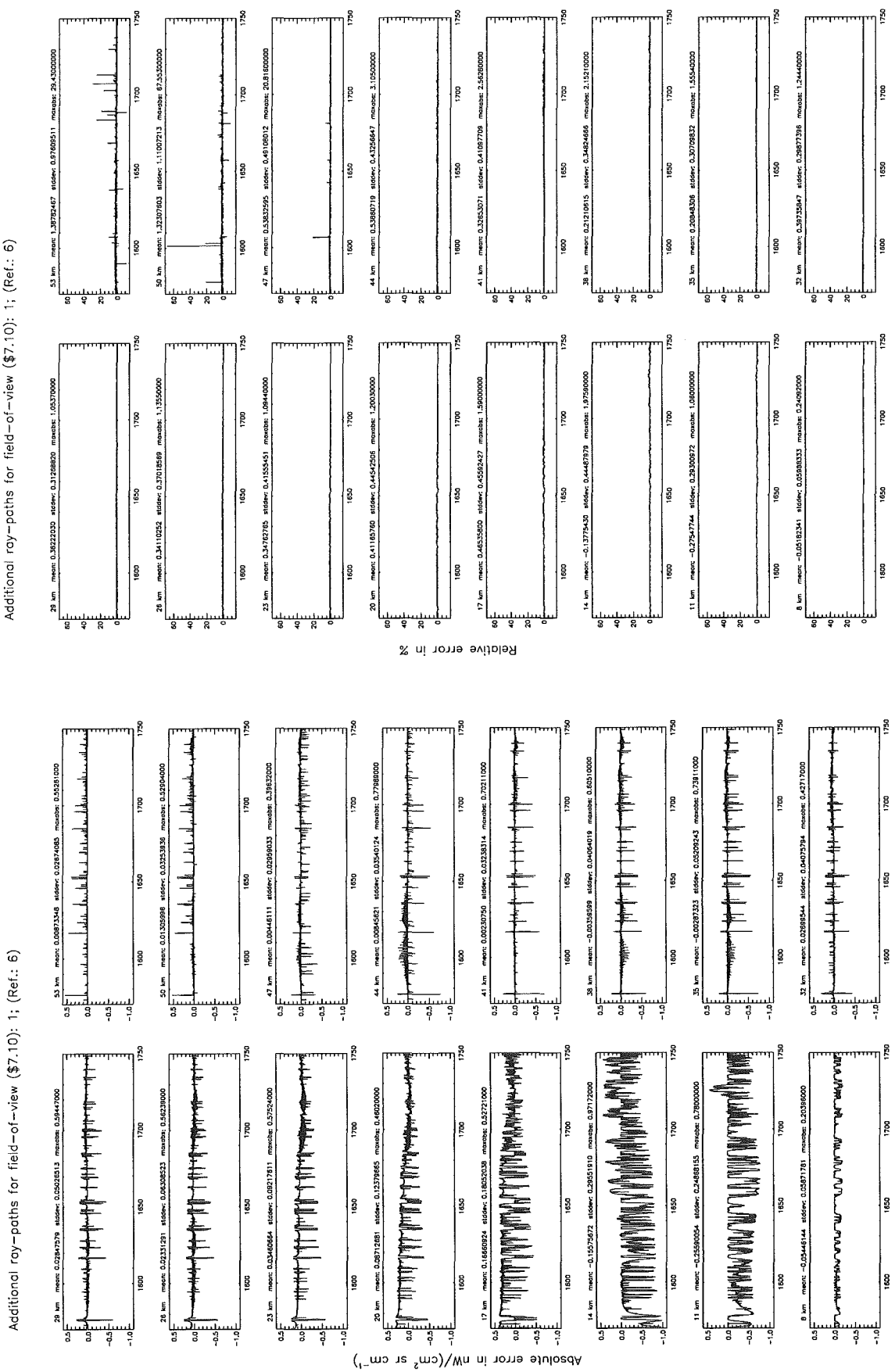


Figure 229: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

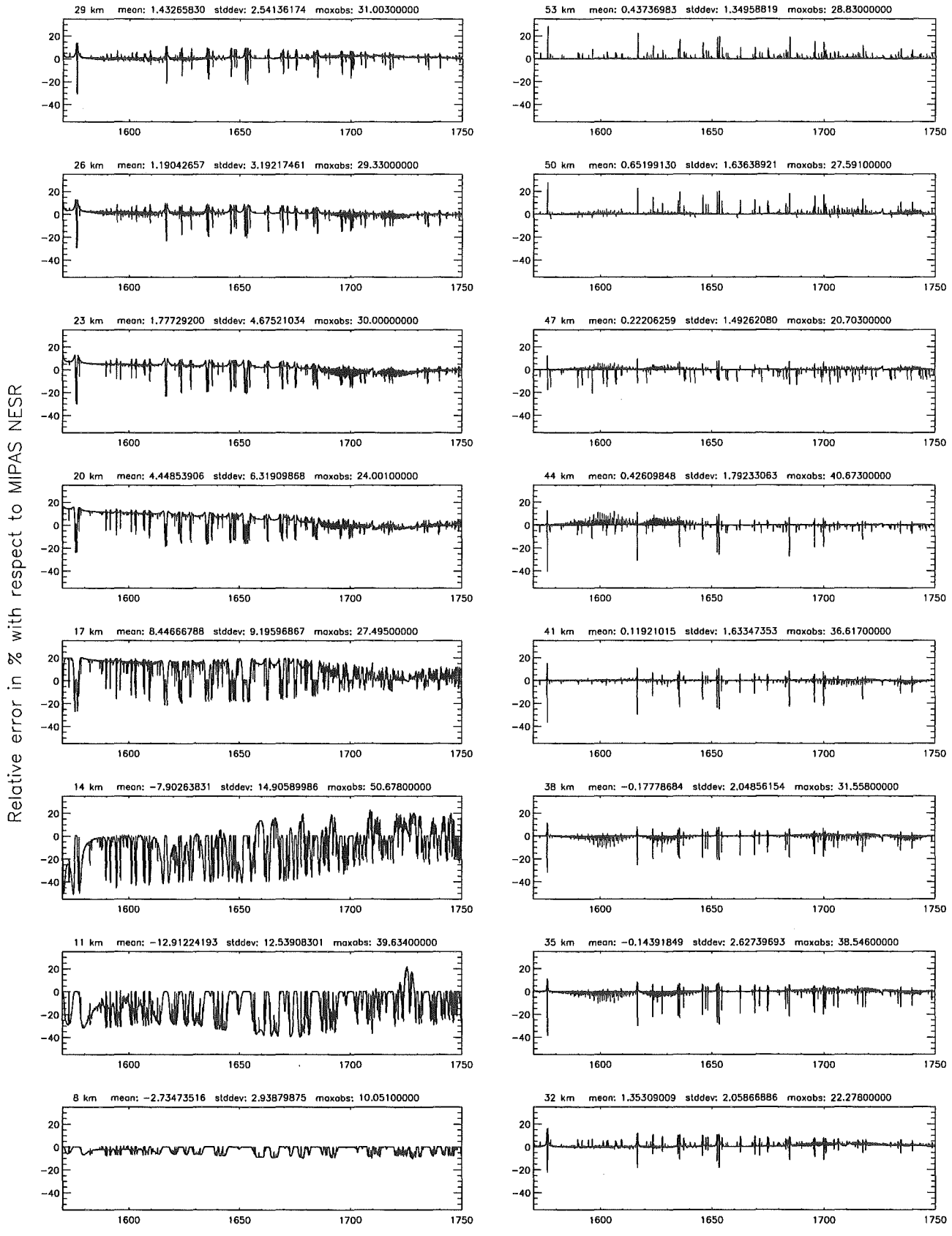


Figure 230: Relative error [%] with respect to MIPAS NESR

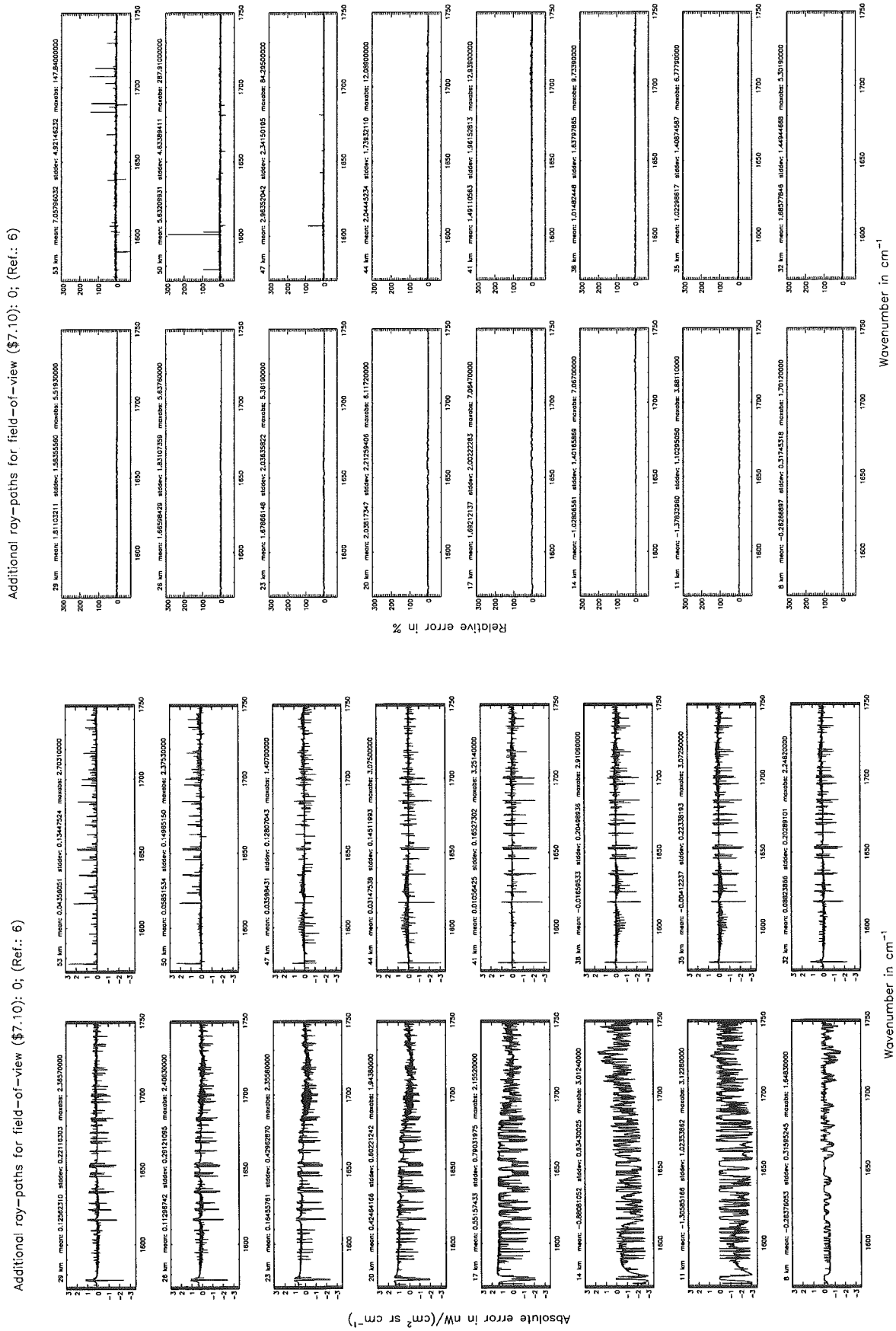


Figure 231: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

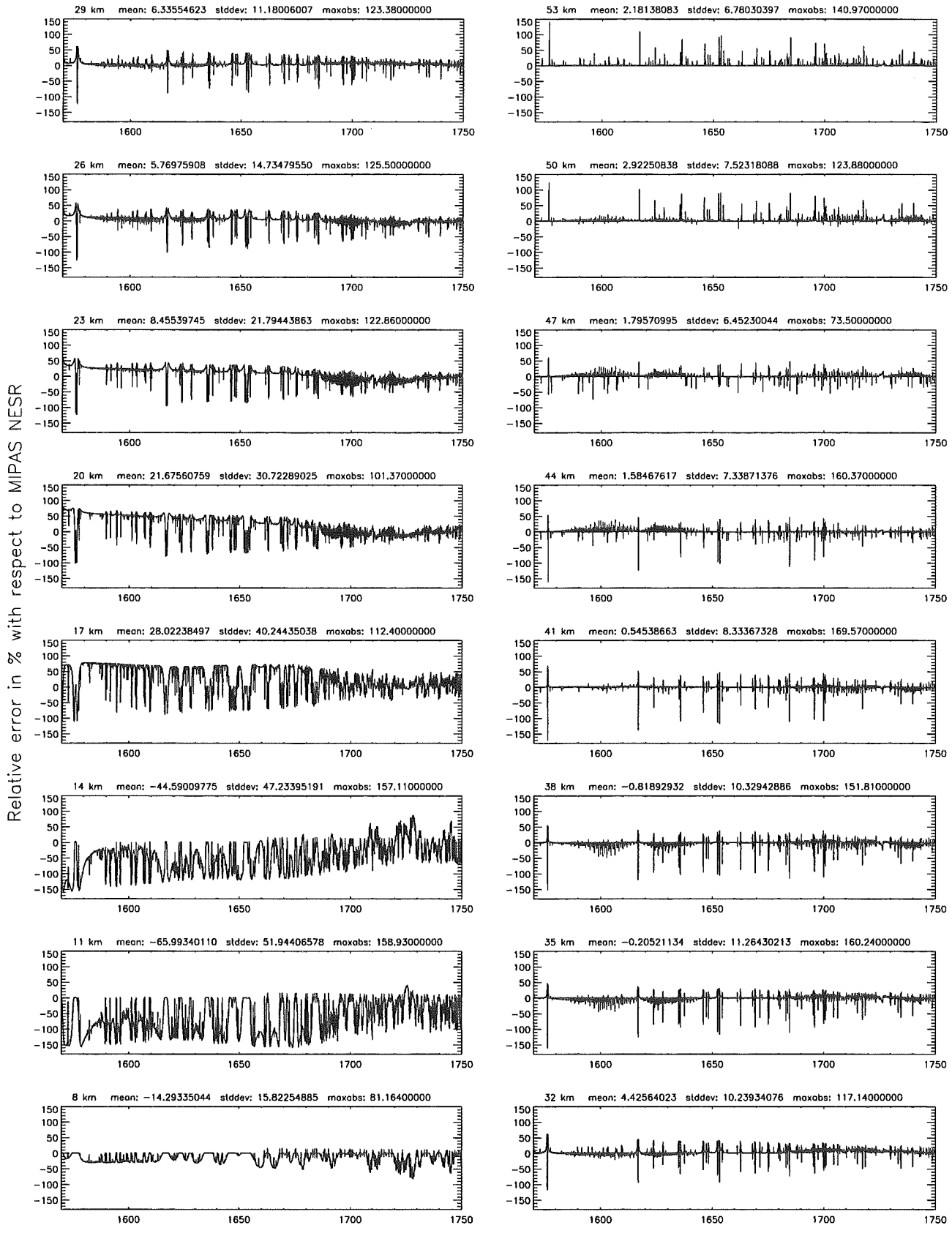


Figure 232: Relative error [%] with respect to MIPAS NESR

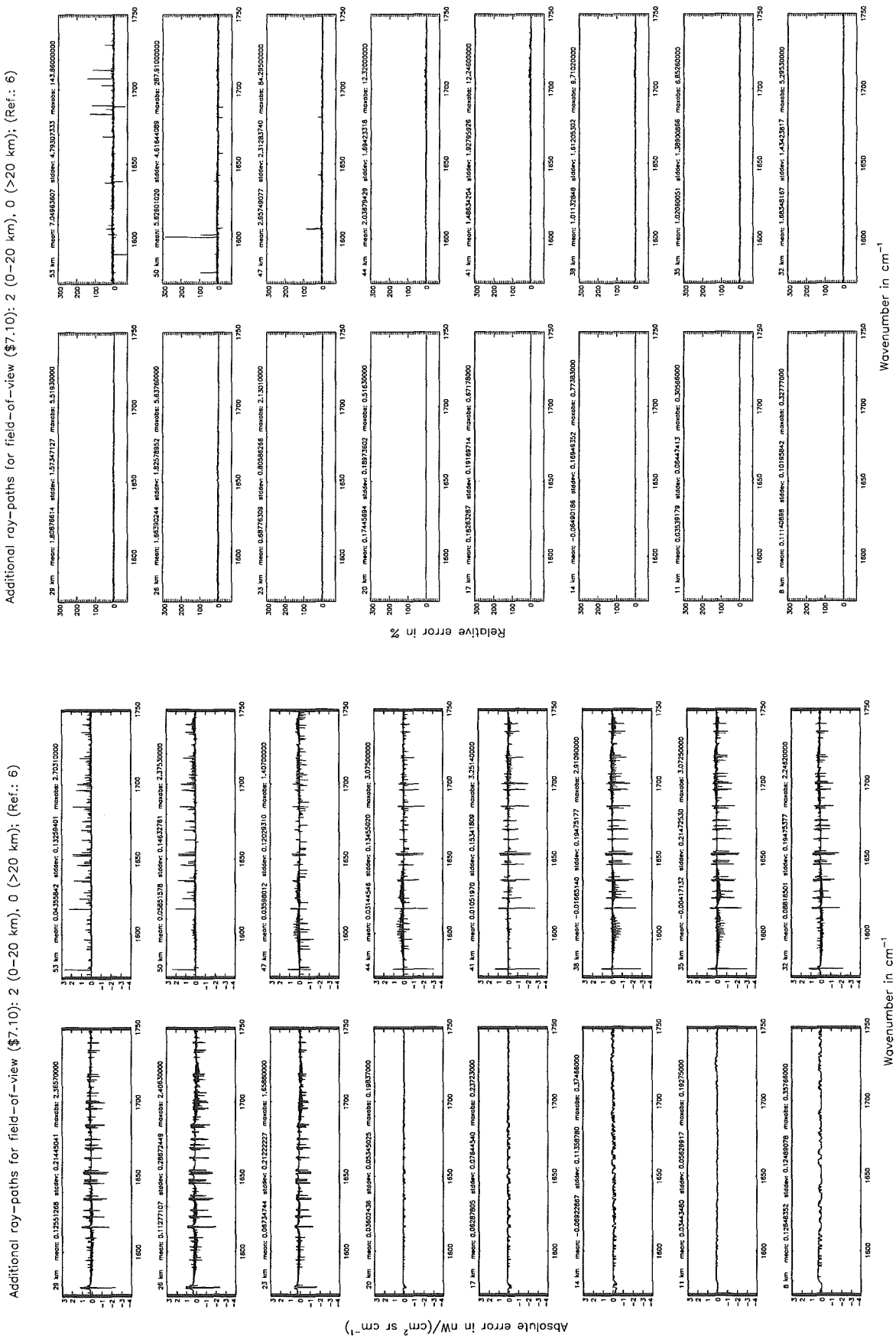


Figure 233: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

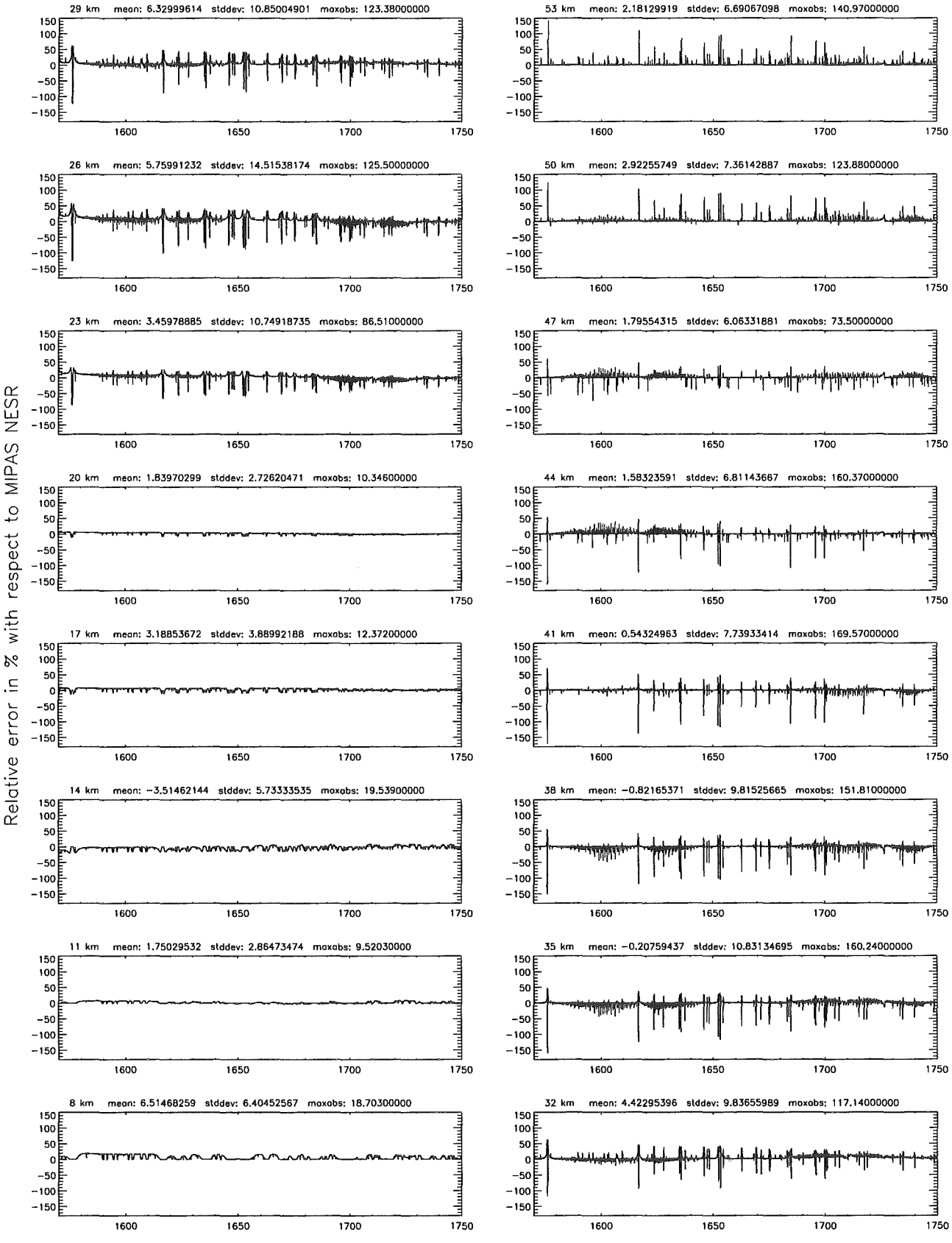
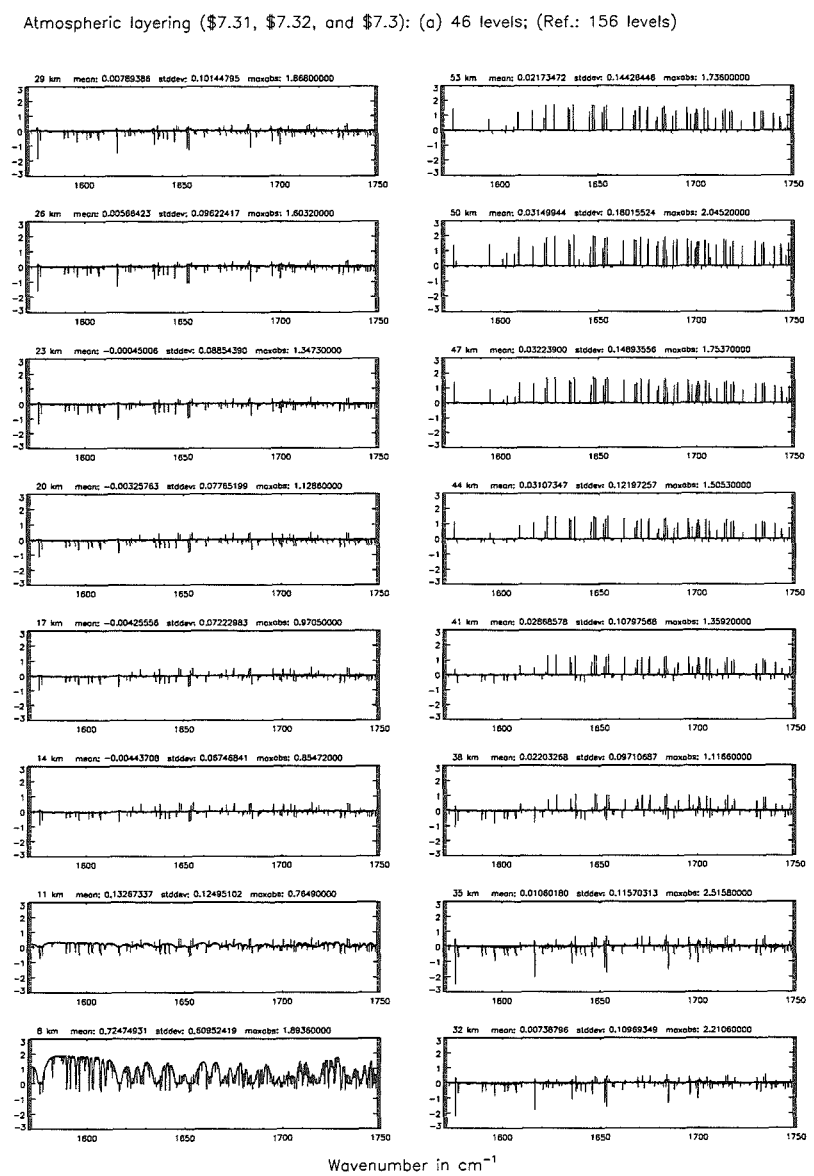


Figure 234: Relative error [%] with respect to MIPAS NESR



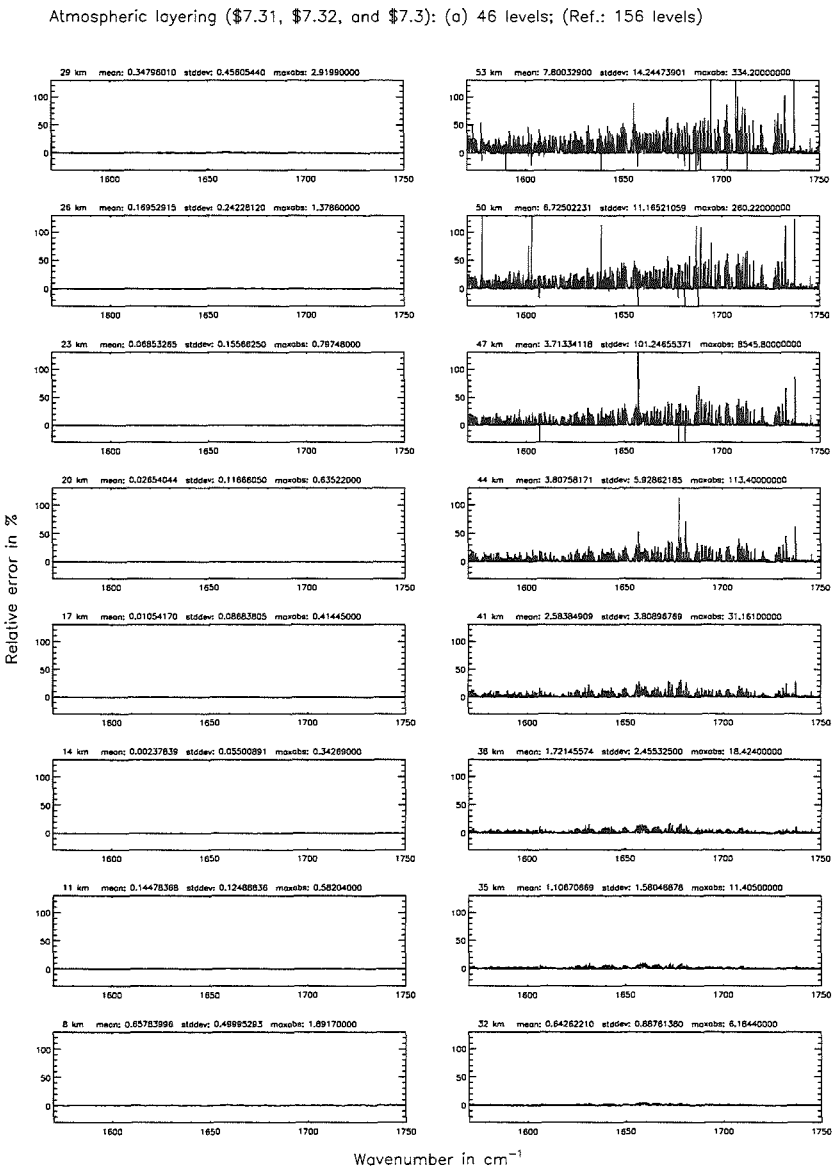


Figure 235: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

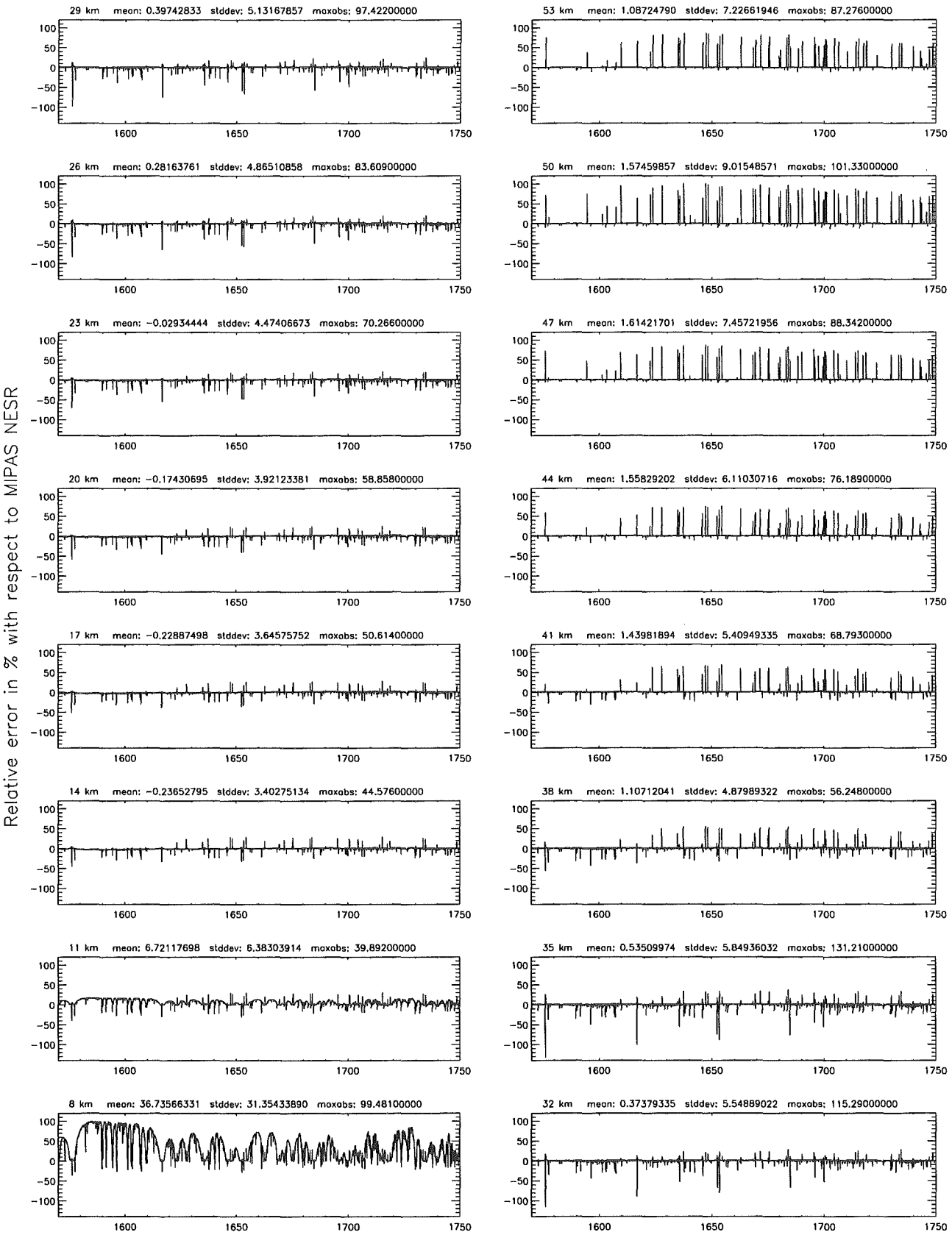


Figure 236: Relative error [%] with respect to MIPAS NESR

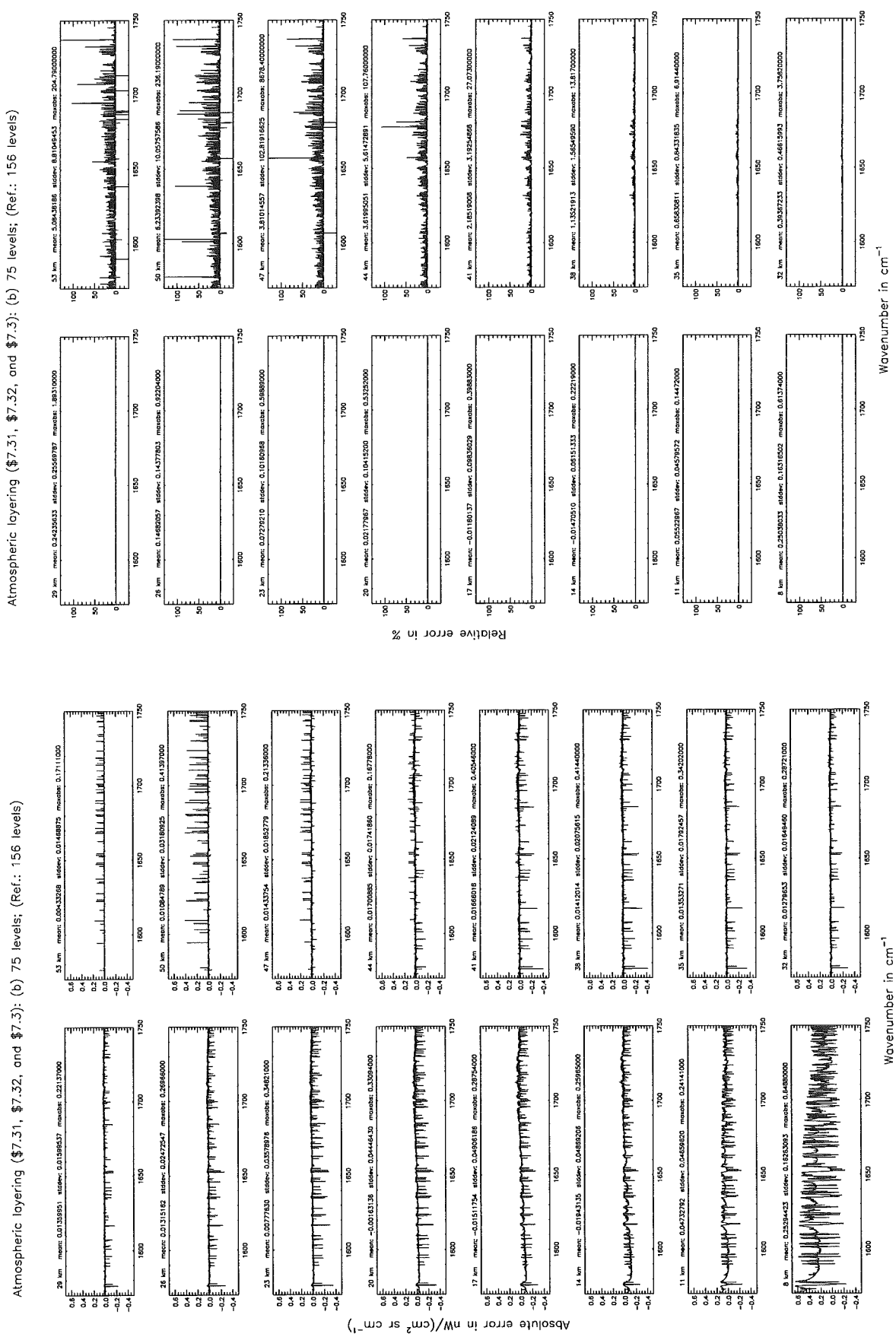


Figure 237: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (b) 75 levels; (Ref.: 156 levels)

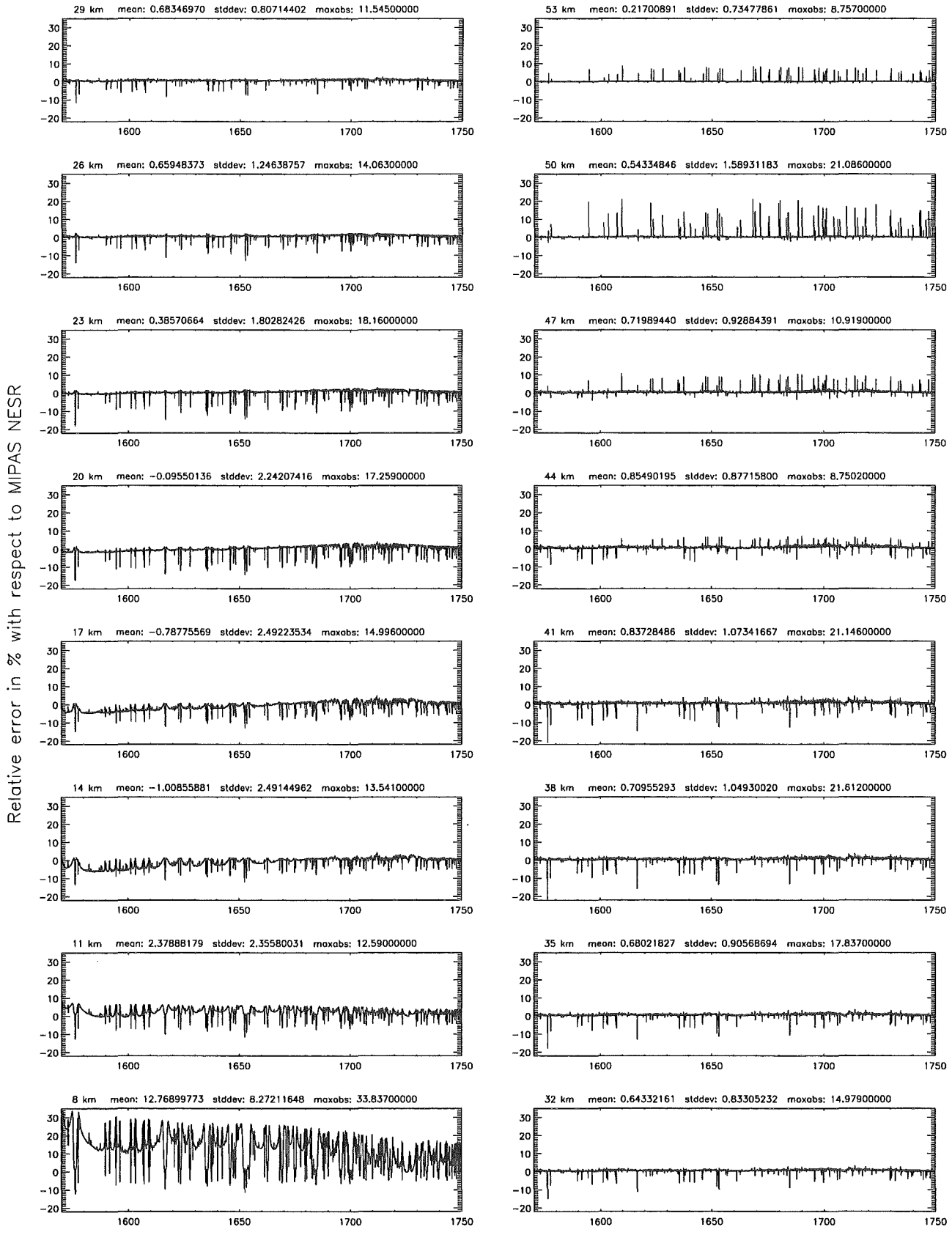


Figure 238: Relative error [%] with respect to MIPAS NESR

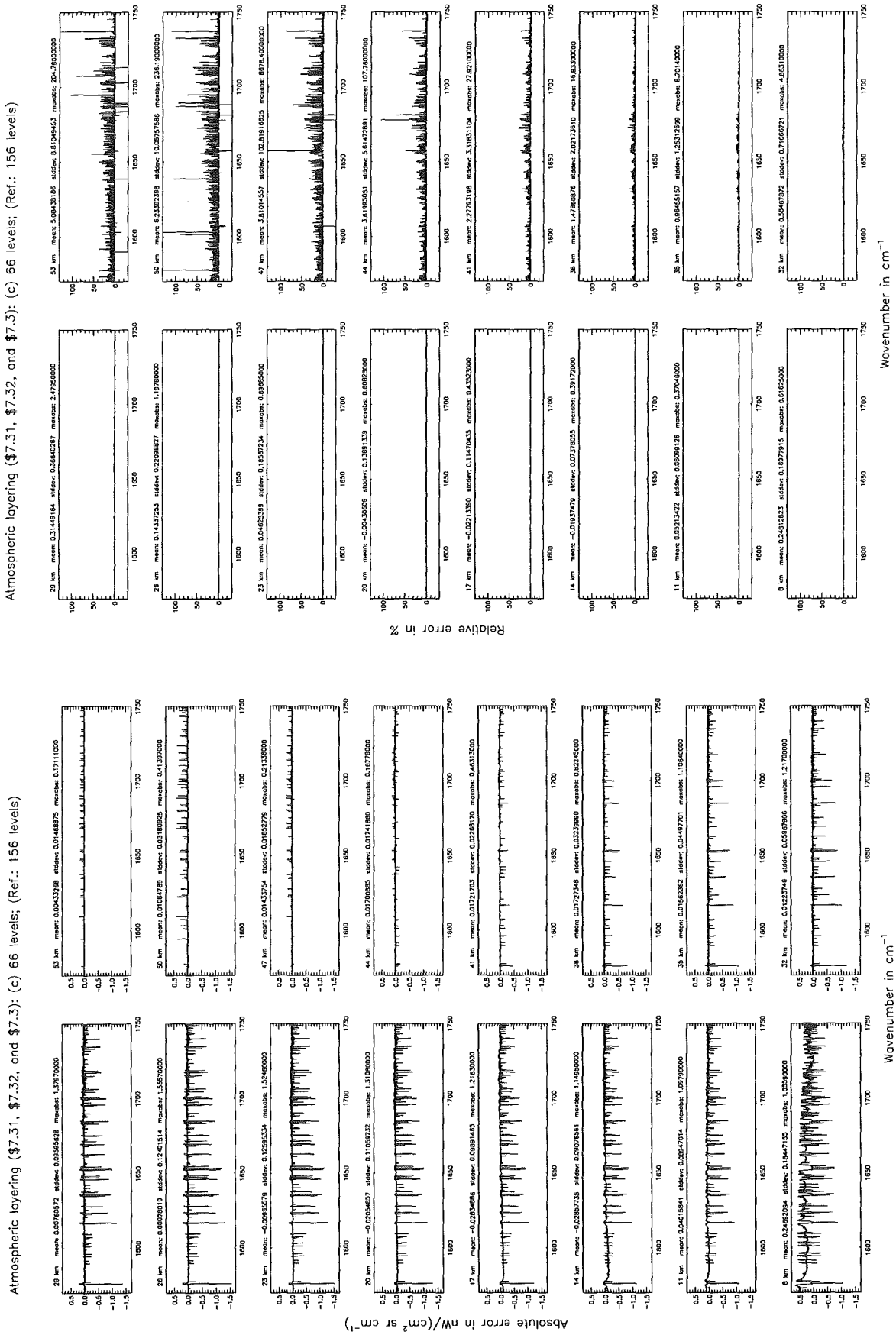


Figure 239: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (c) 66 levels; (Ref.: 156 levels)

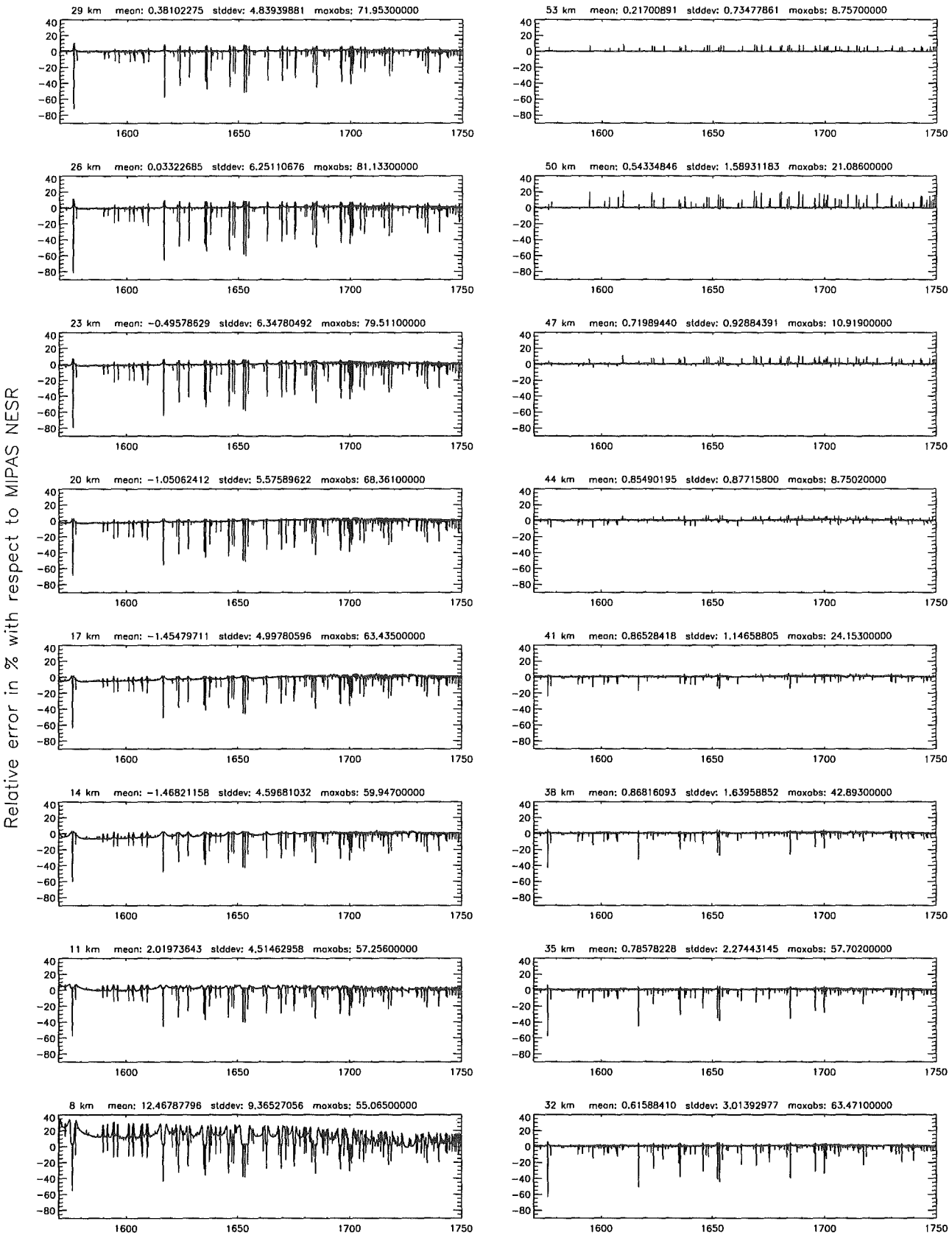
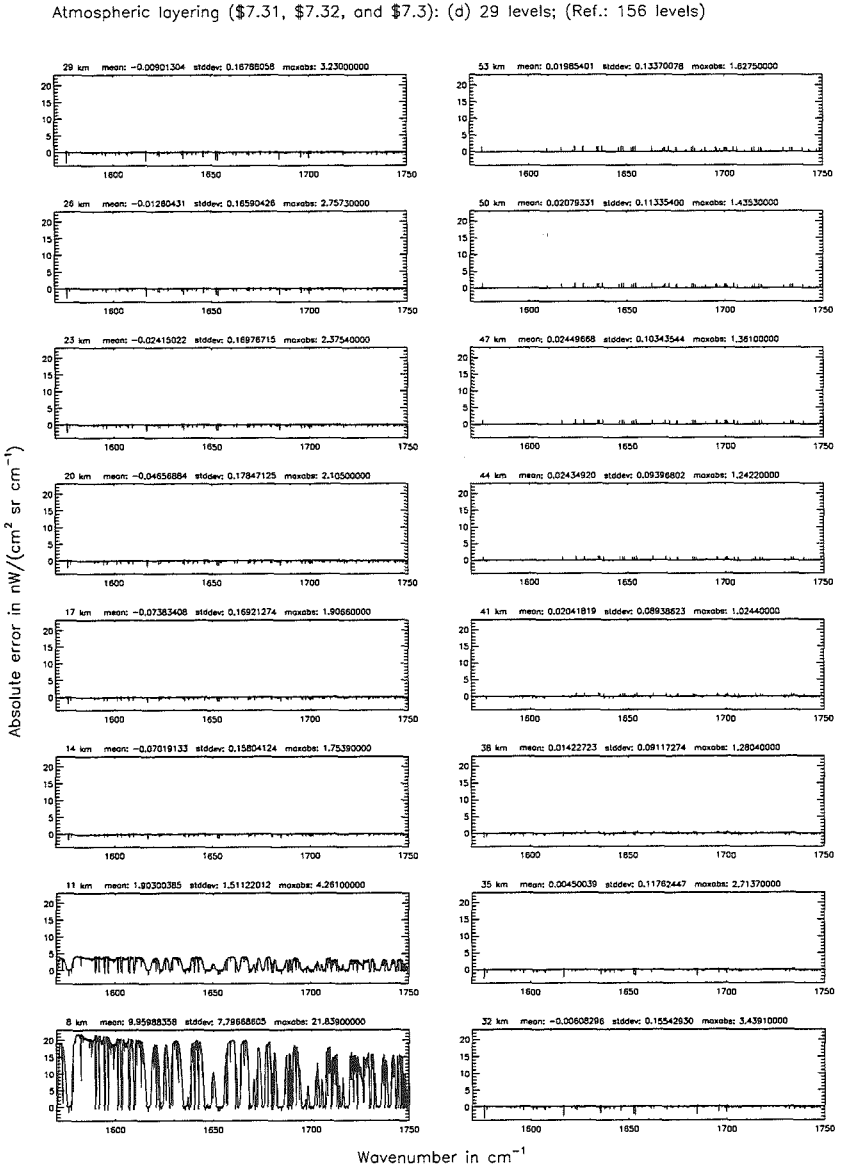


Figure 240: Relative error [%] with respect to MIPAS NESR



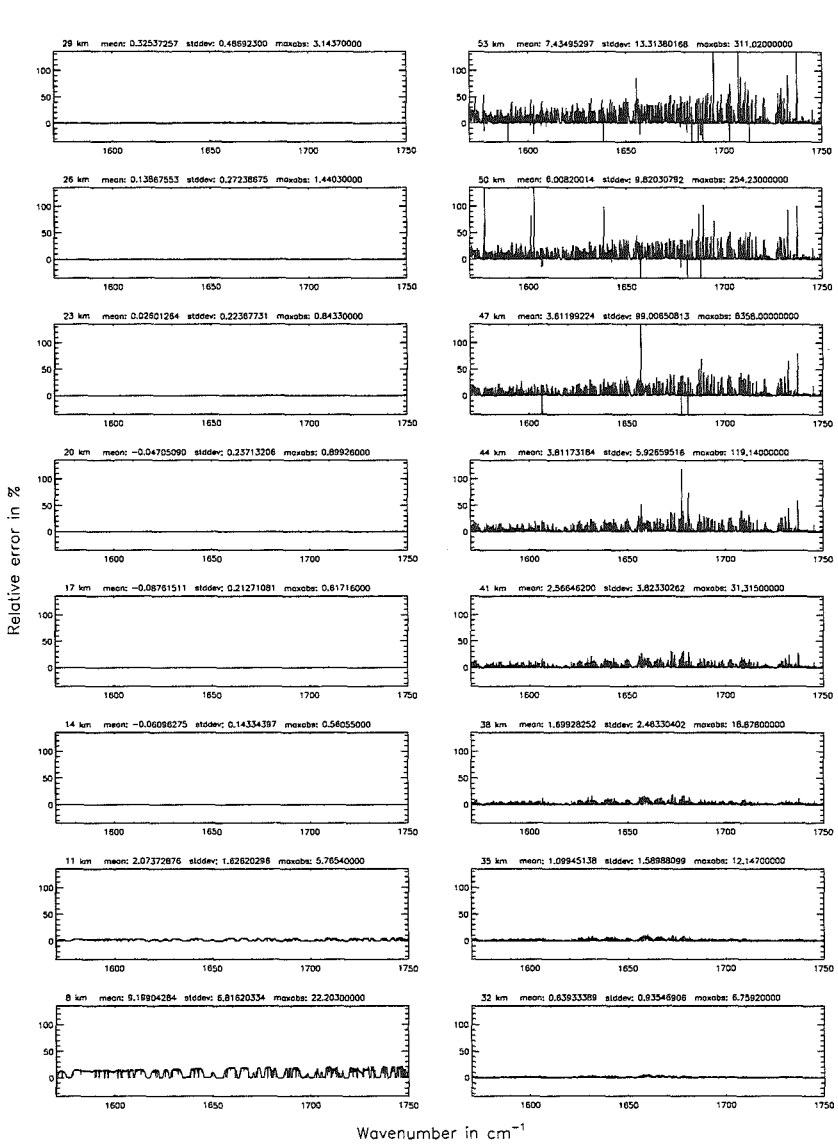


Figure 241: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (d) 29 levels; (Ref.: 156 levels)

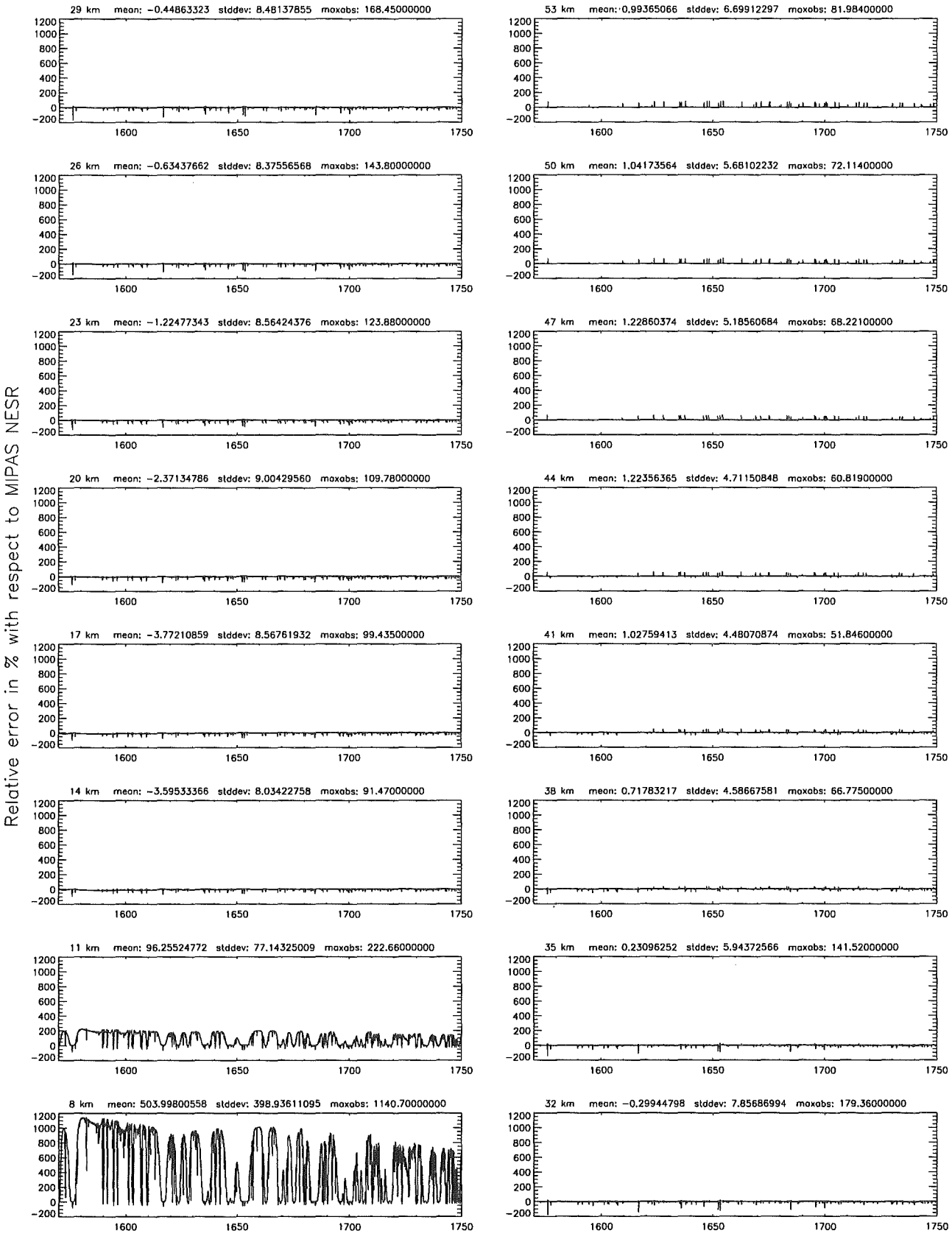
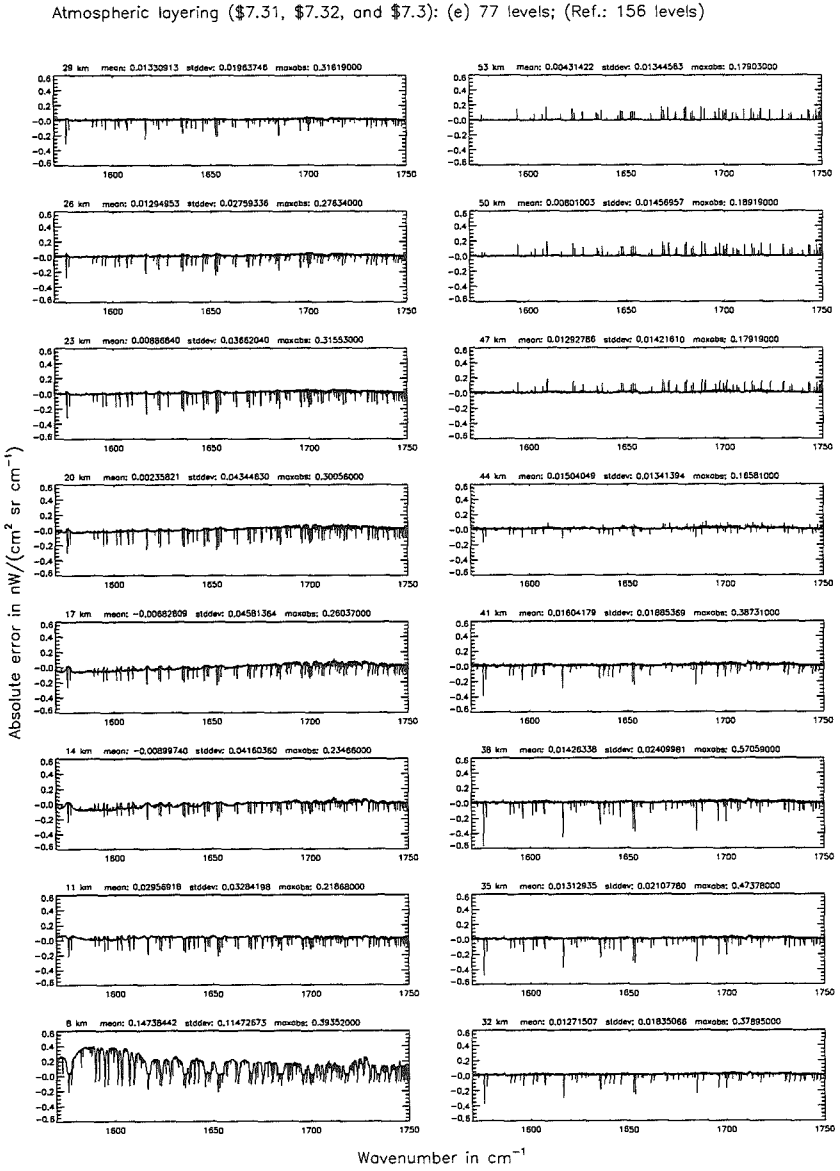


Figure 242: Relative error [%] with respect to MIPAS NESR



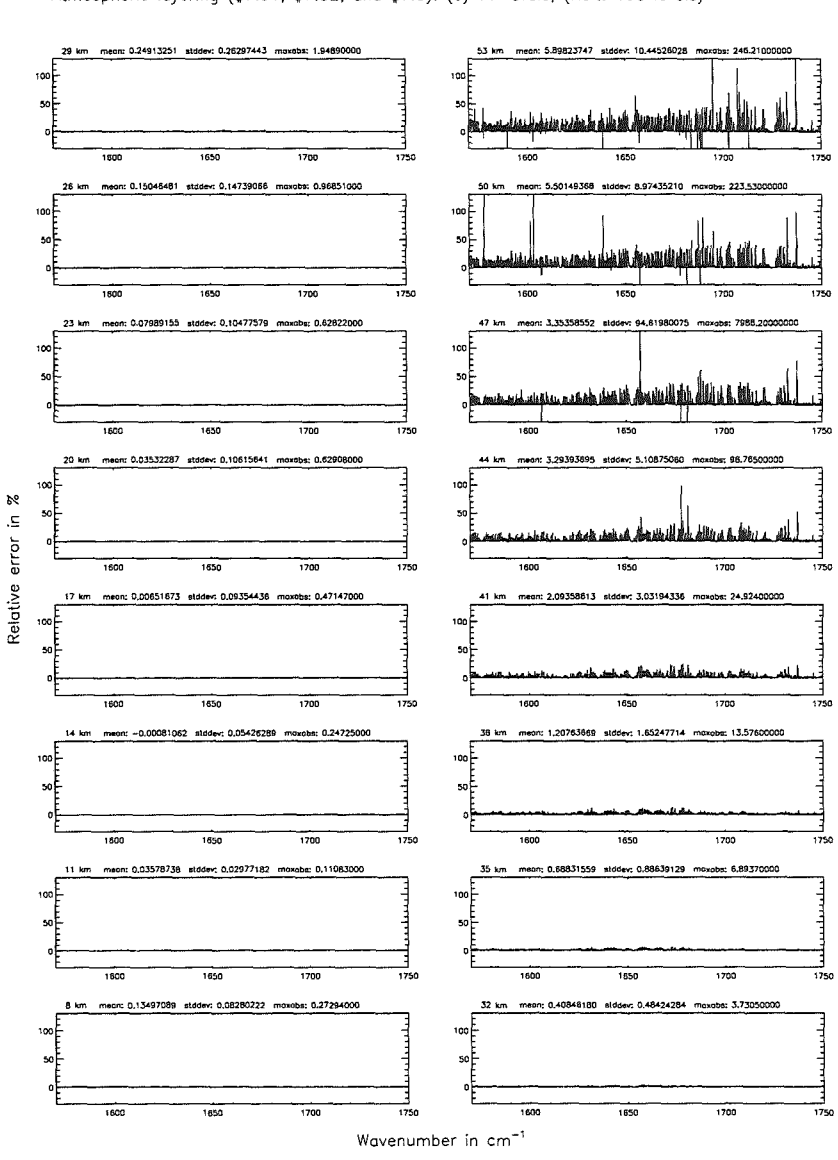


Figure 243: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (e) 77 levels; (Ref.: 156 levels)

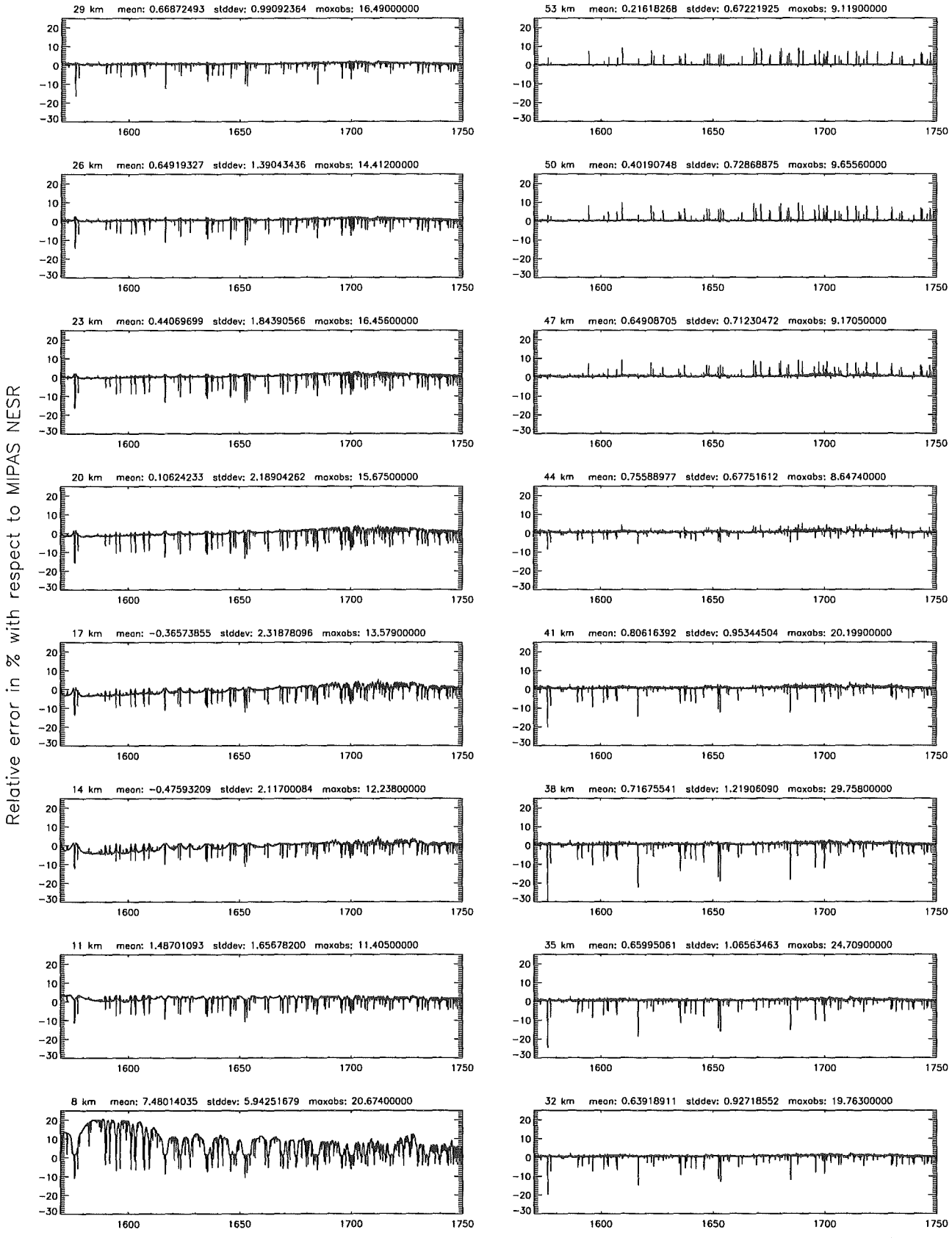
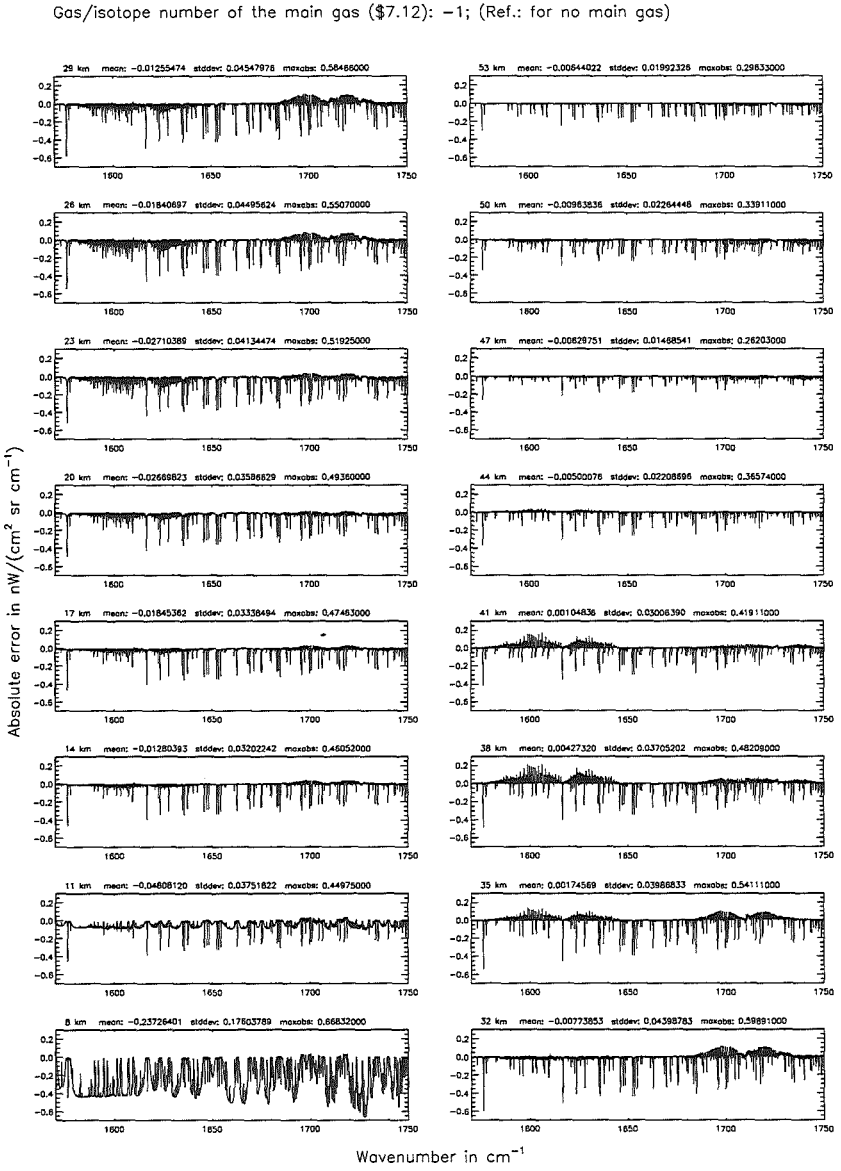


Figure 244: Relative error [%] with respect to MIPAS NESR



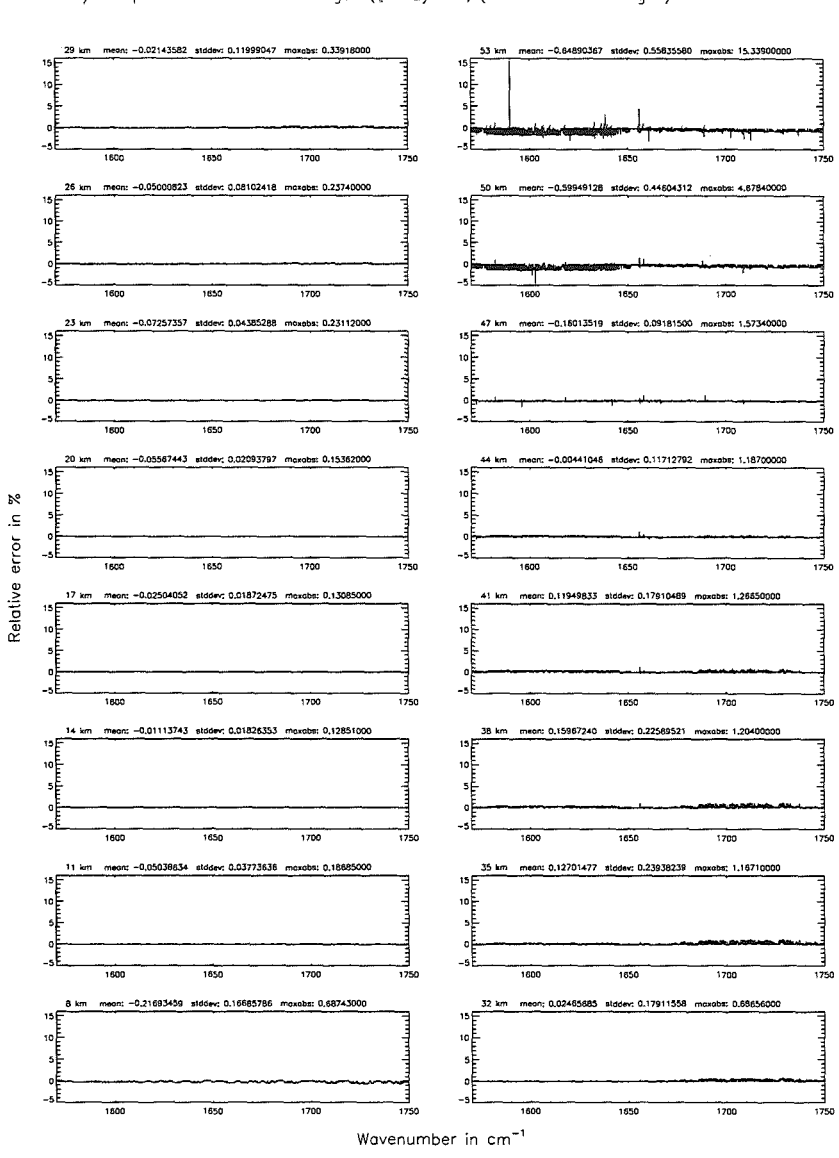


Figure 245: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Gas/isotope number of the main gas ($7.12): −1; (Ref.: for no main gas)

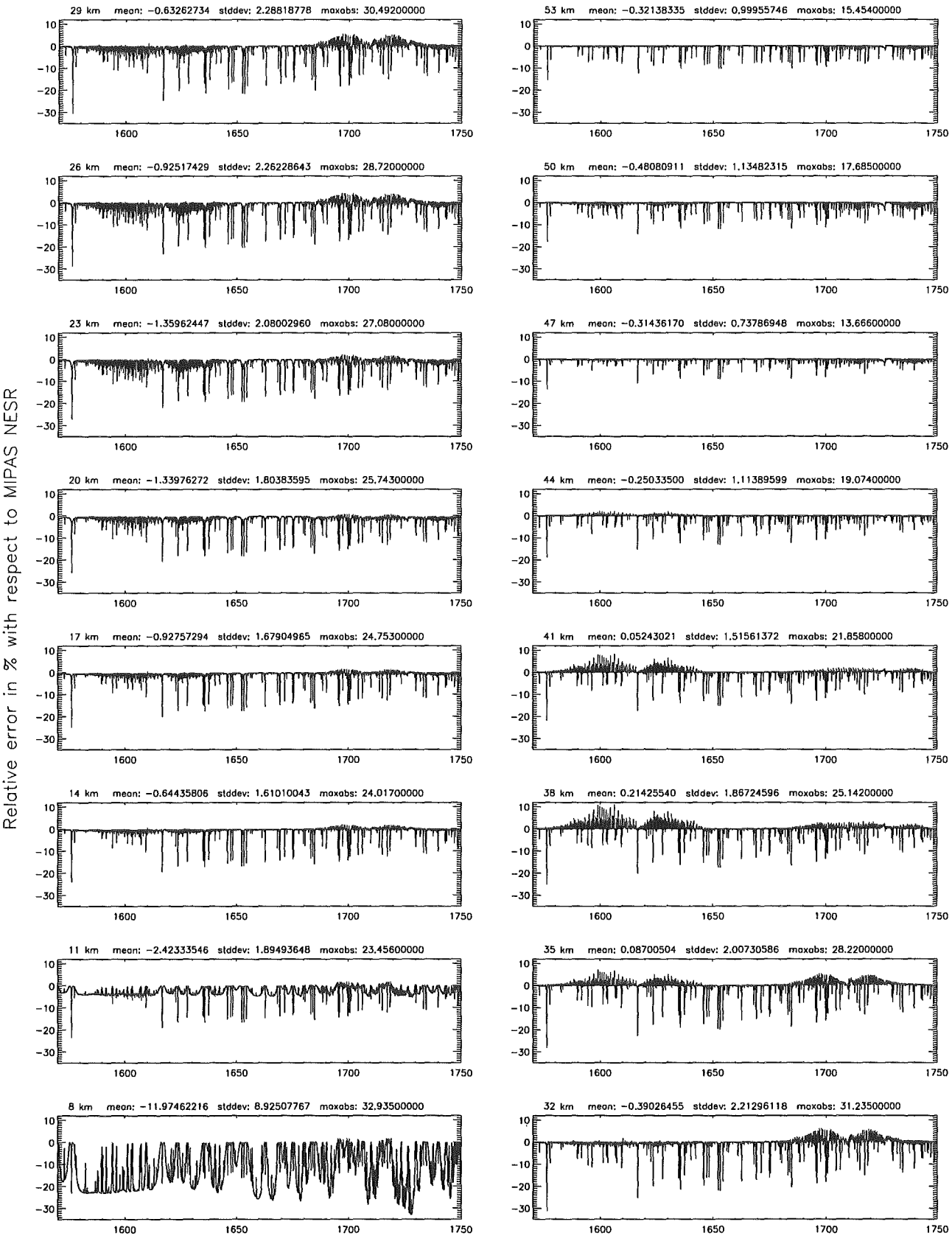


Figure 246: Relative error [%] with respect to MIPAS NESR

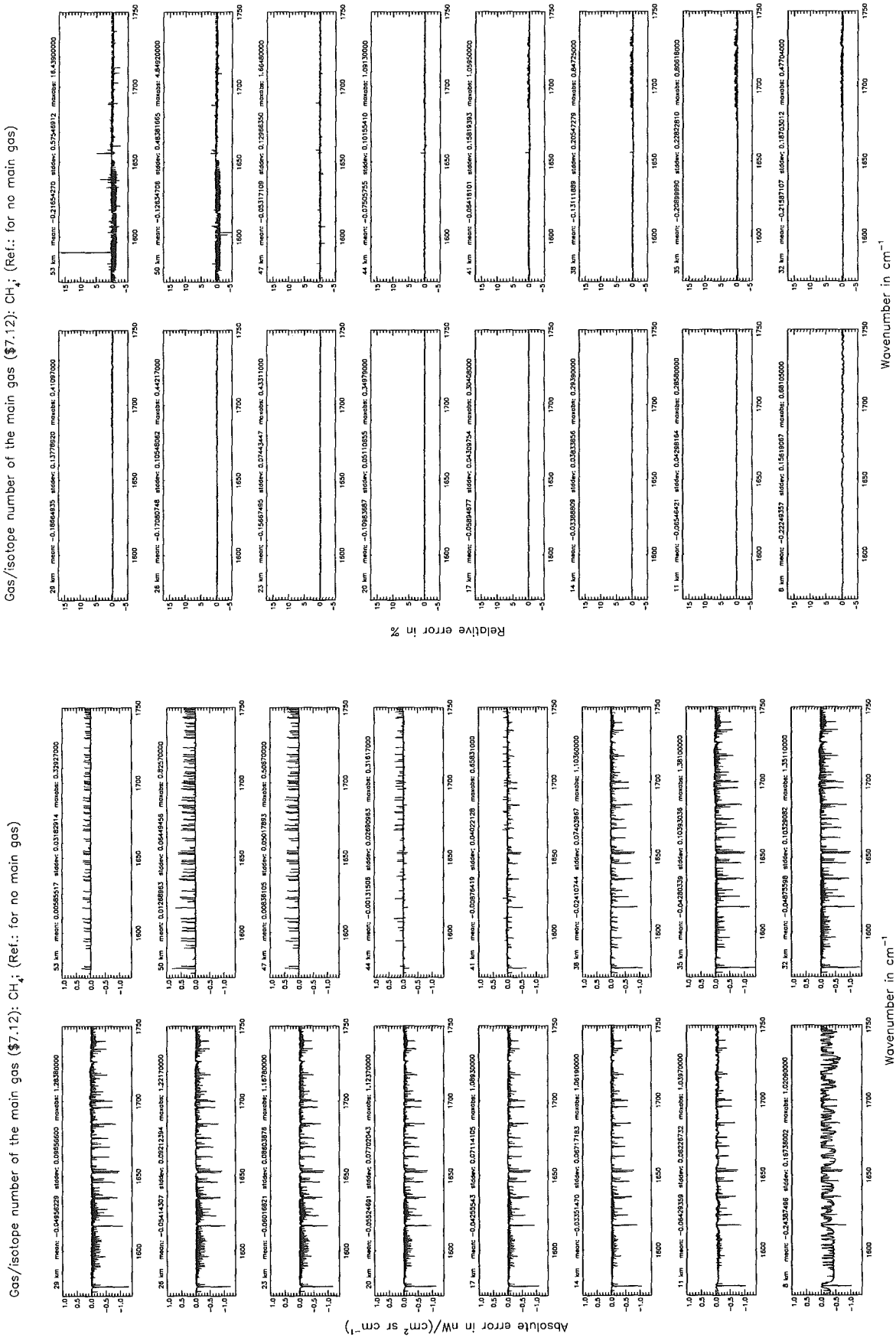


Figure 247: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

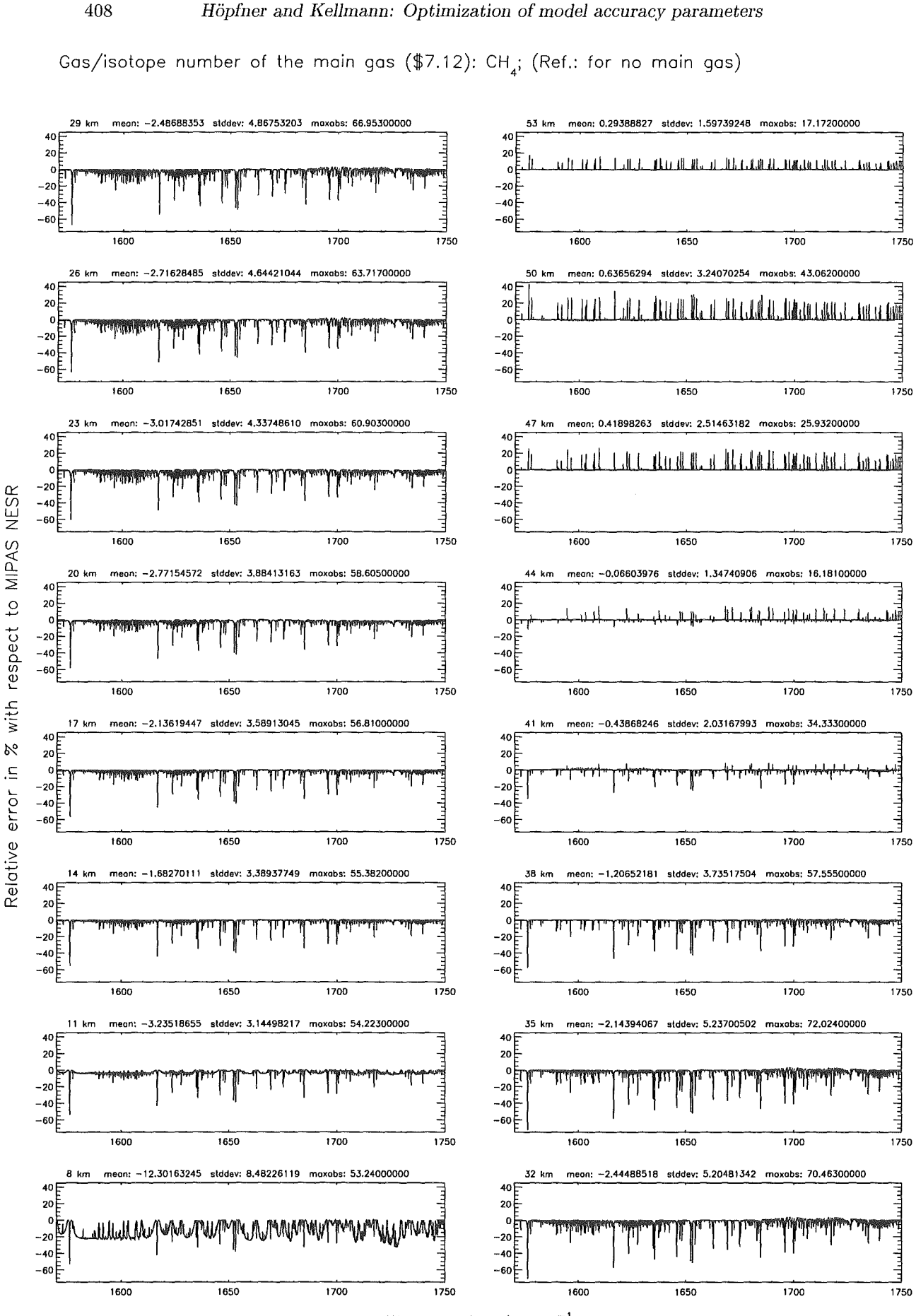


Figure 248: Relative error [%] with respect to MIPAS NESR

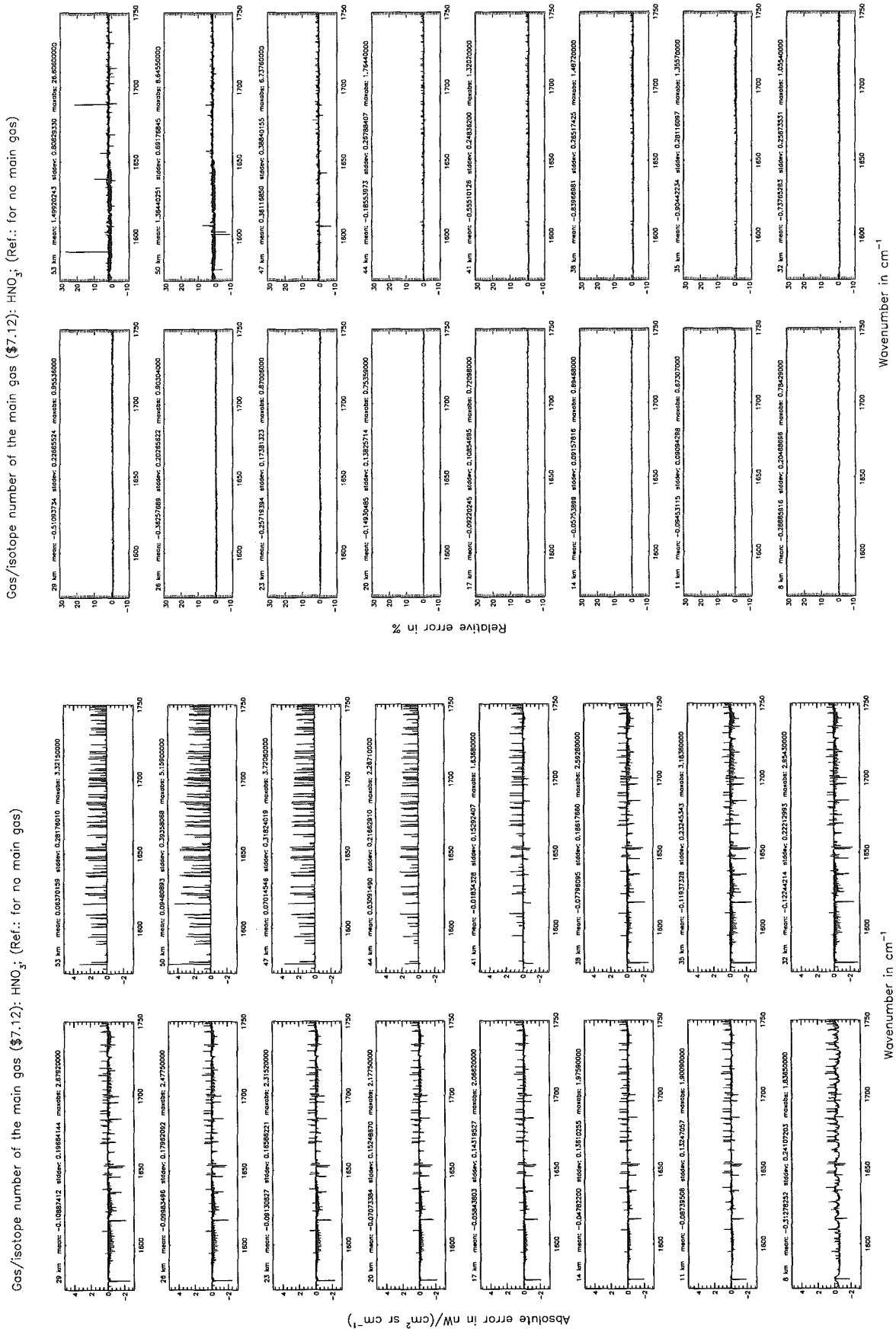


Figure 249: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

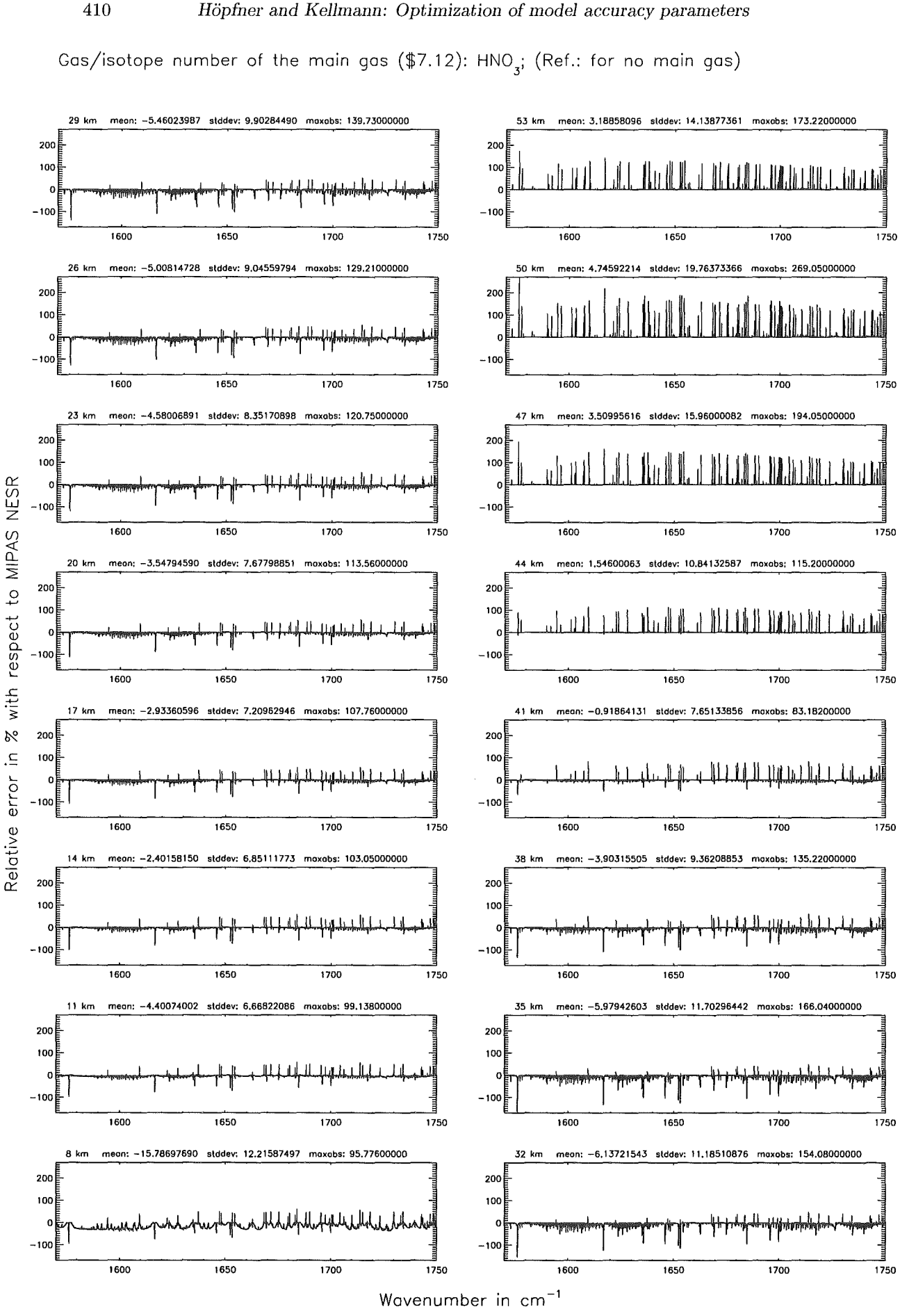
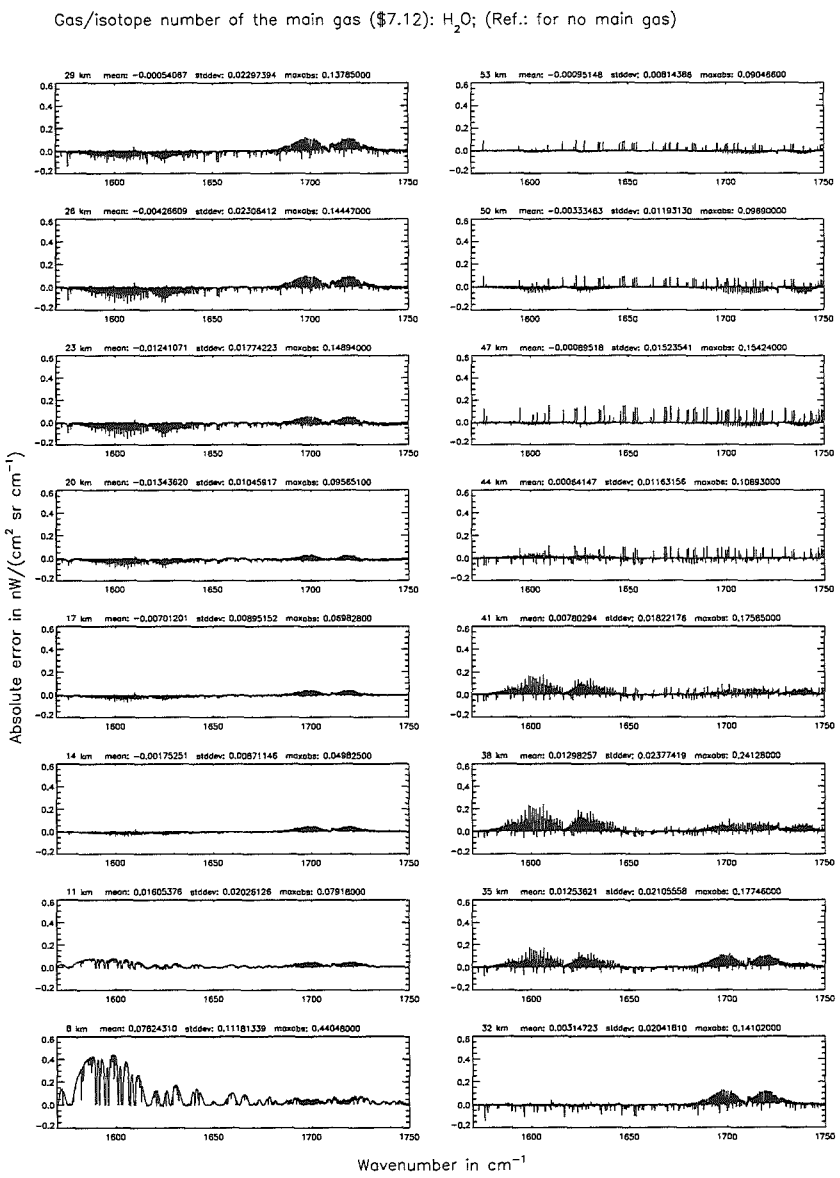


Figure 250: Relative error [%] with respect to MIPAS NESR



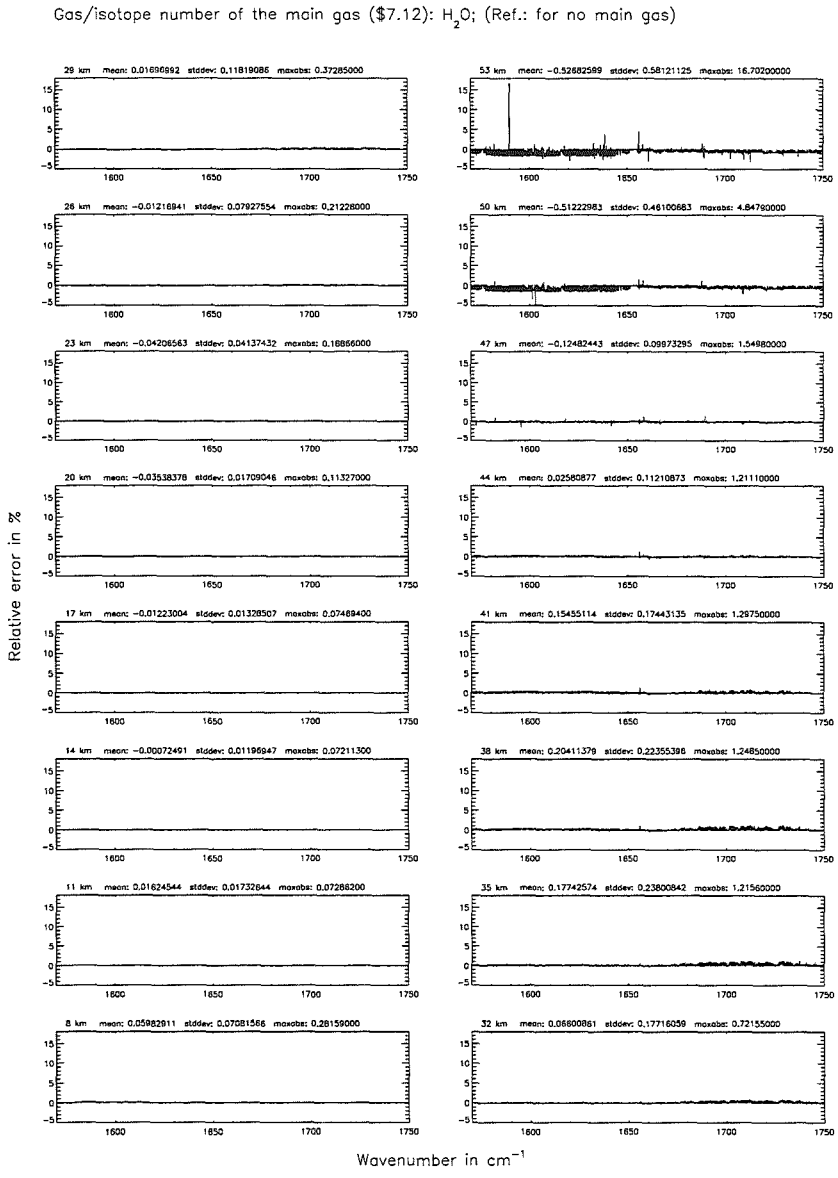


Figure 251: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

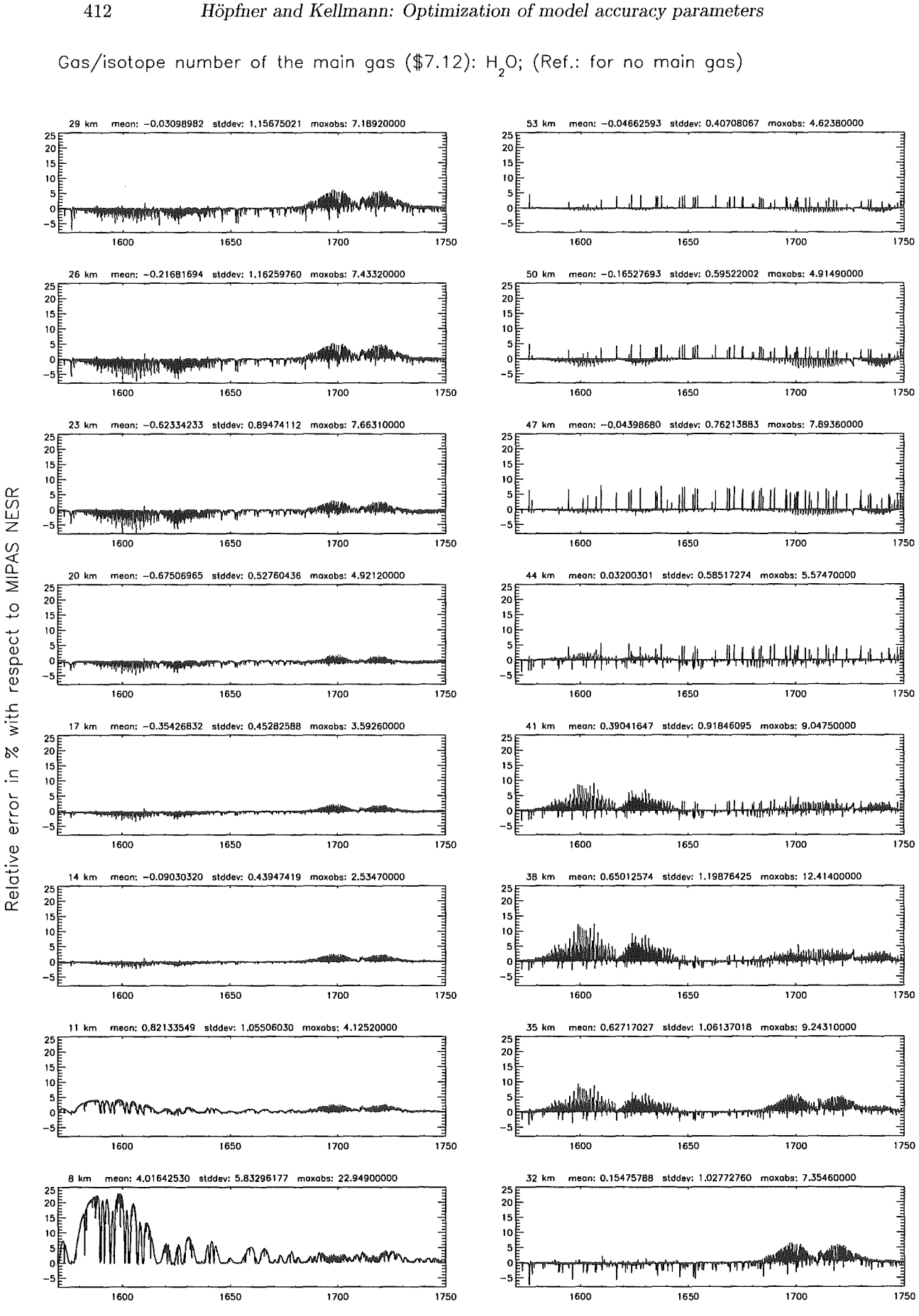


Figure 252: Relative error [%] with respect to MIPAS NESR

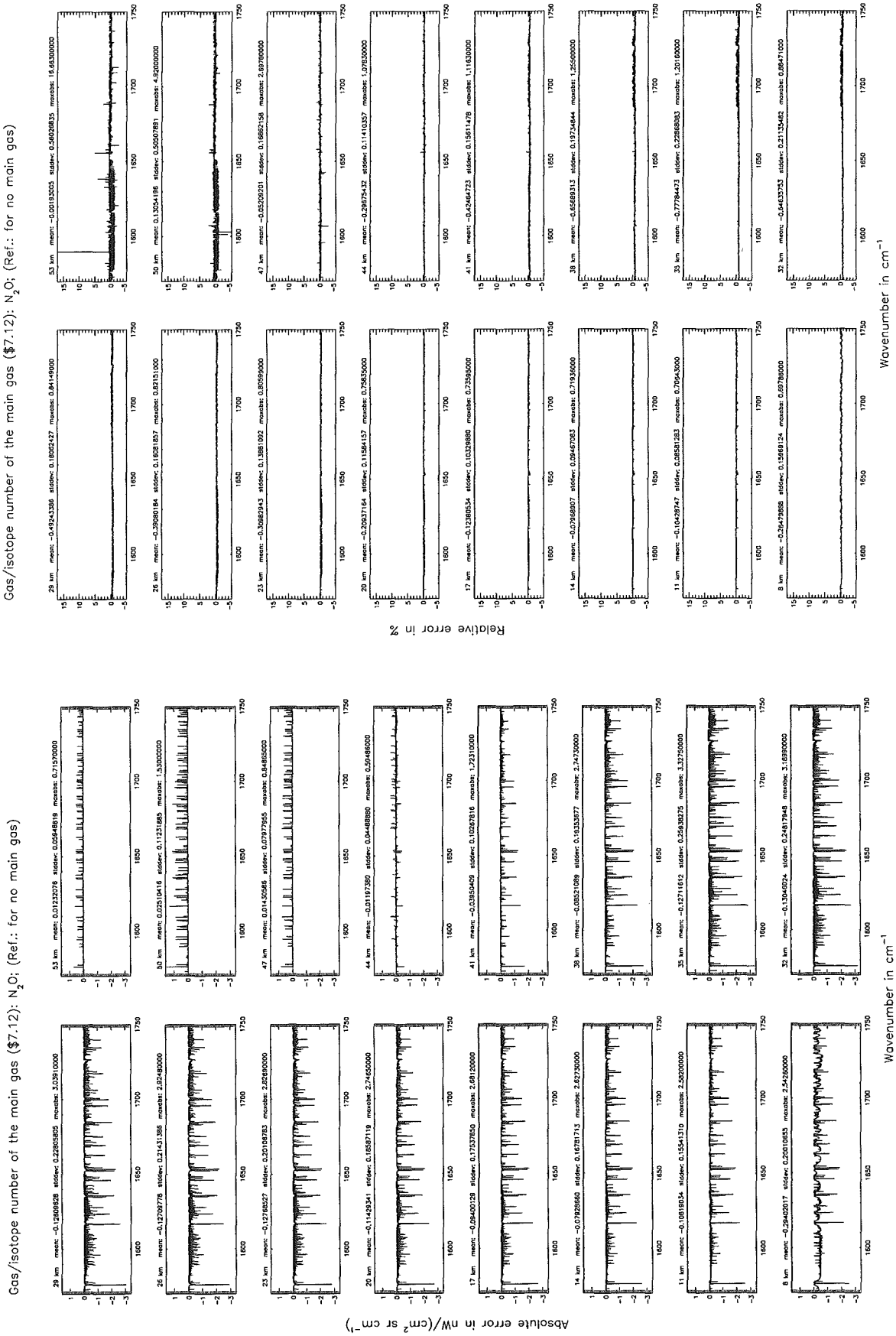


Figure 253: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

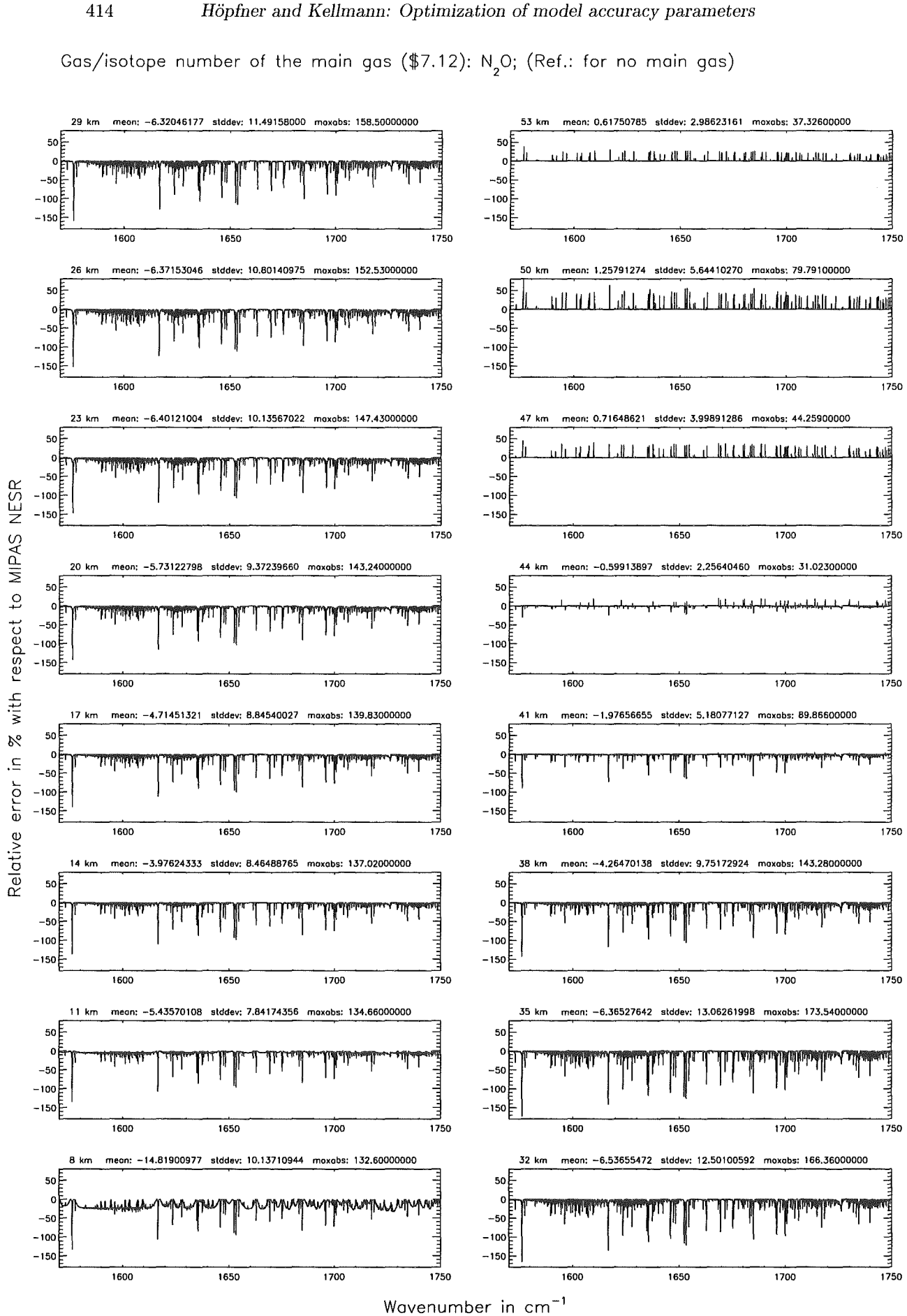


Figure 254: Relative error [%] with respect to MIPAS NESR

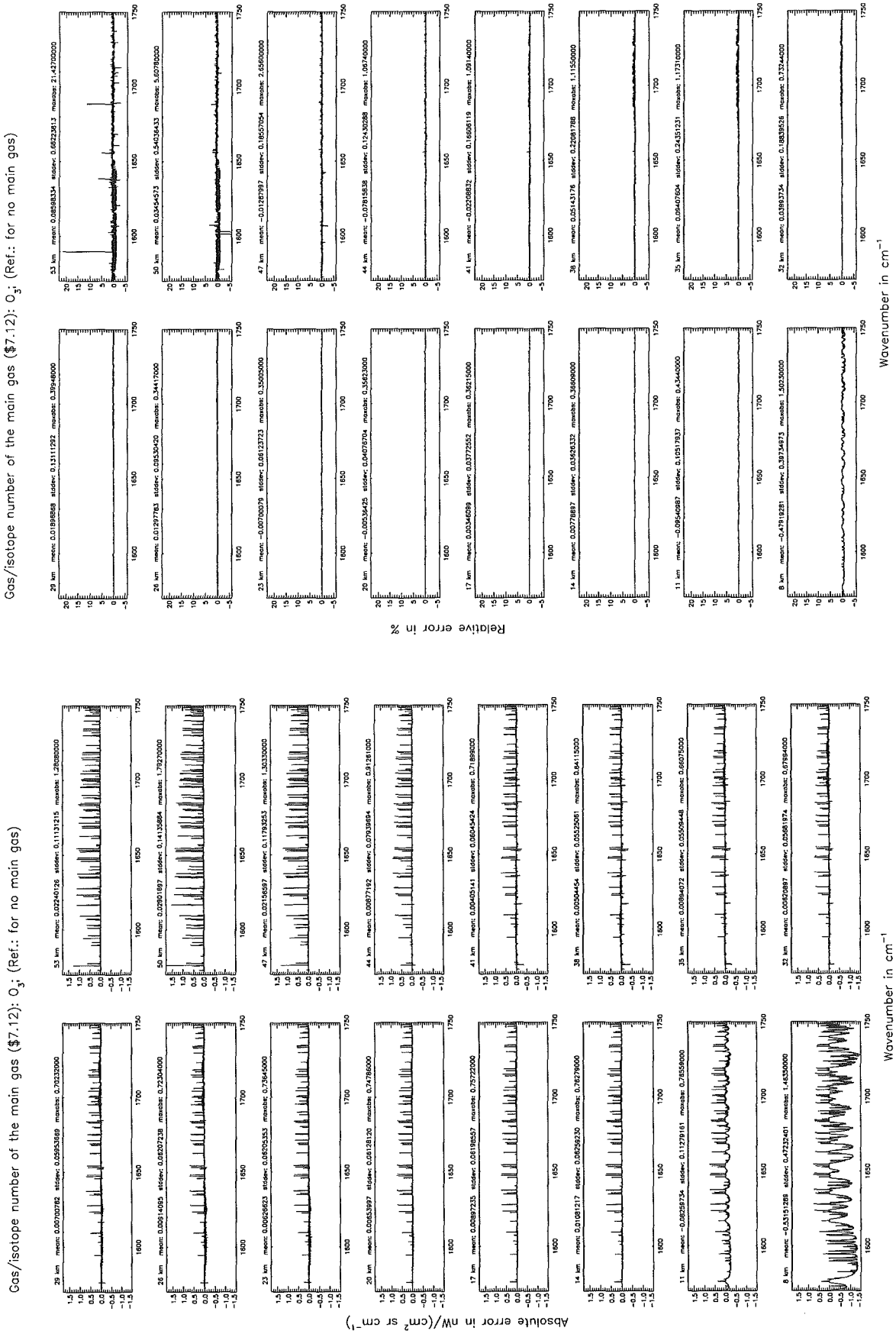


Figure 255: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

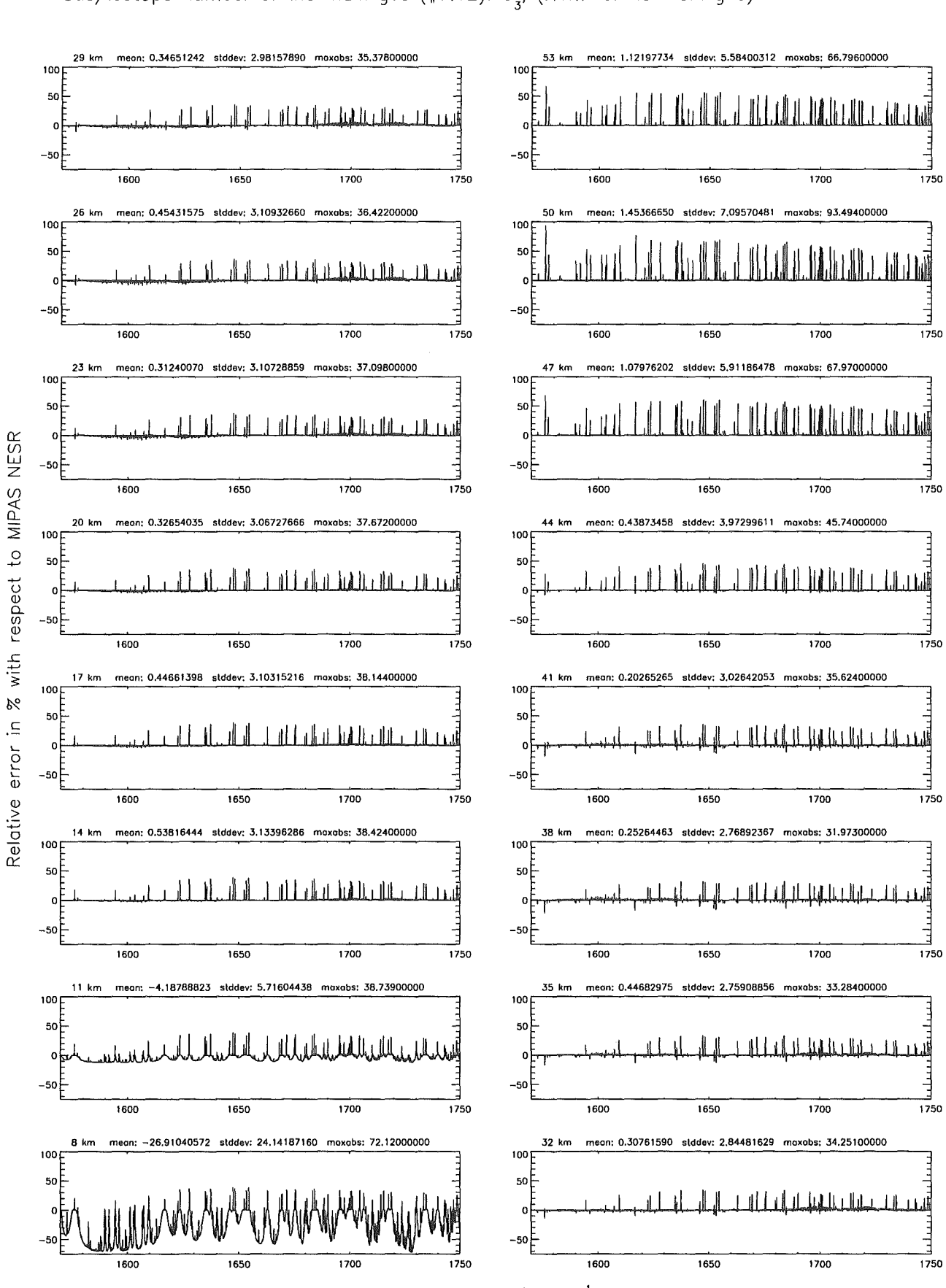
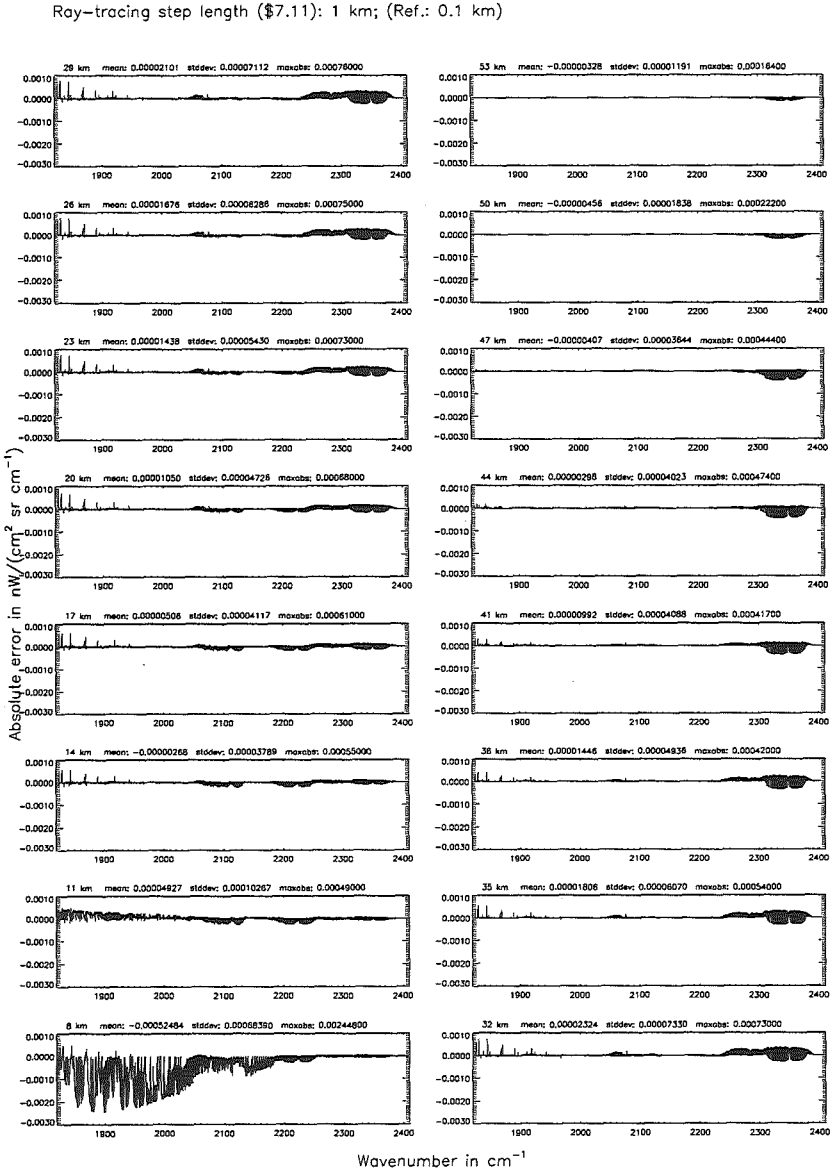


Figure 256: Relative error [%] with respect to MIPAS NESR

Band D: 1820 – 2410 cm^{-1}



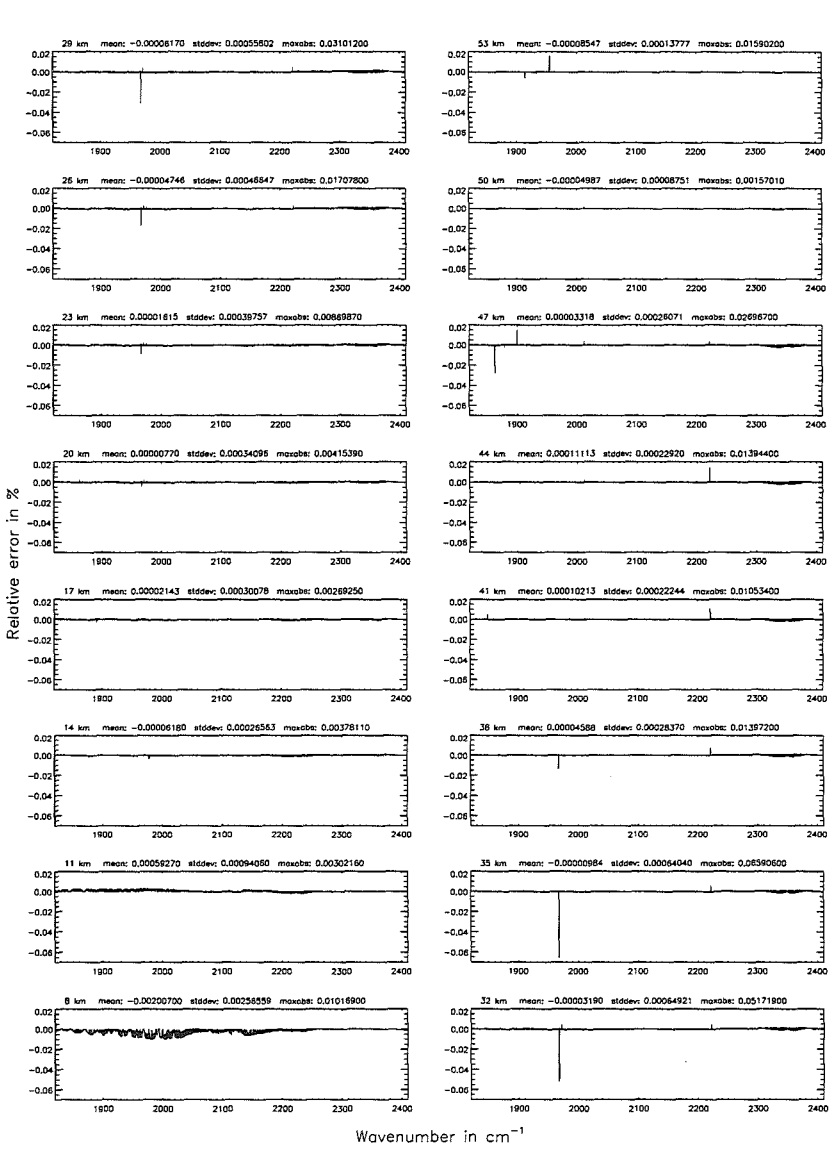


Figure 257: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

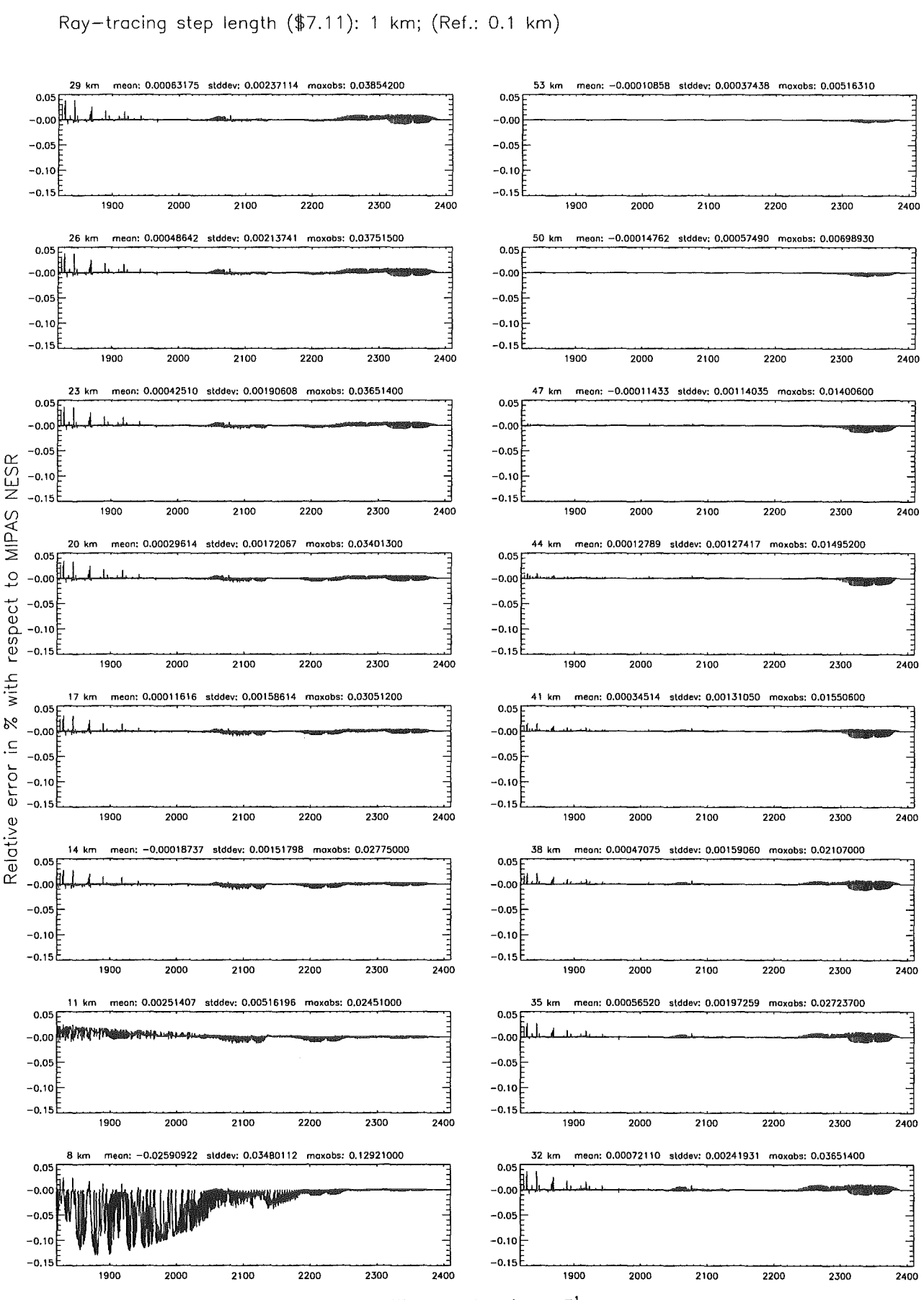
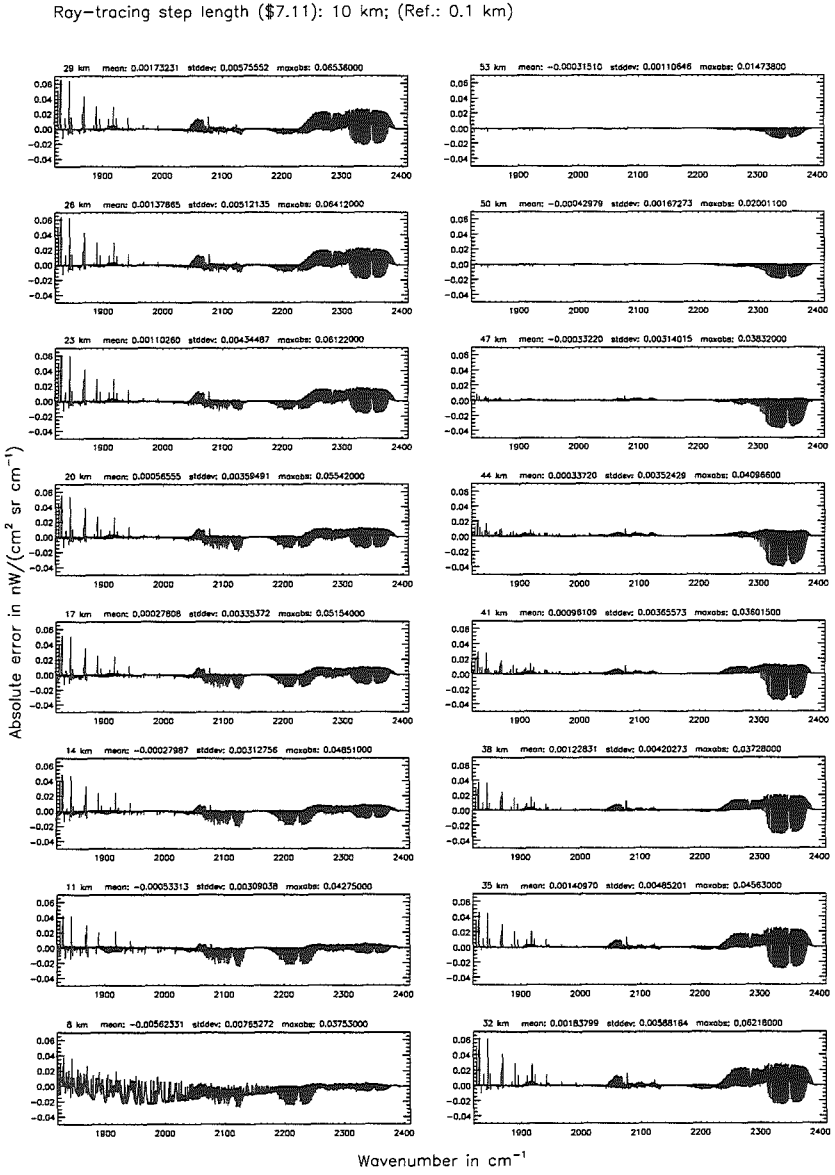


Figure 258: Relative error [%] with respect to MIPAS NESR



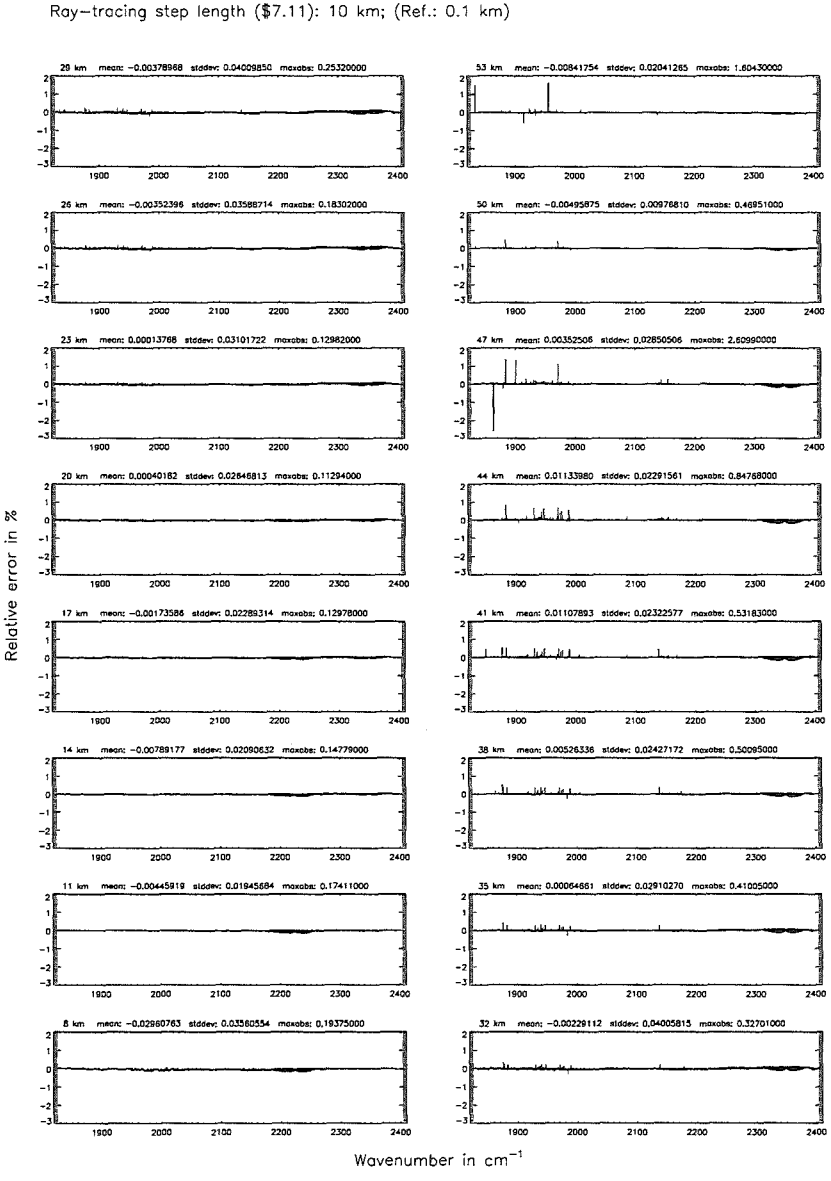


Figure 259: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

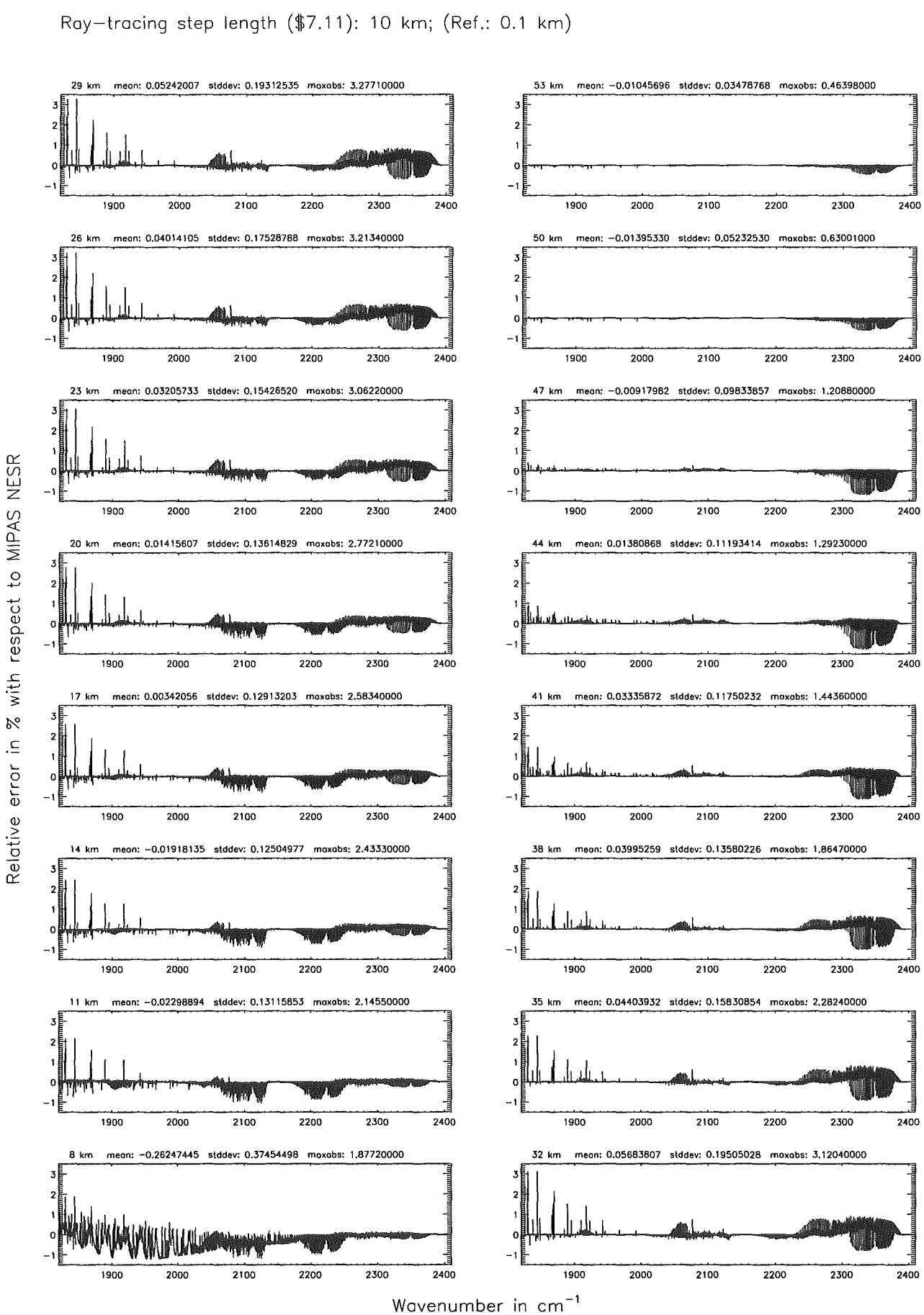
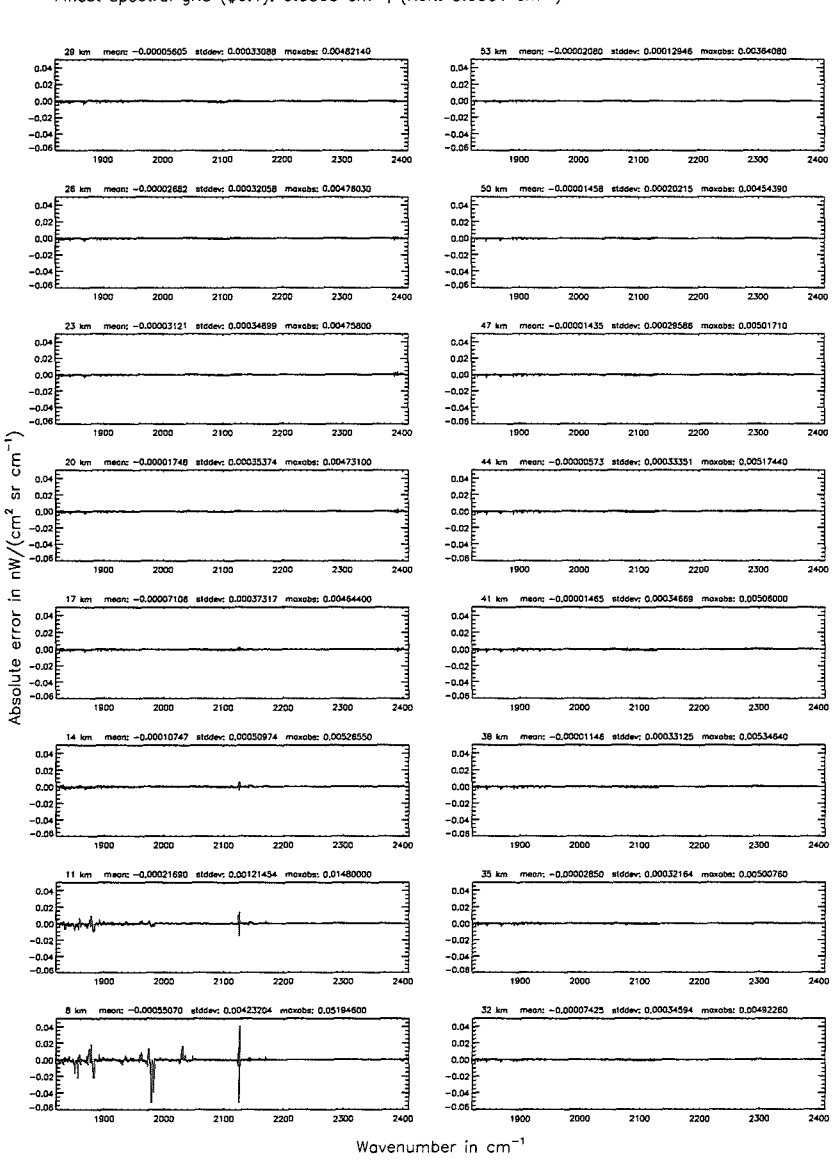


Figure 260: Relative error [%] with respect to MIPAS NESR



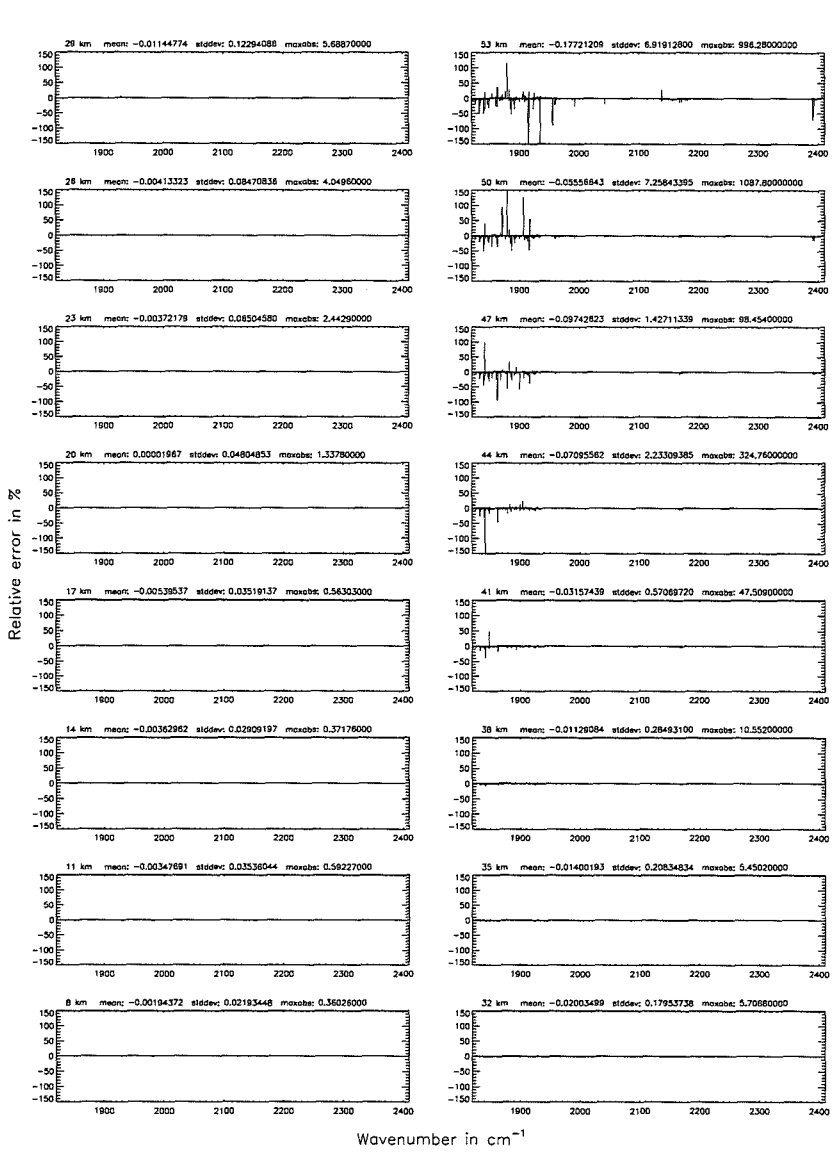


Figure 261: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

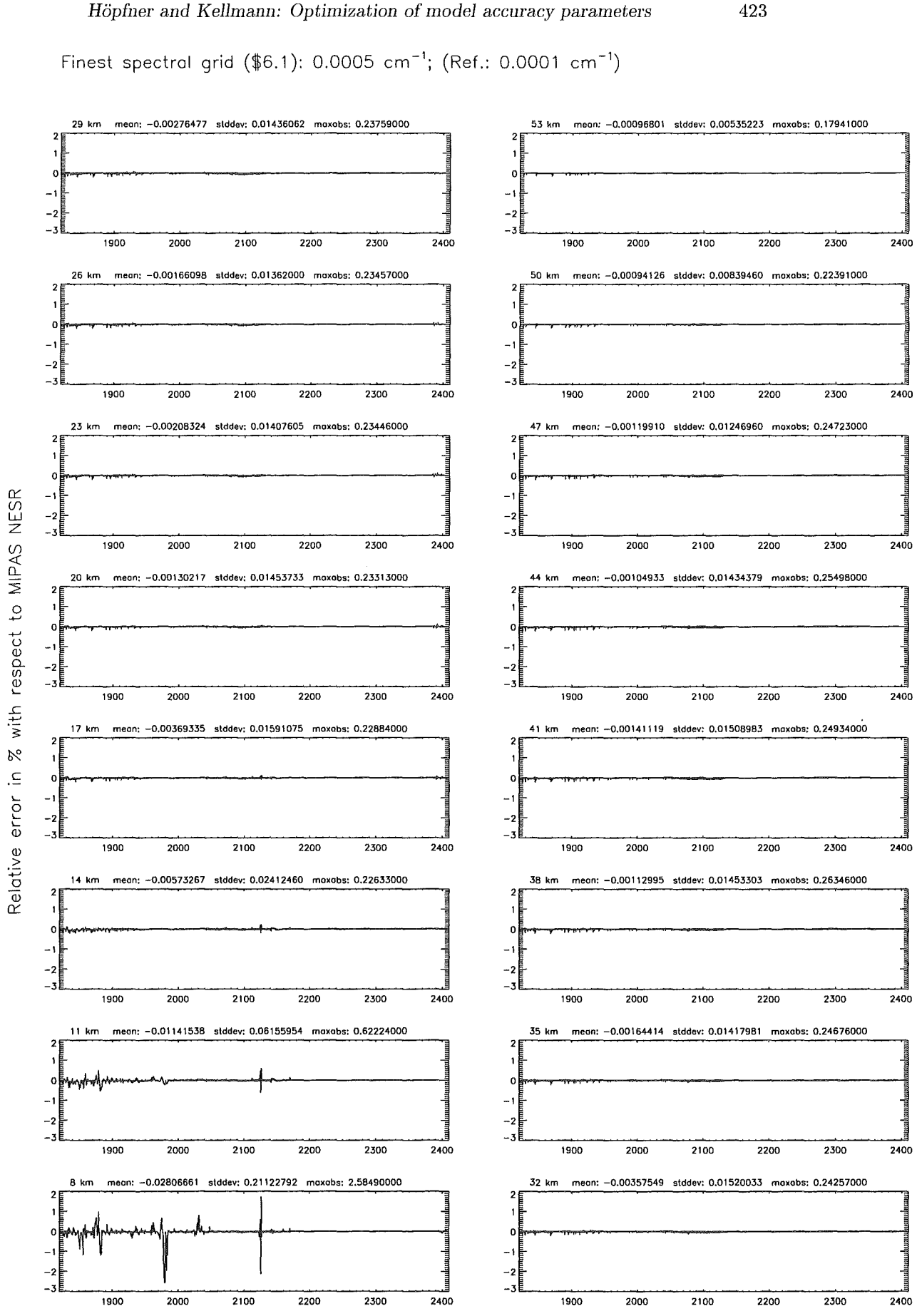


Figure 262: Relative error [%] with respect to MIPAS NESR

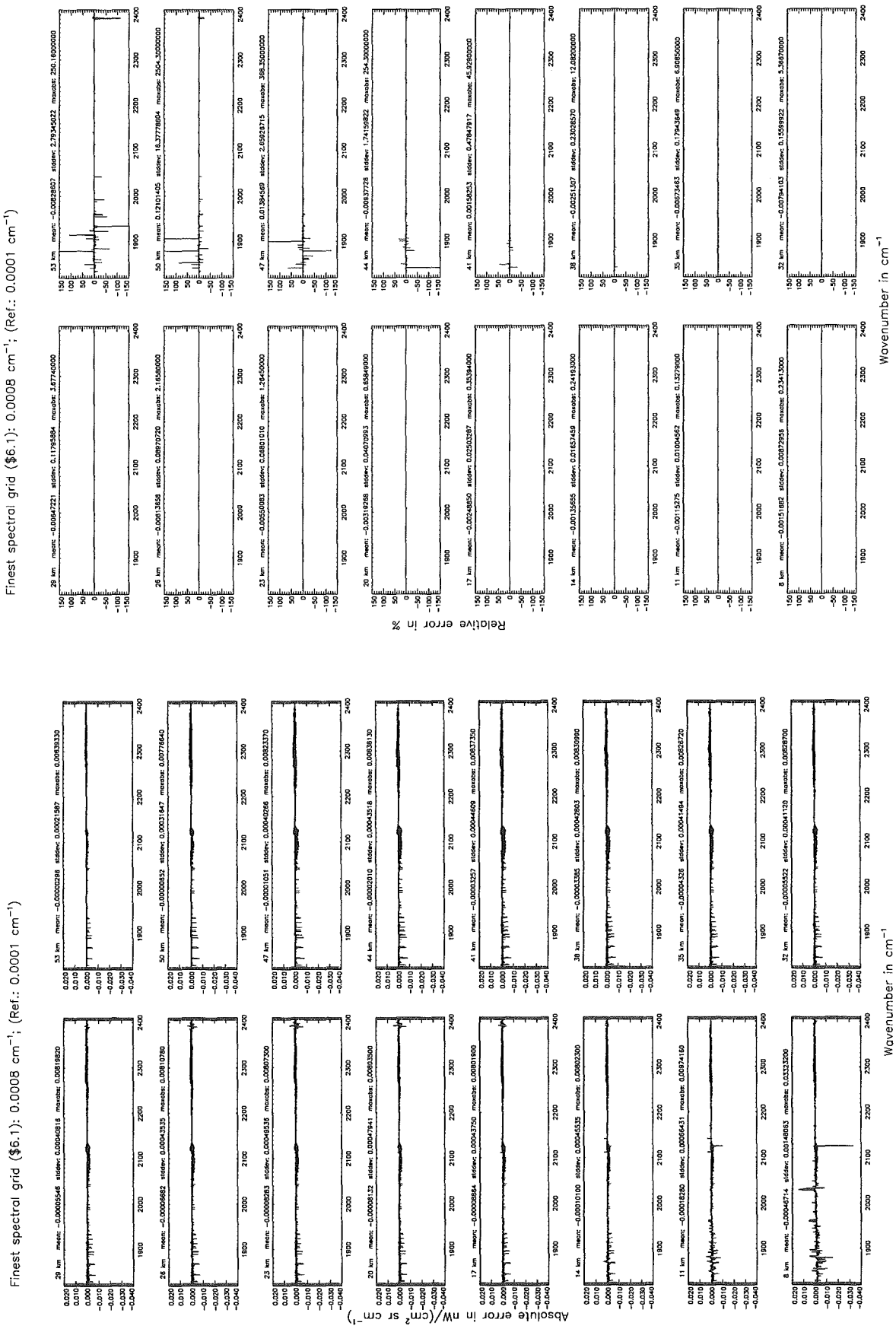


Figure 263: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

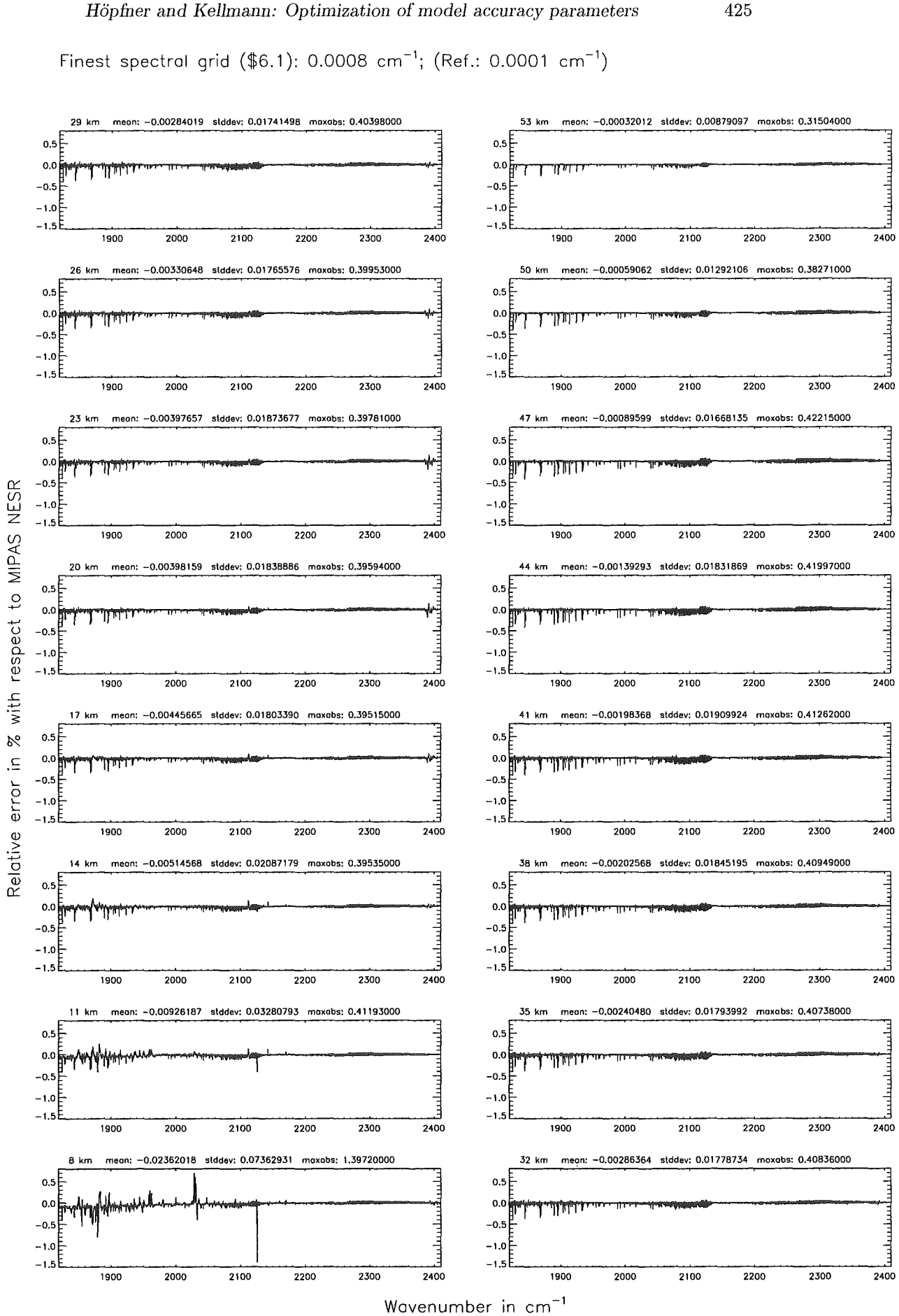


Figure 264: Relative error [%] with respect to MIPAS NESR

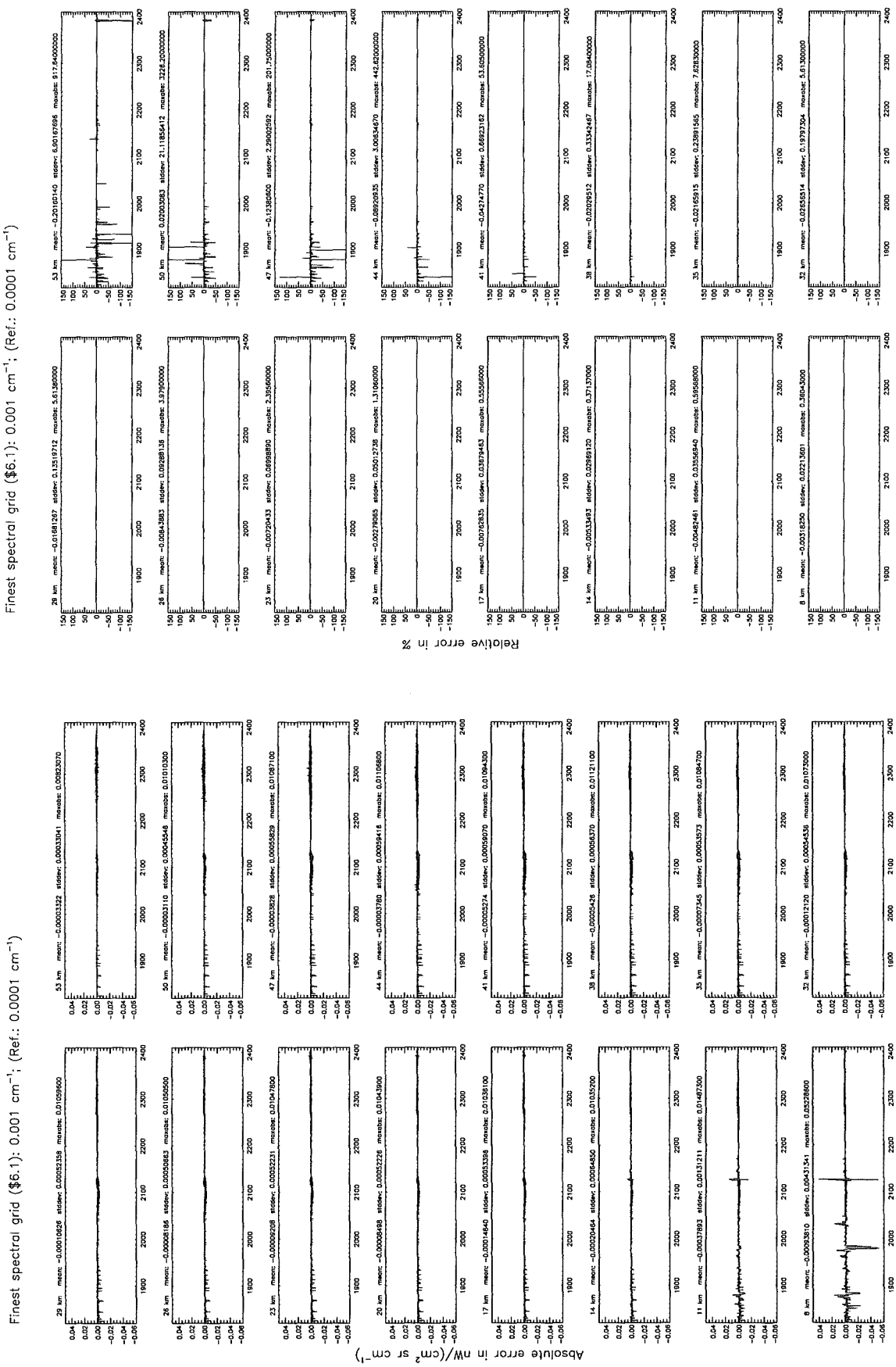


Figure 265: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

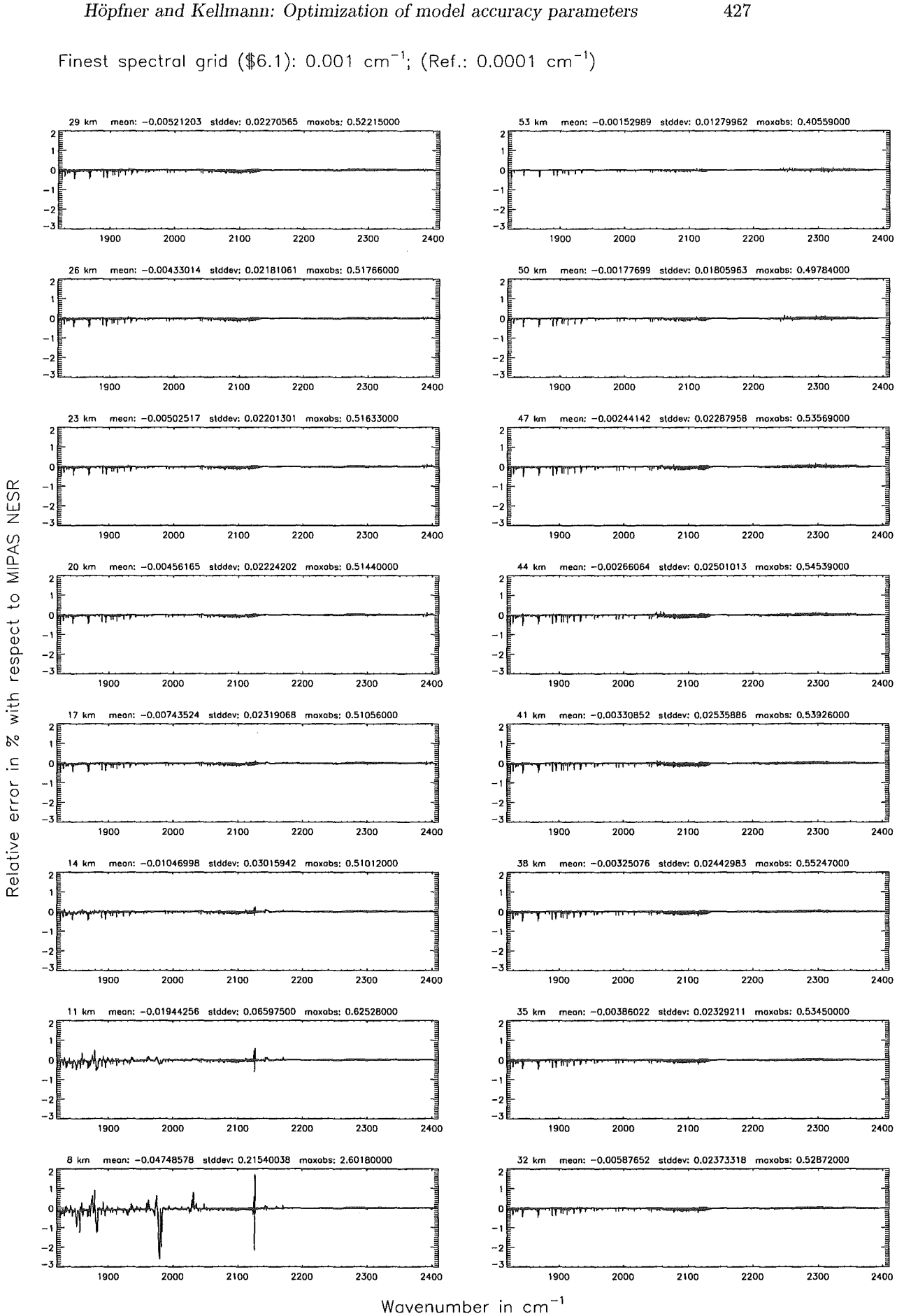


Figure 266: Relative error [%] with respect to MIPAS NESR

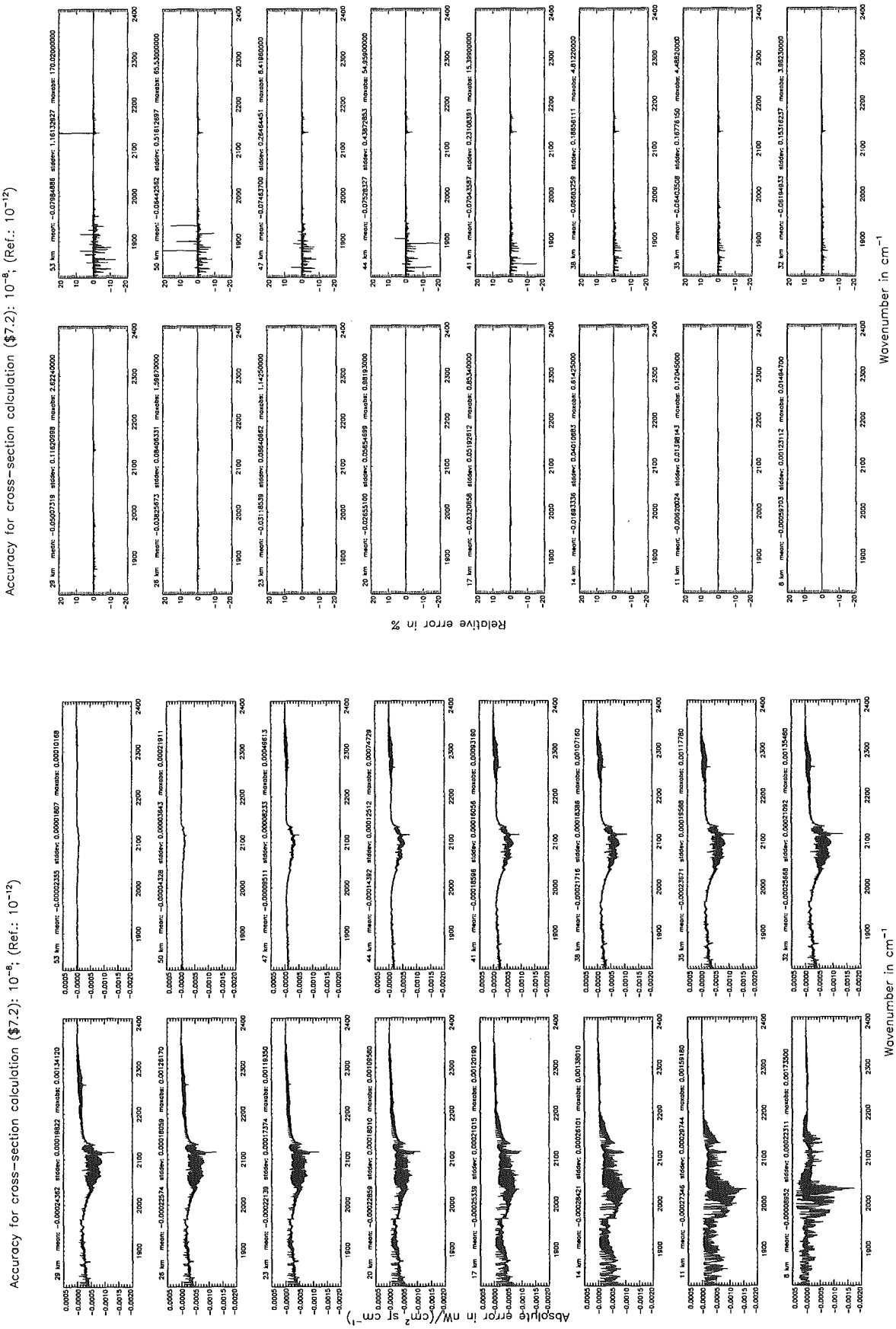


Figure 267: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

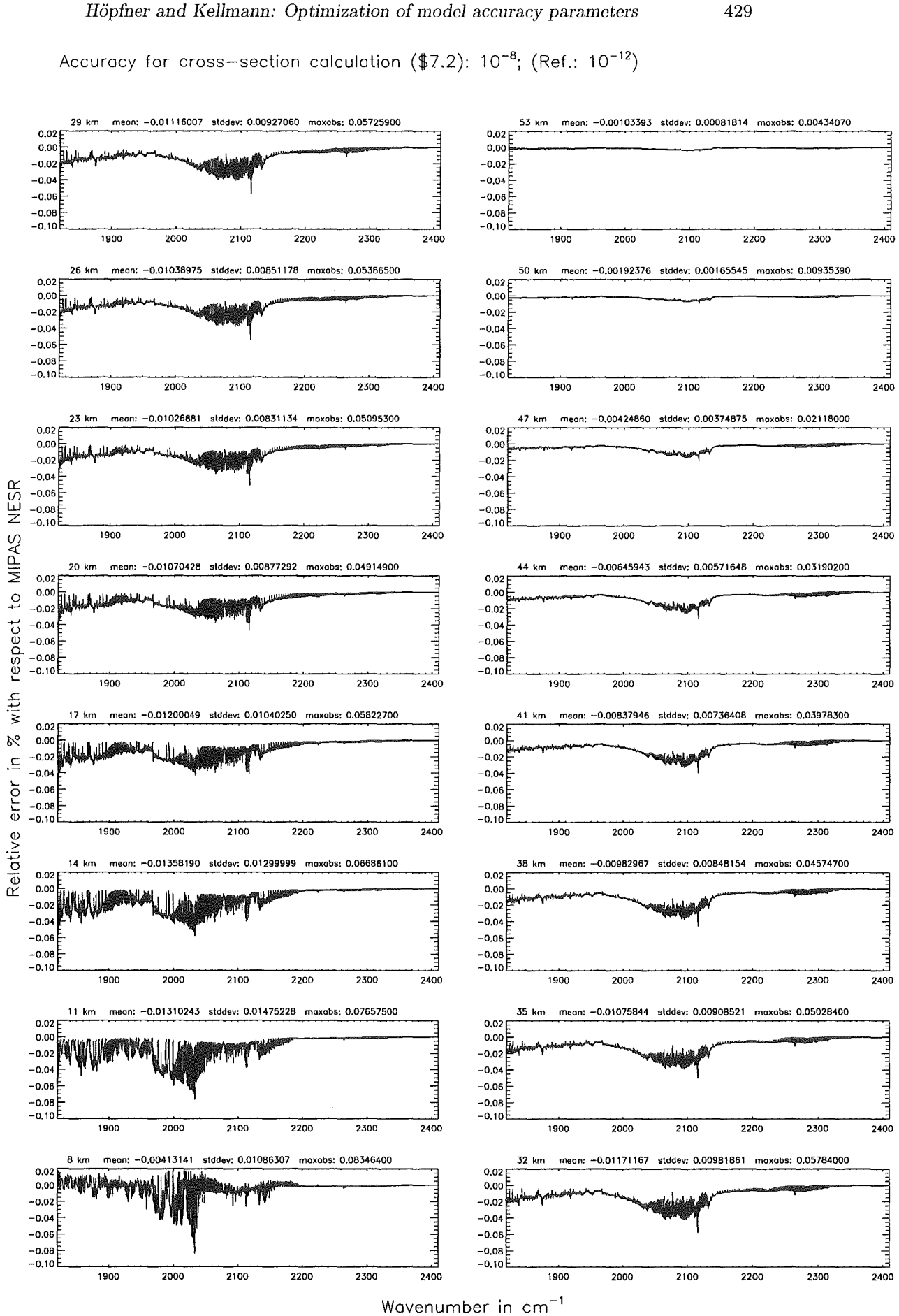
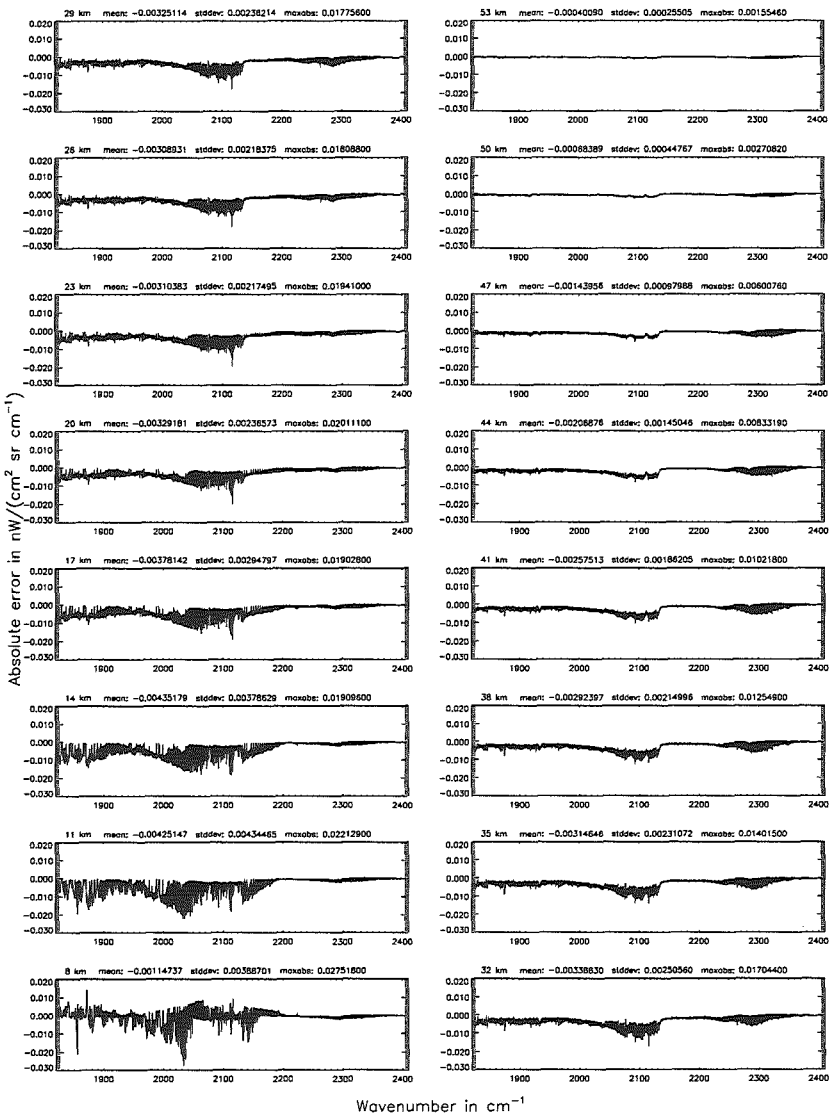


Figure 268: Relative error [%] with respect to MIPAS NESR



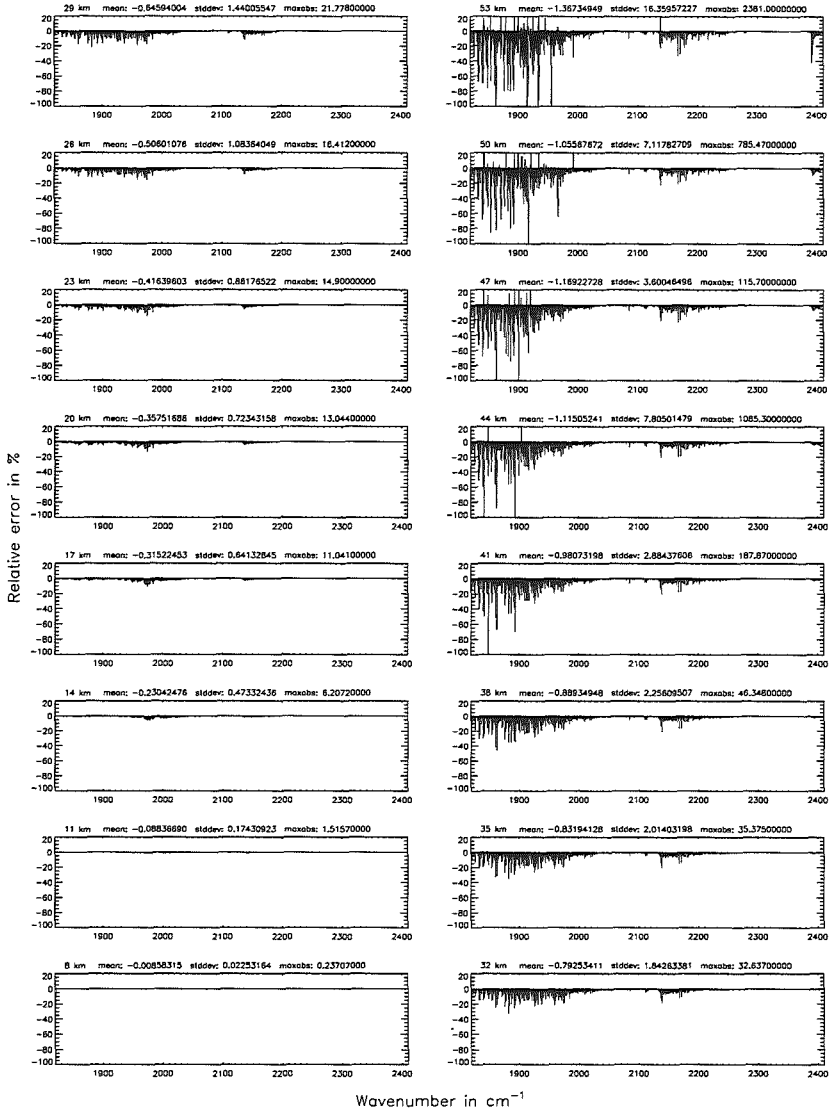


Figure 269: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

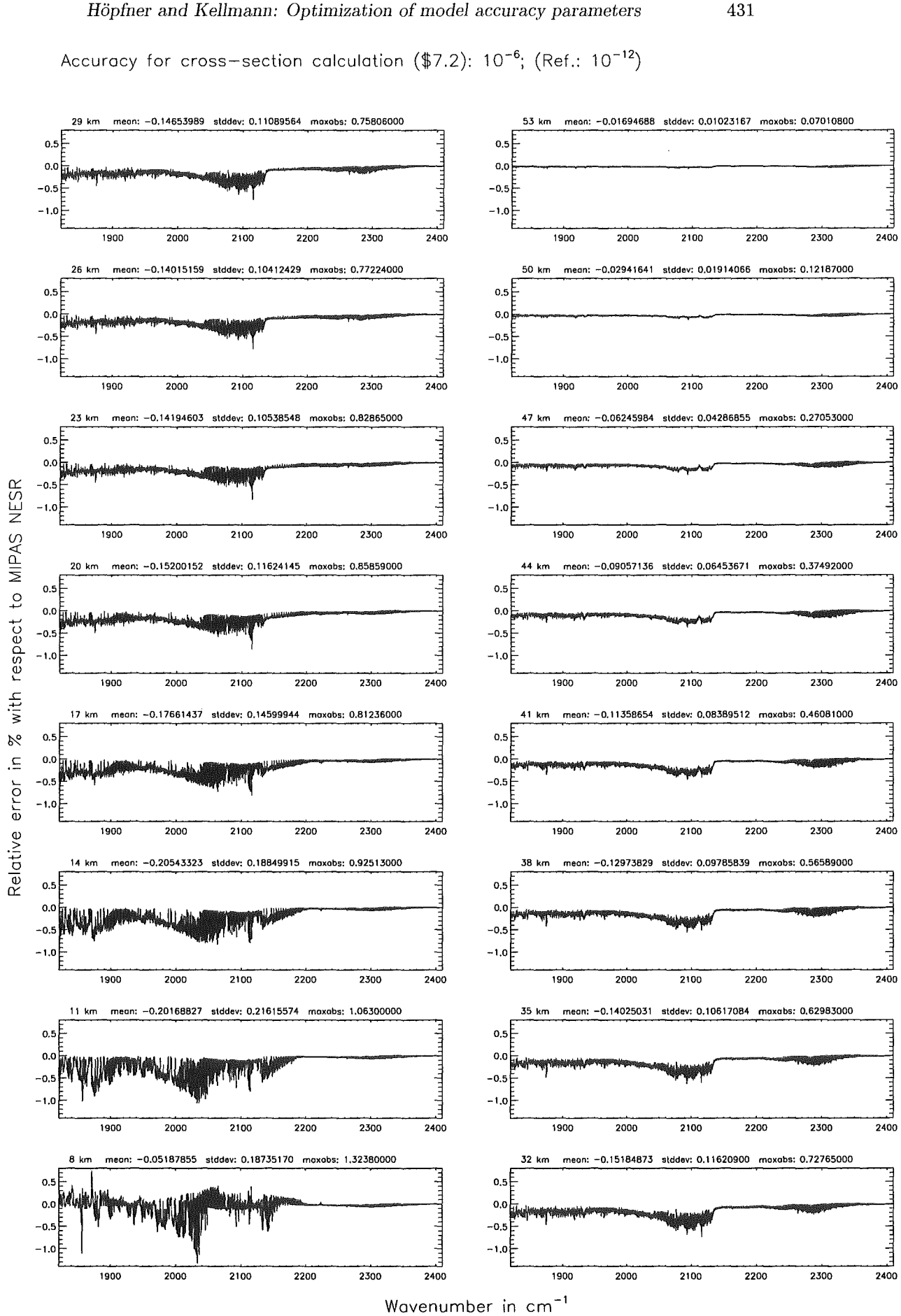
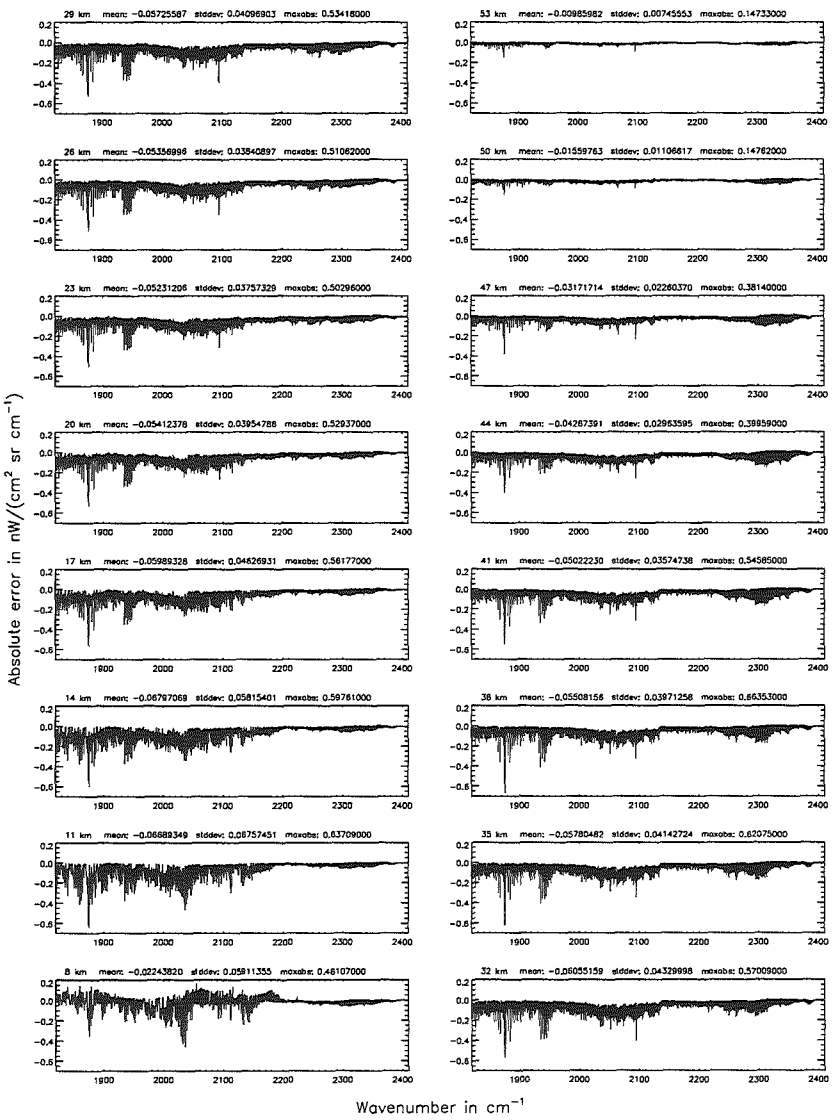


Figure 270: Relative error [%] with respect to MIPAS NESR



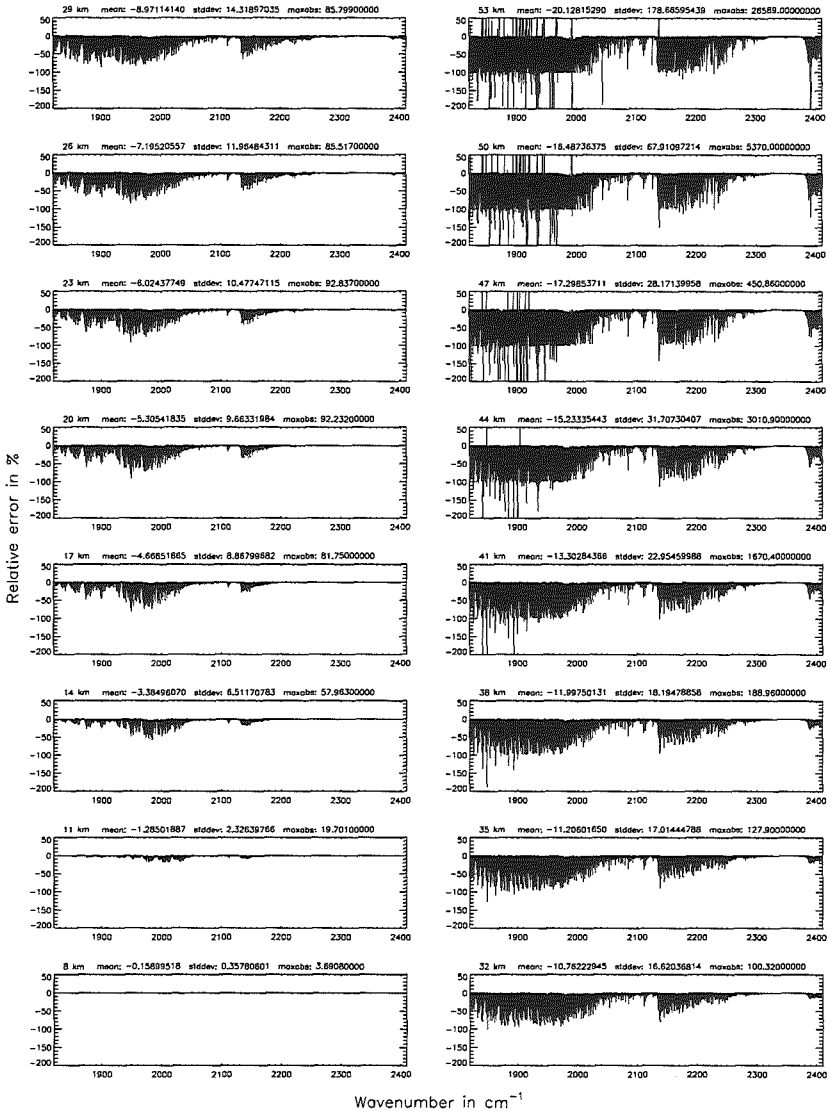


Figure 271: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

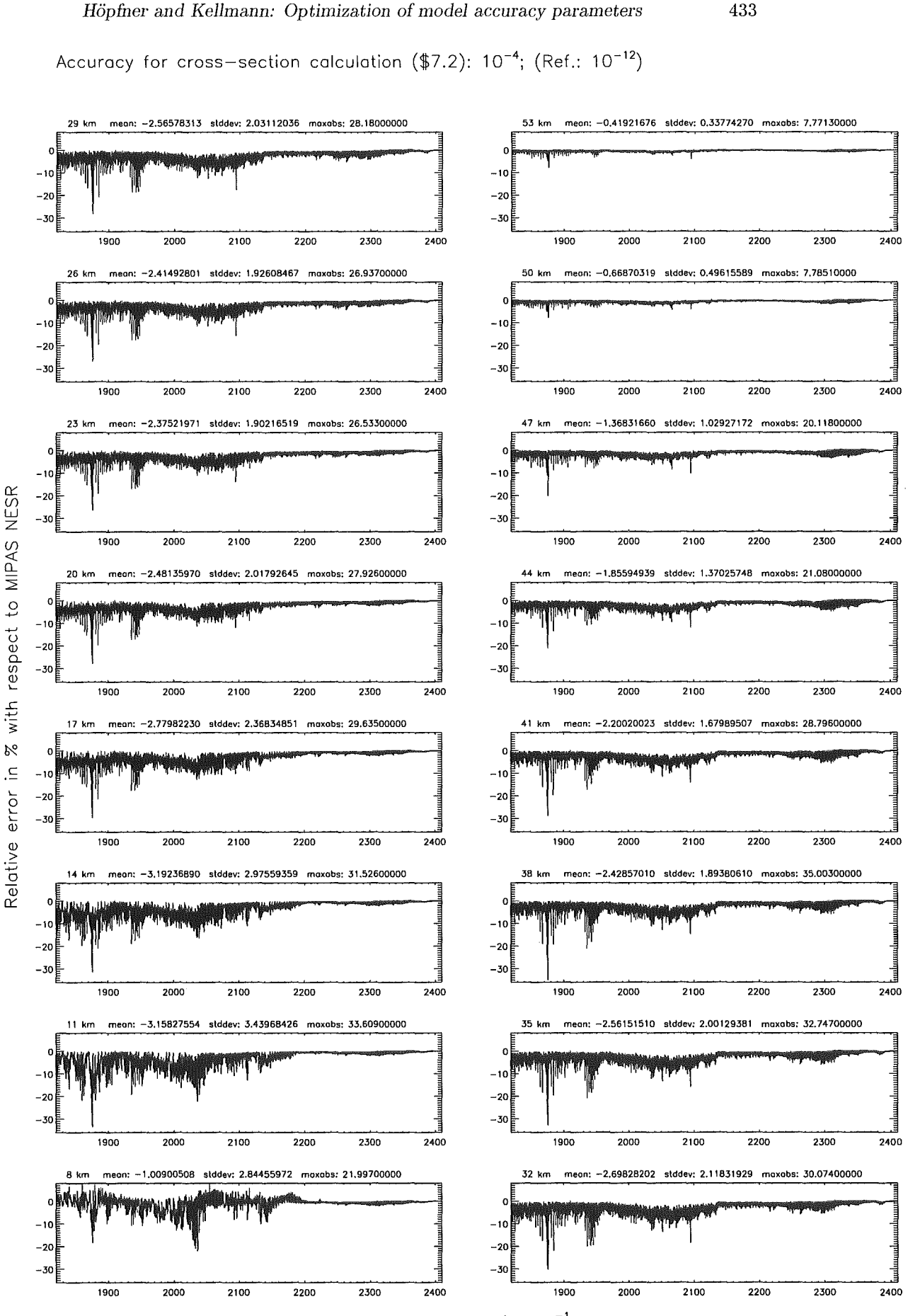
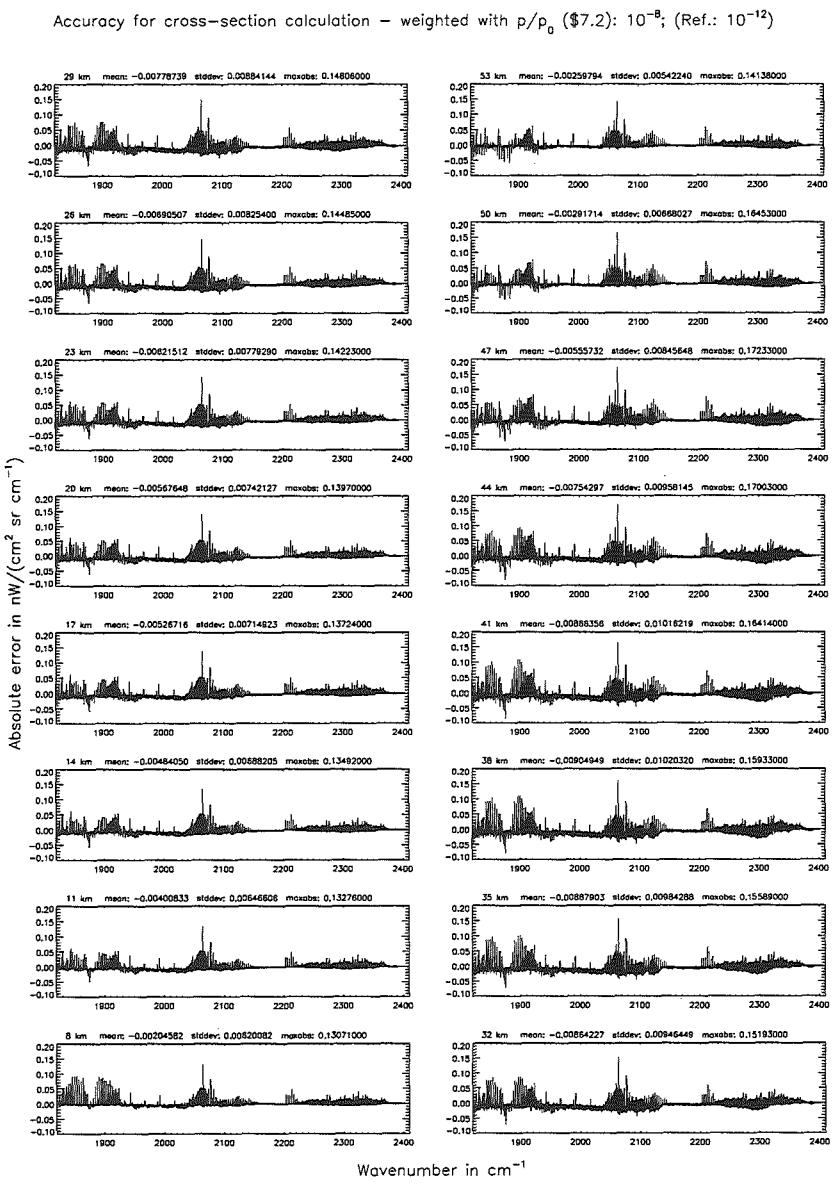


Figure 272: Relative error [%] with respect to MIPAS NESR



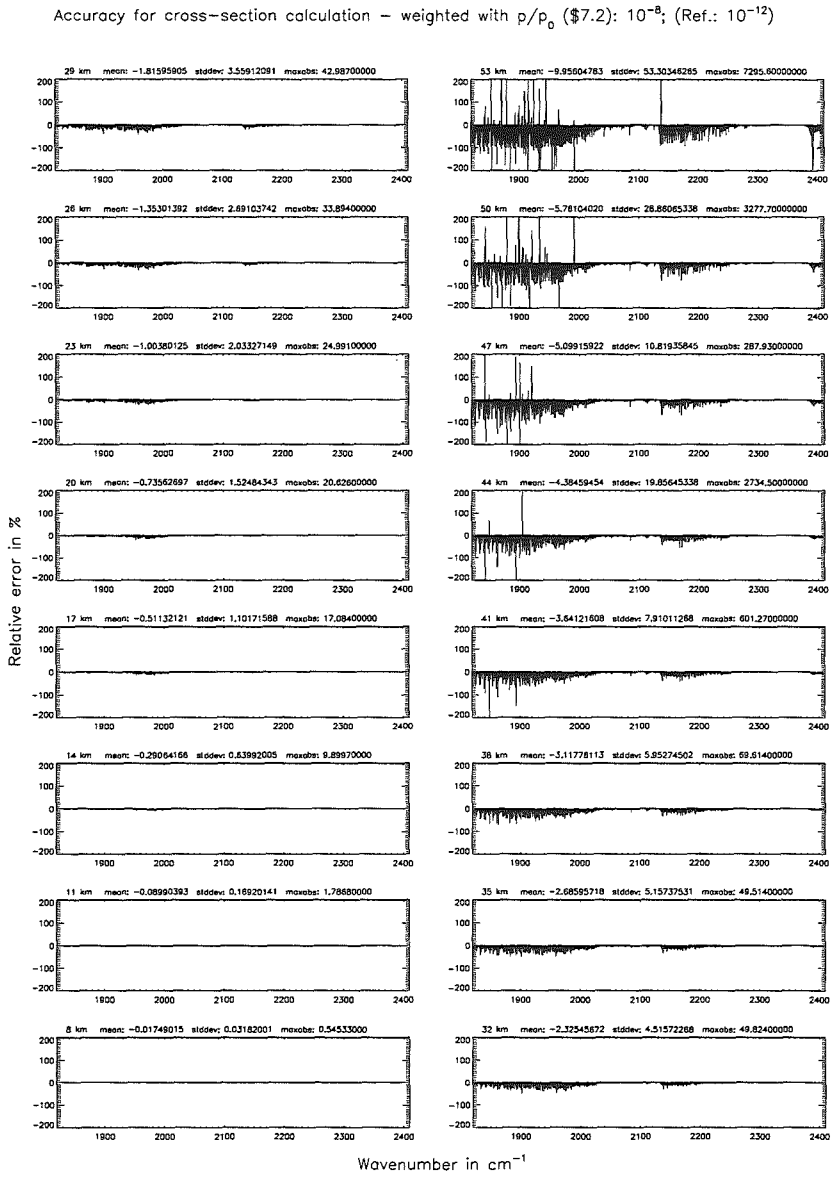


Figure 273: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

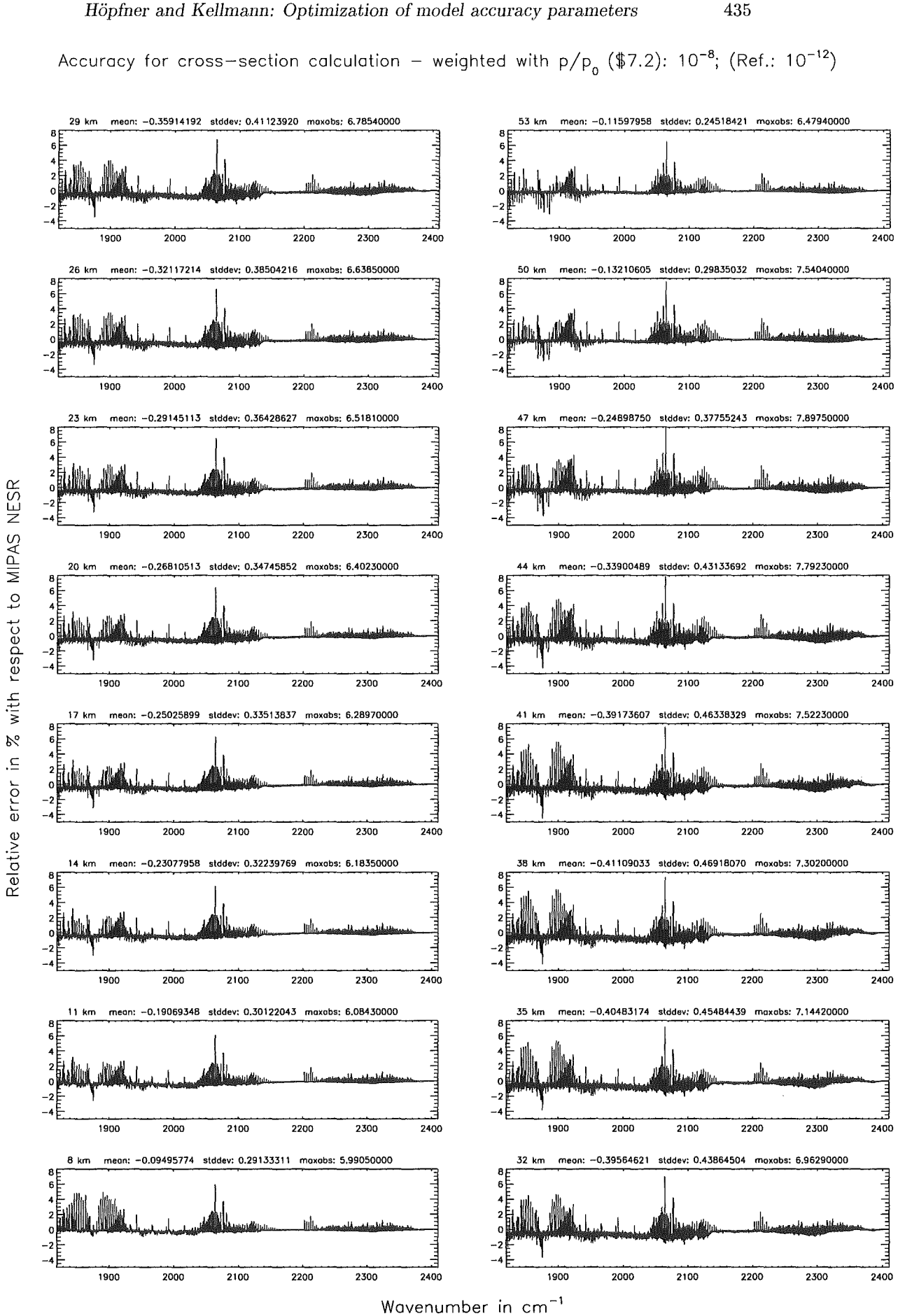
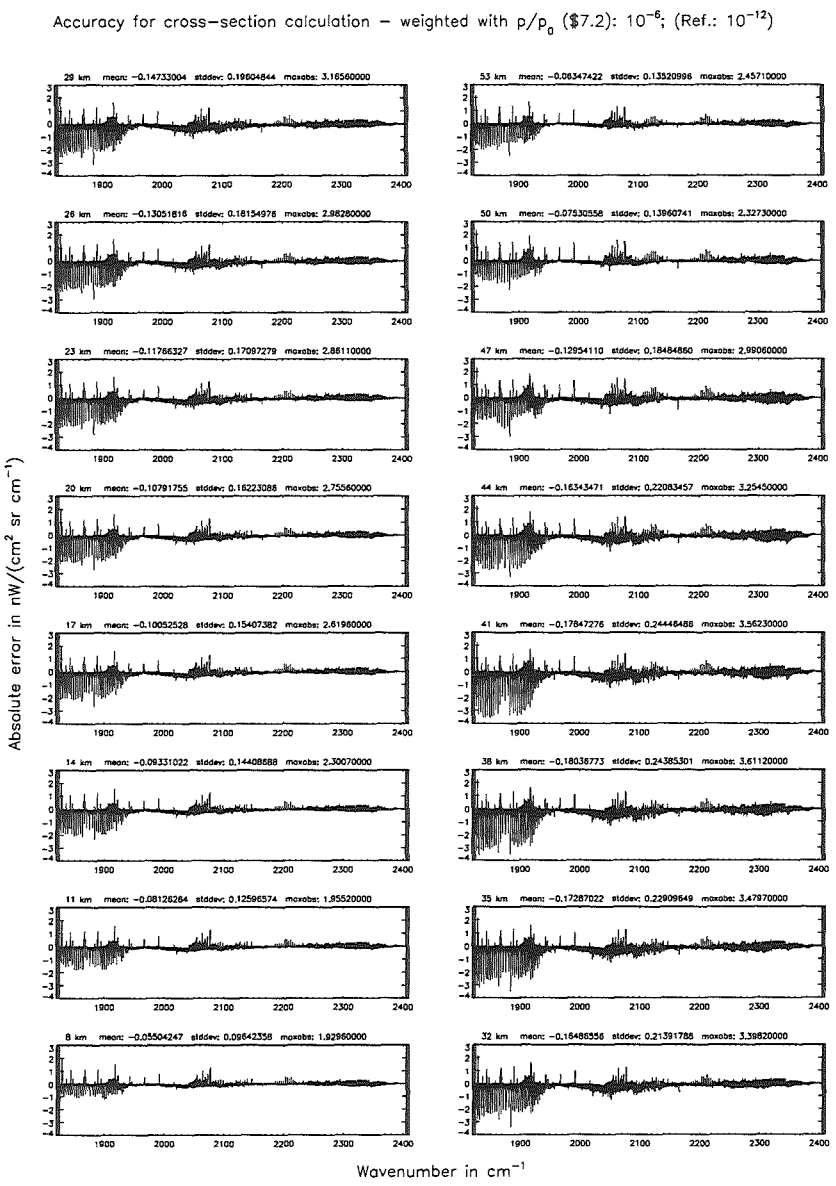


Figure 274: Relative error [%] with respect to MIPAS NESR



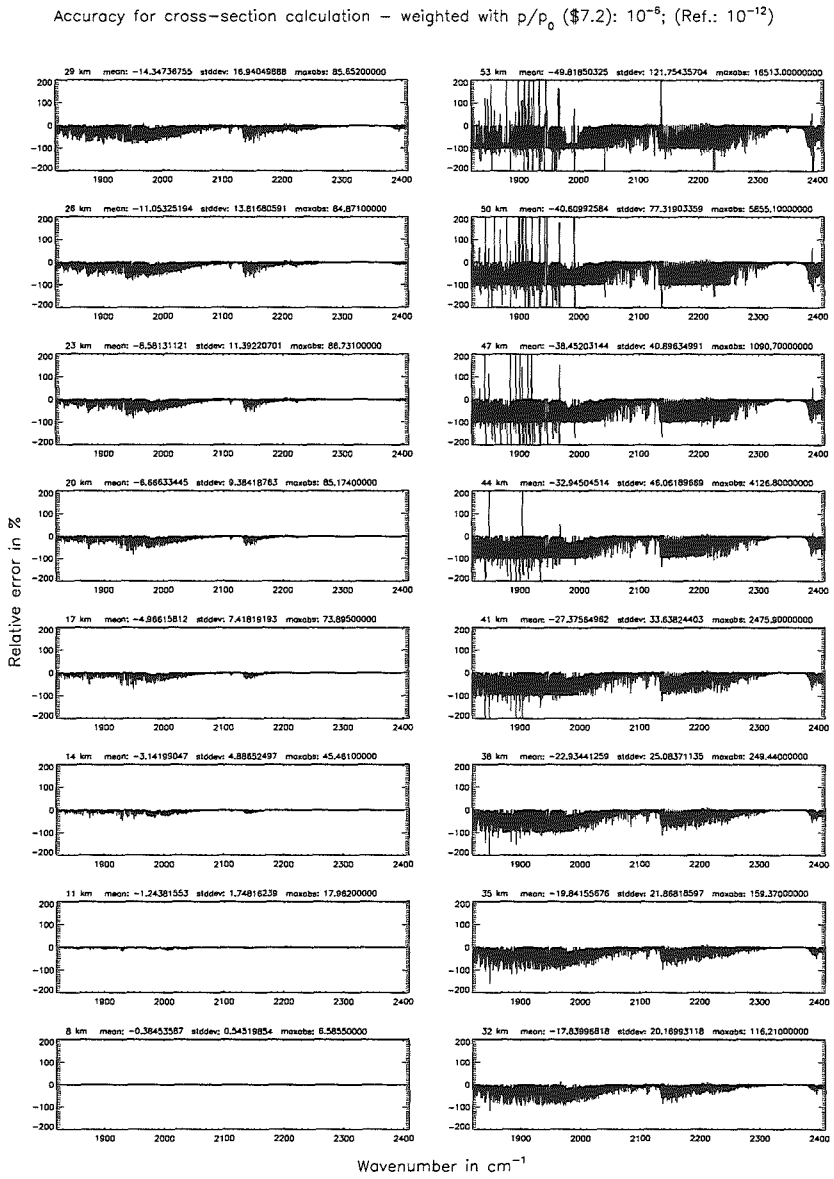


Figure 275: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

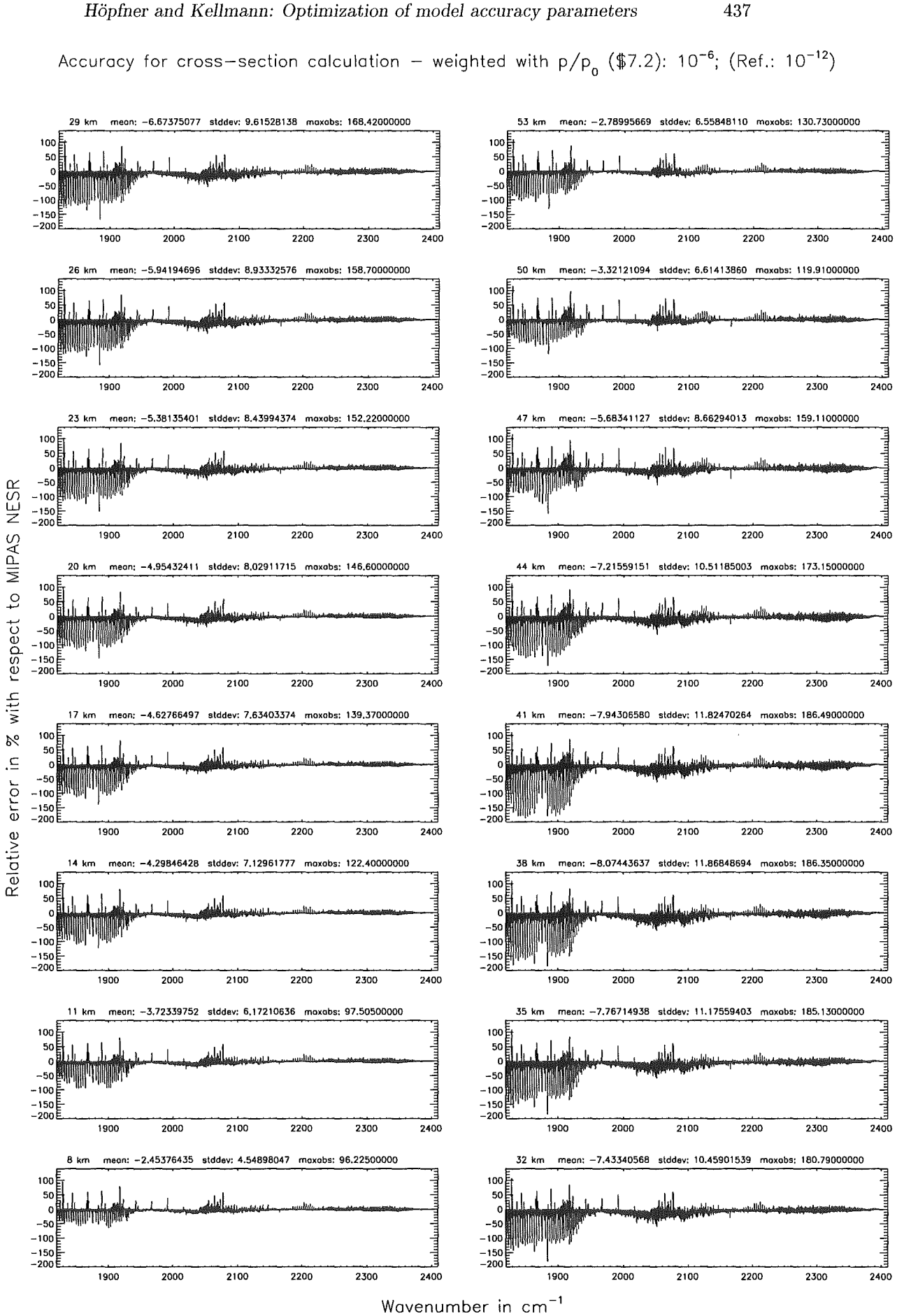
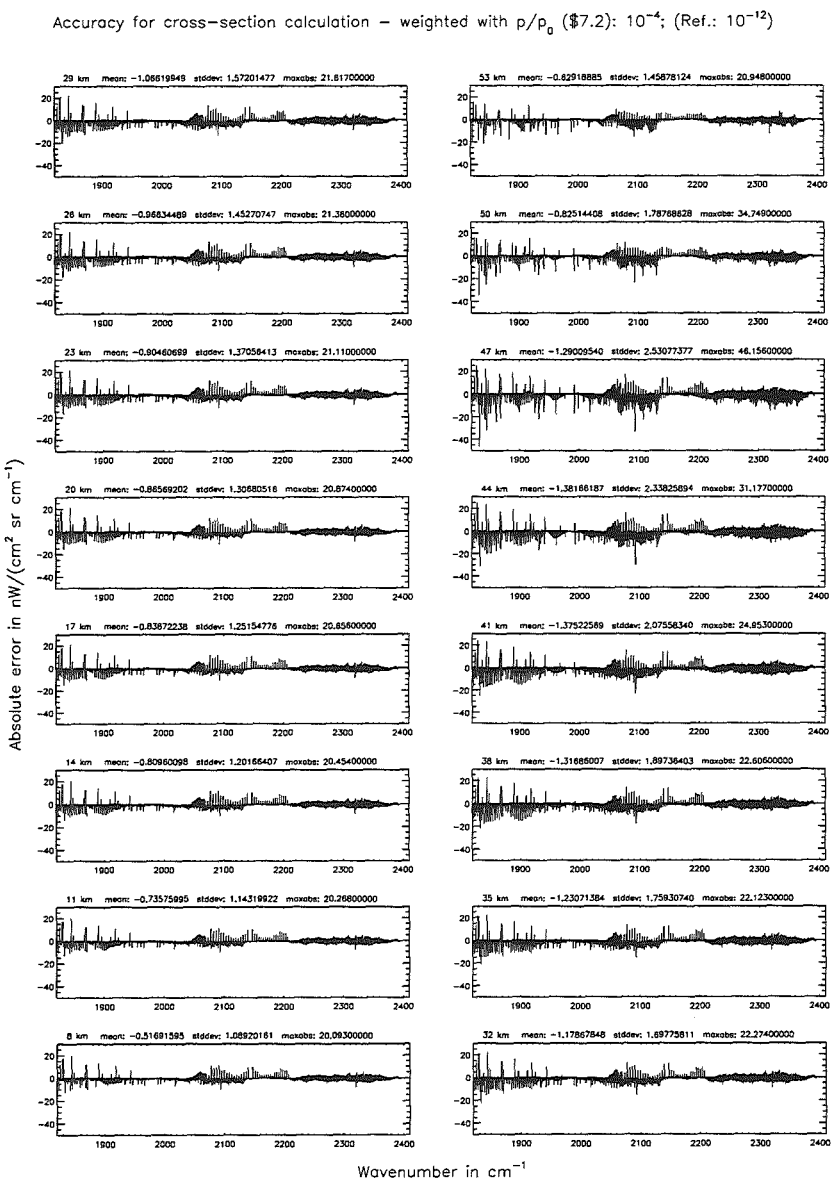


Figure 276: Relative error [%] with respect to MIPAS NESR



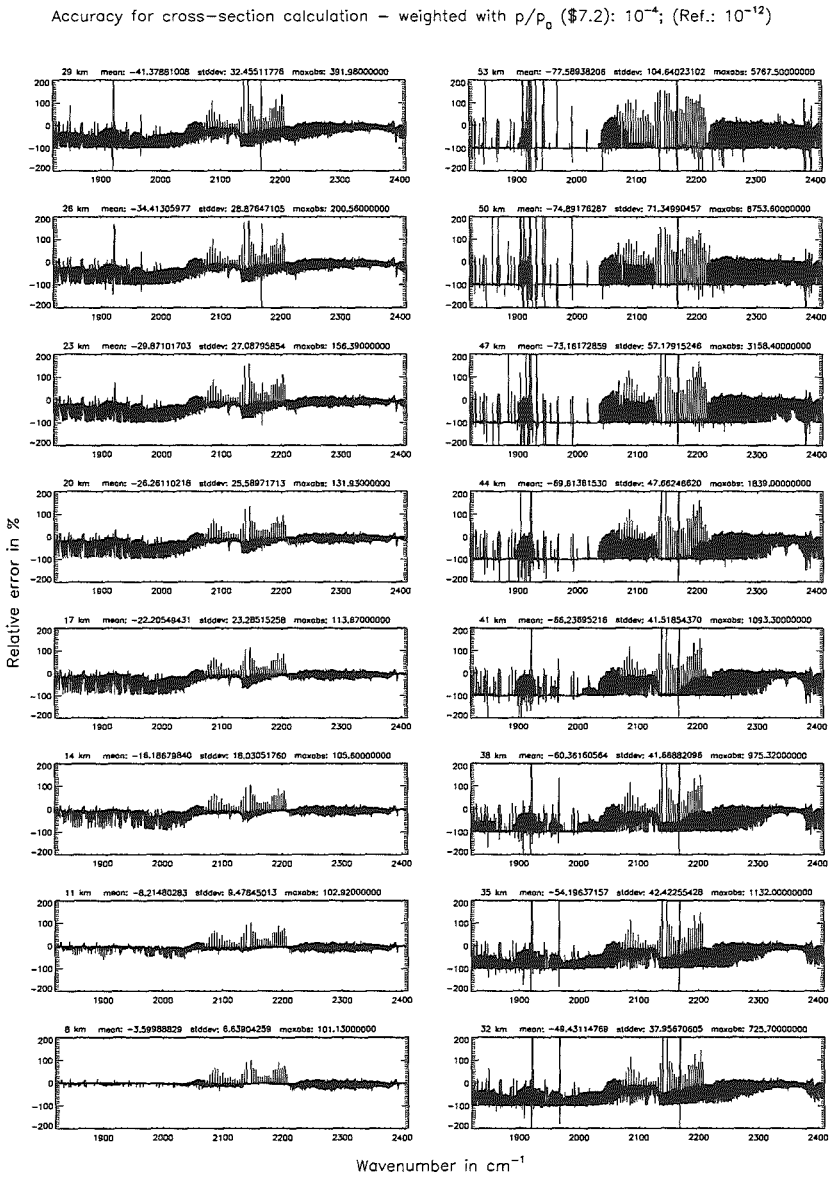


Figure 277: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

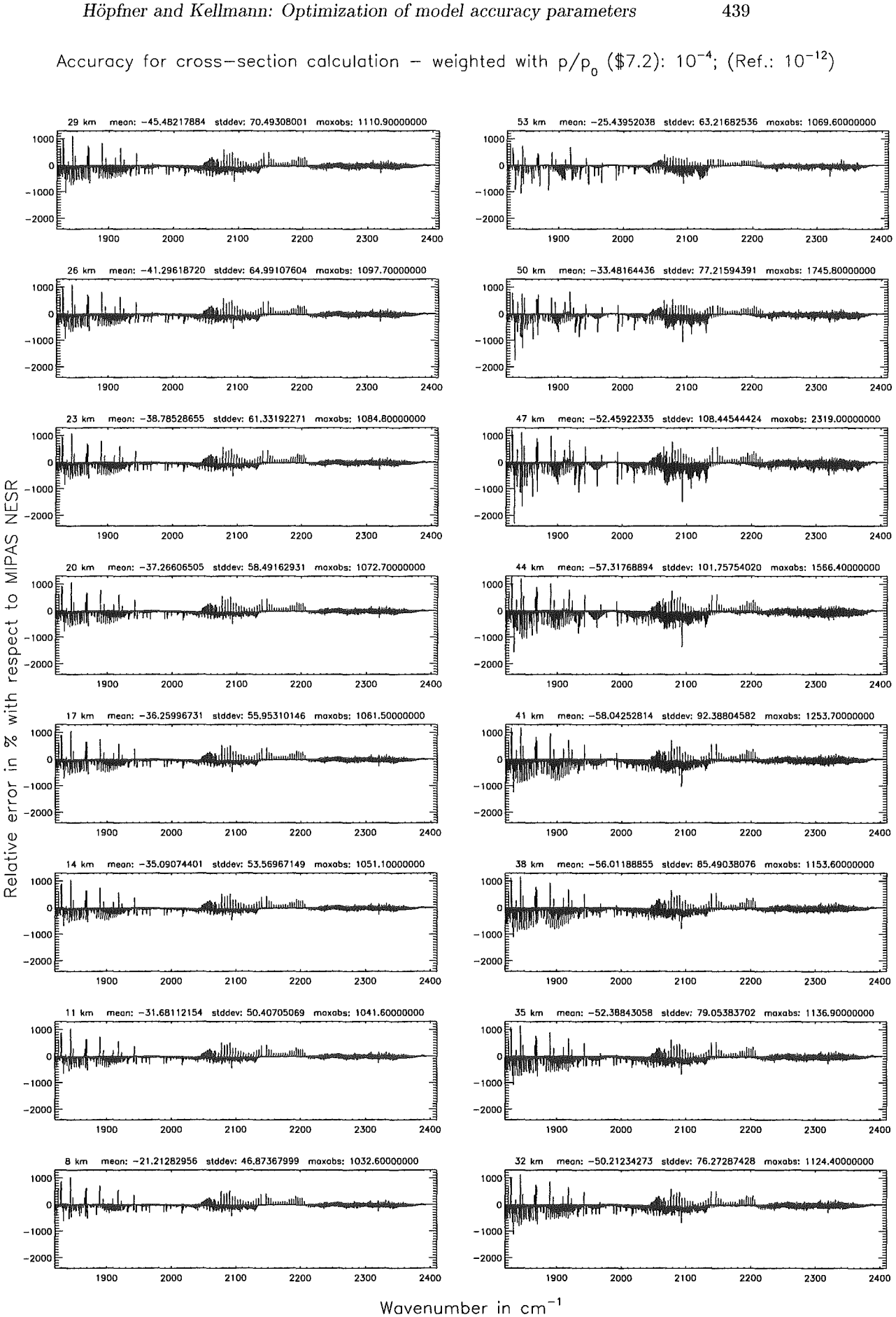


Figure 278: Relative error [%] with respect to MIPAS NESR

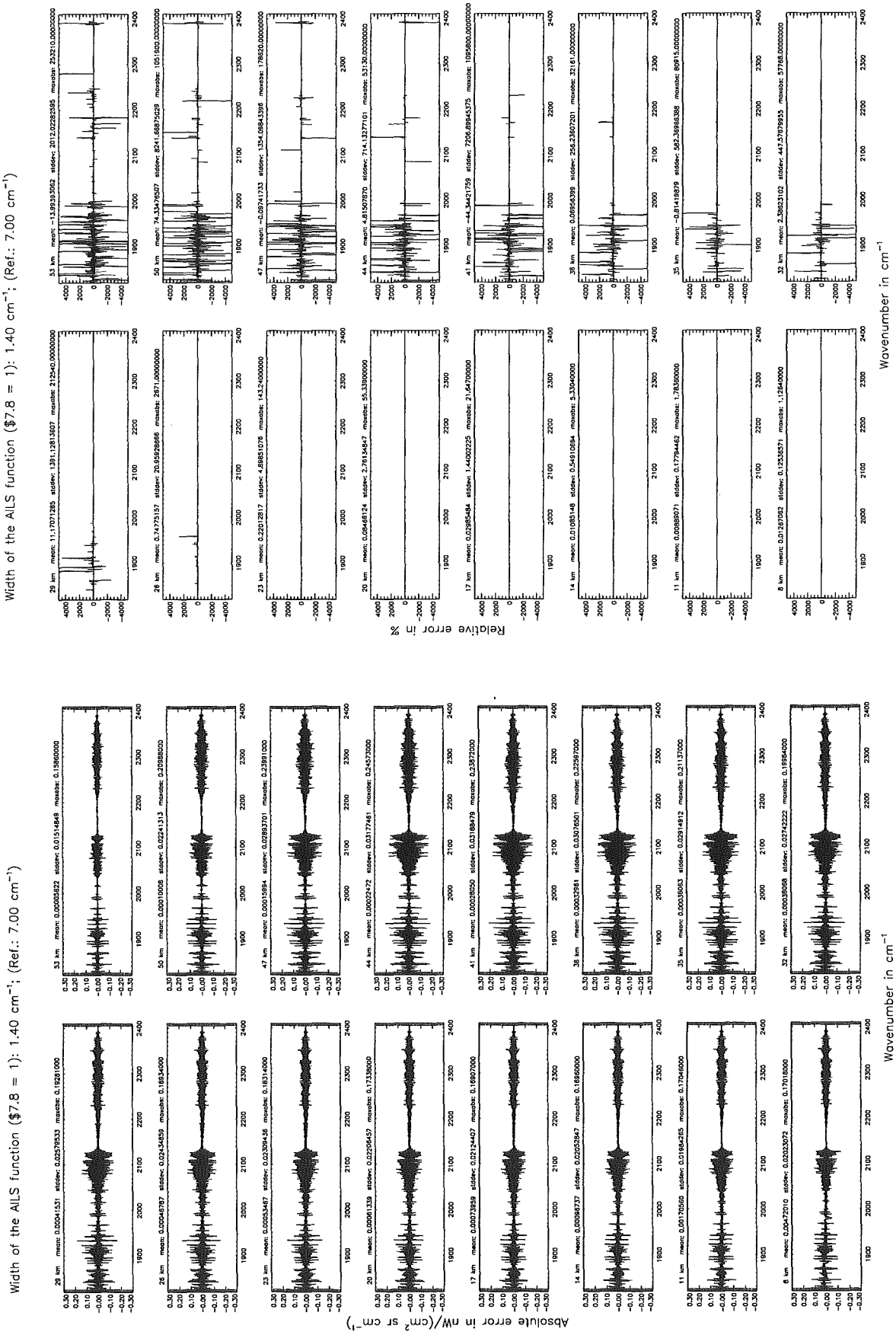


Figure 279: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

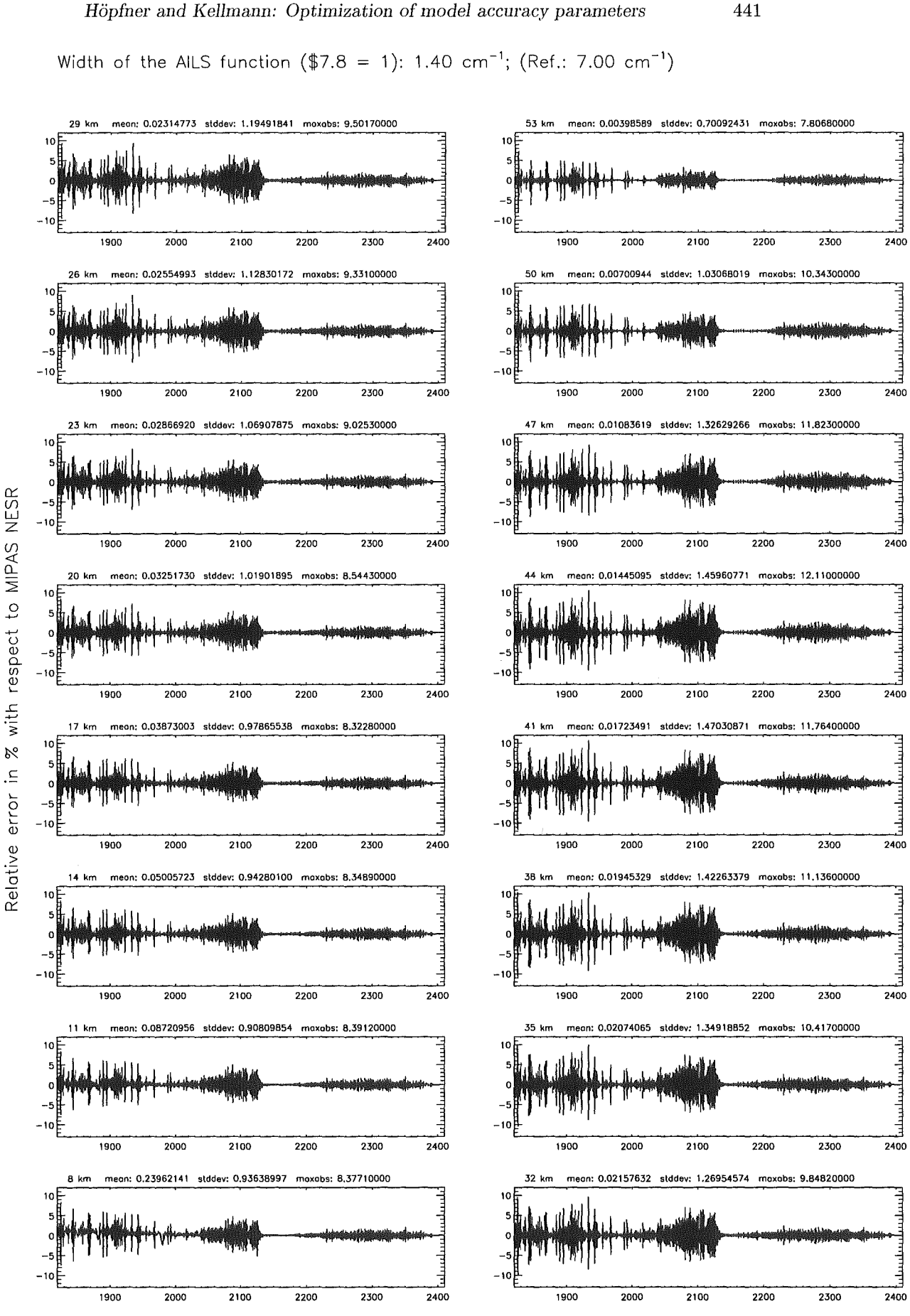


Figure 280: Relative error [%] with respect to MIPAS NESR

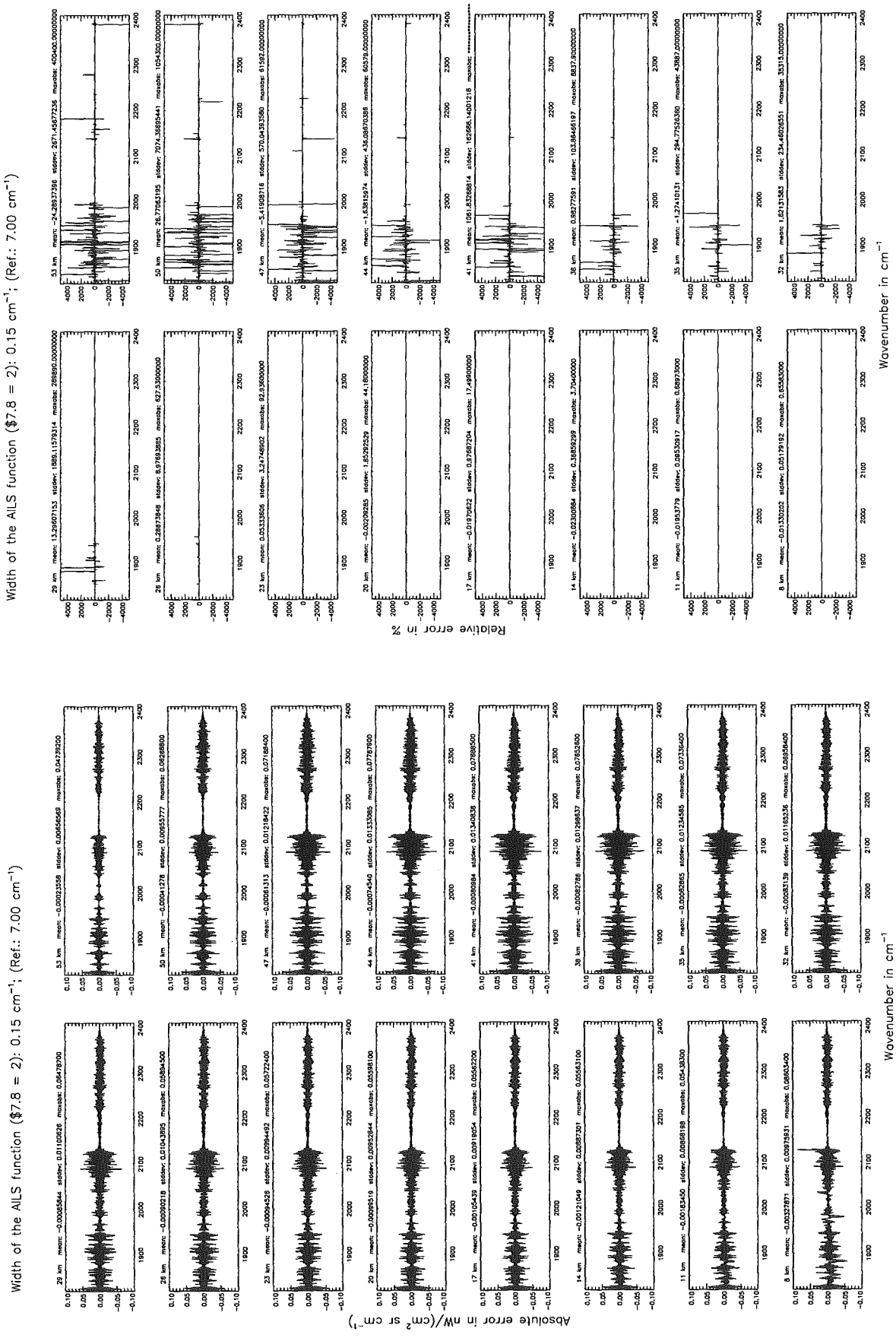


Figure 281: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

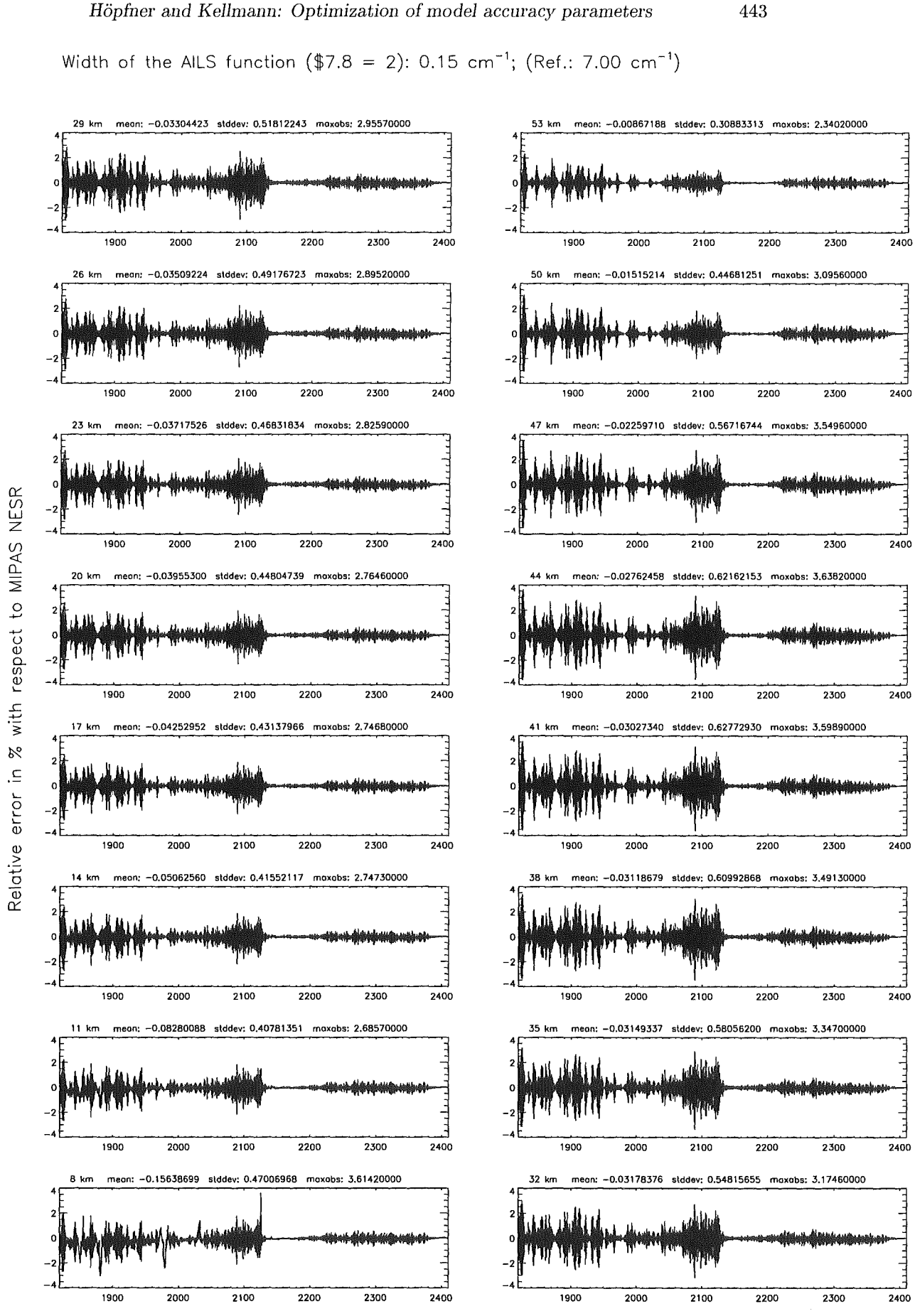
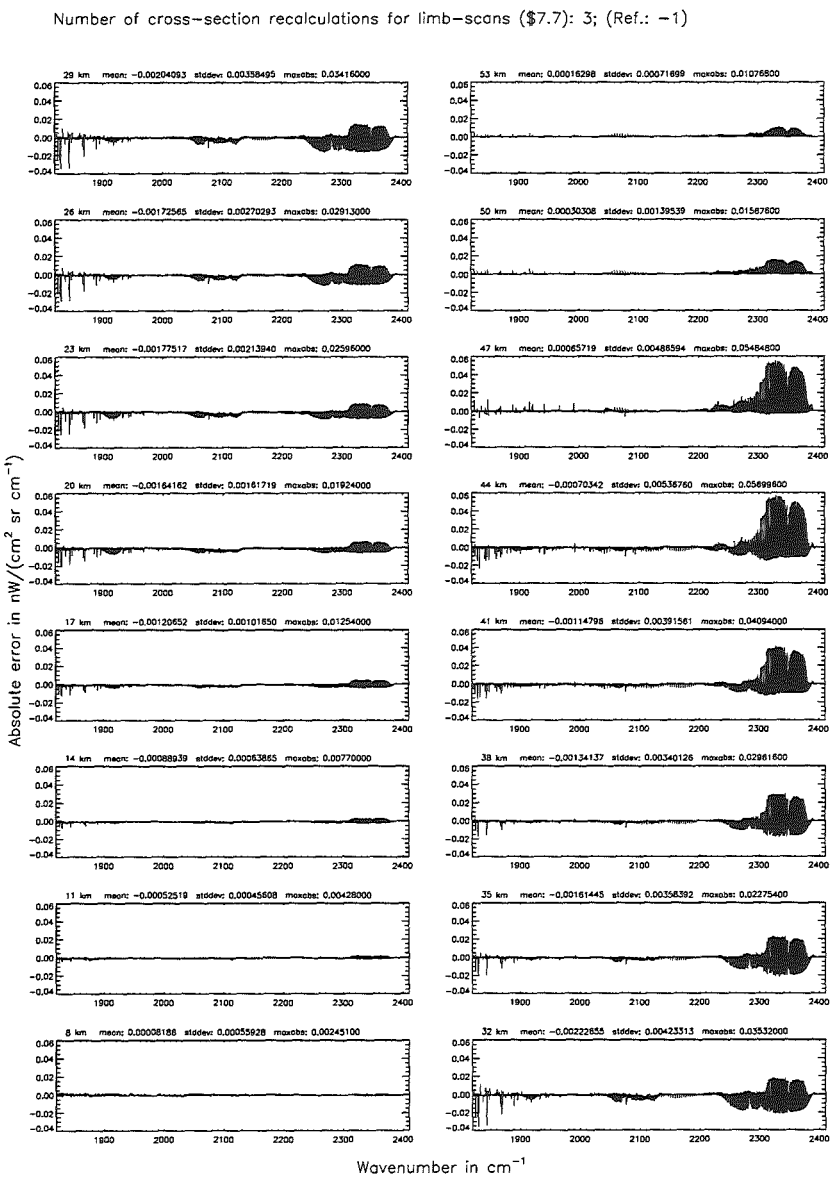


Figure 282: Relative error [%] with respect to MIPAS NESR



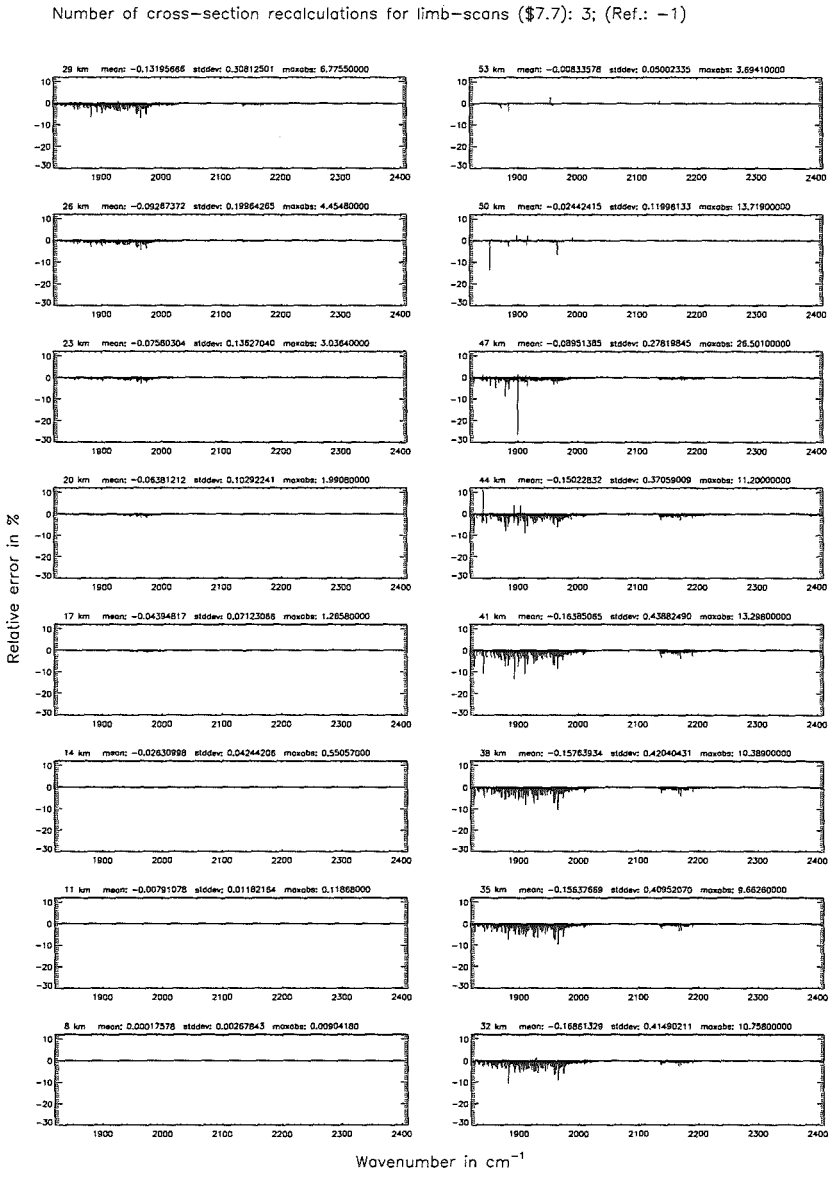


Figure 283: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

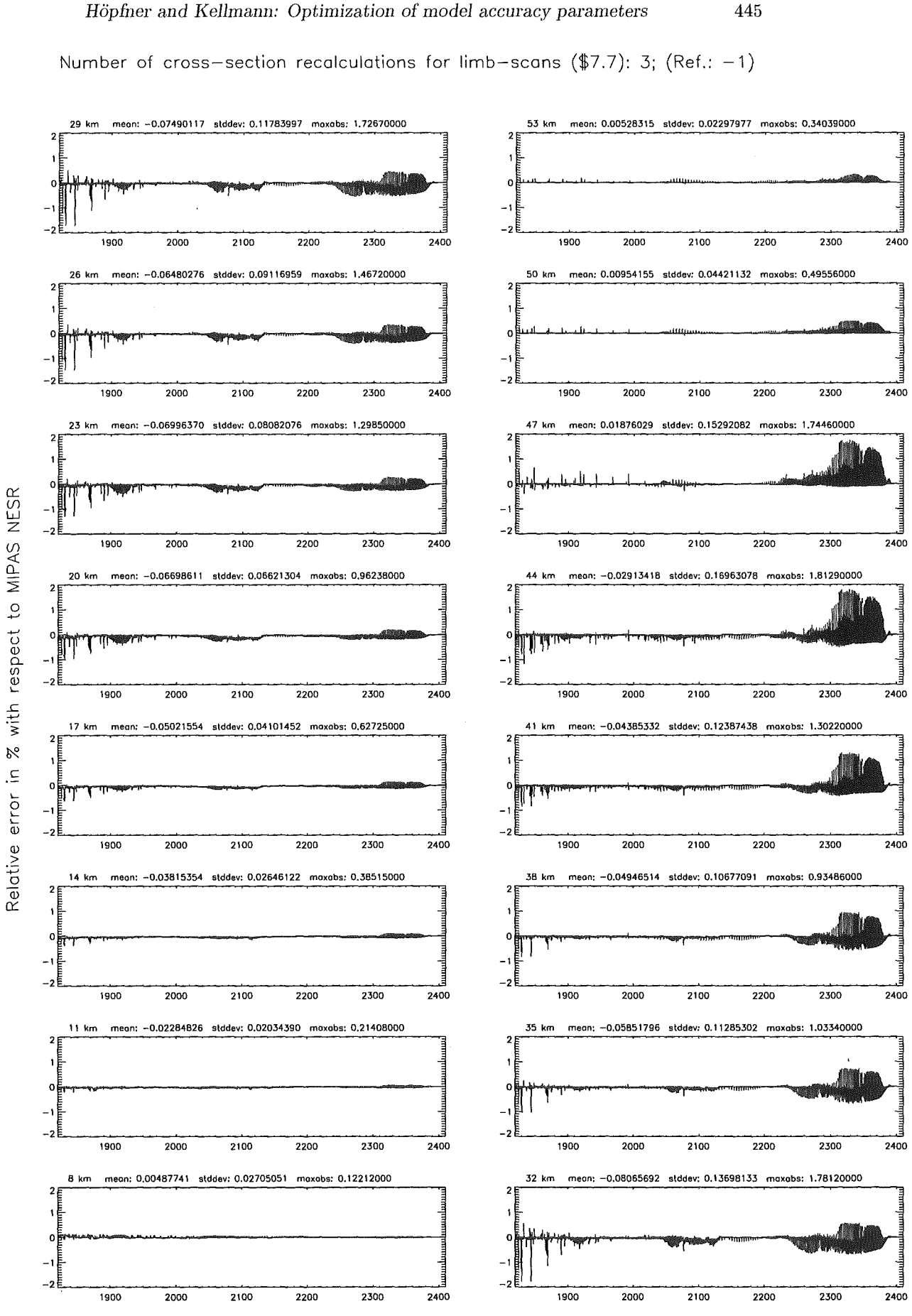
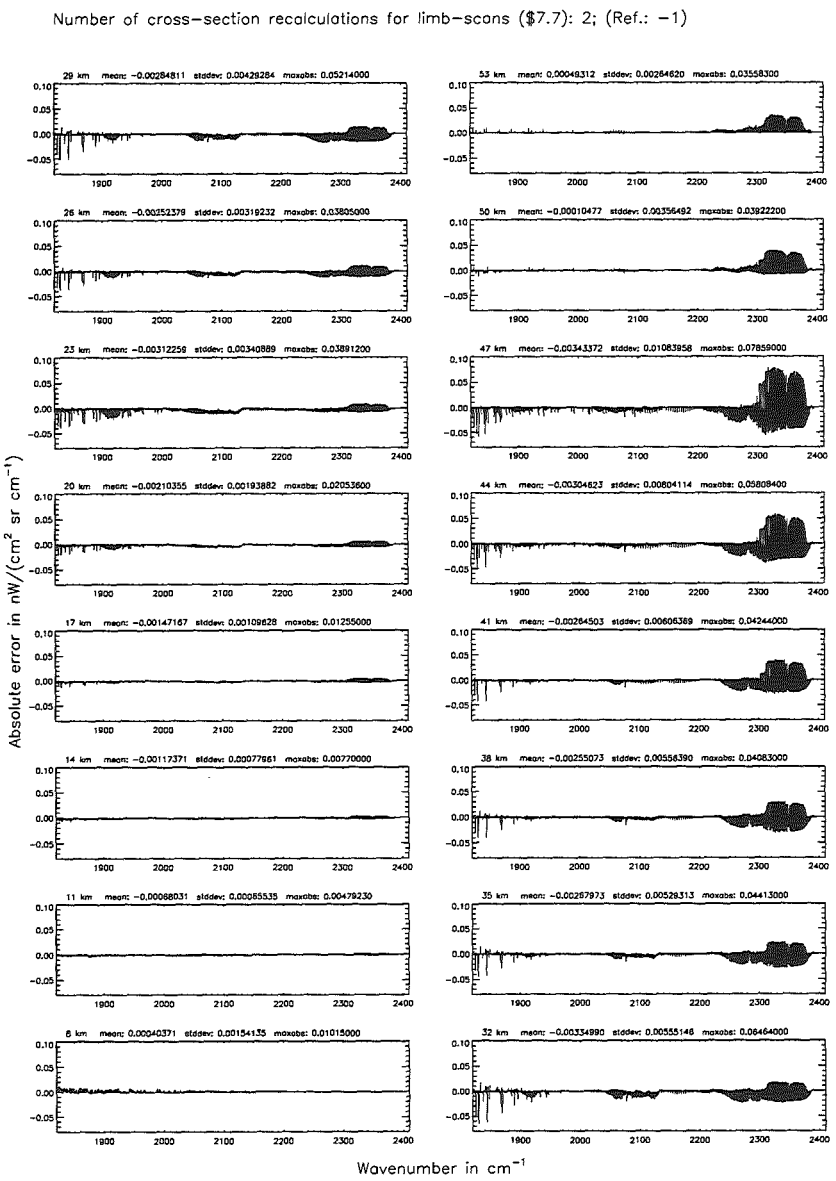


Figure 284: Relative error [%] with respect to MIPAS NESR



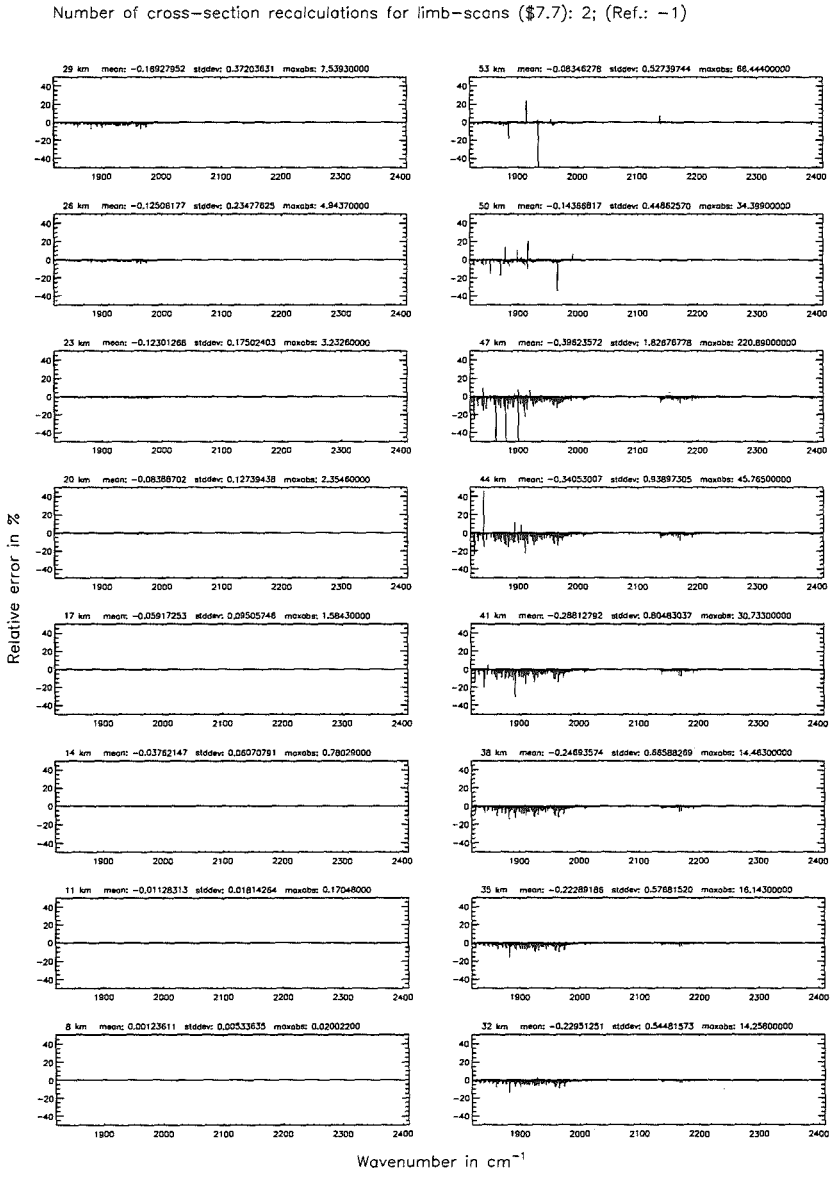


Figure 285: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

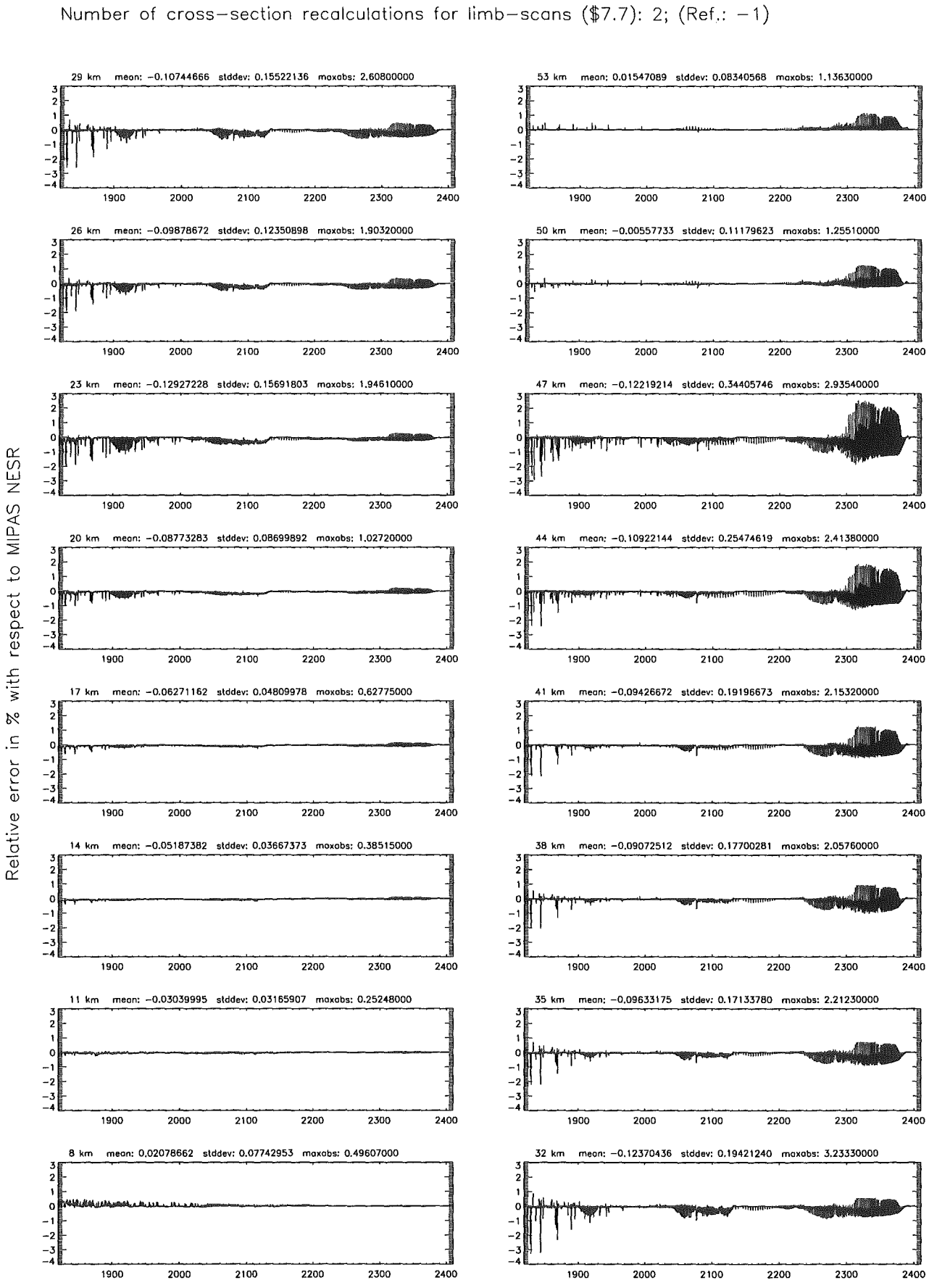
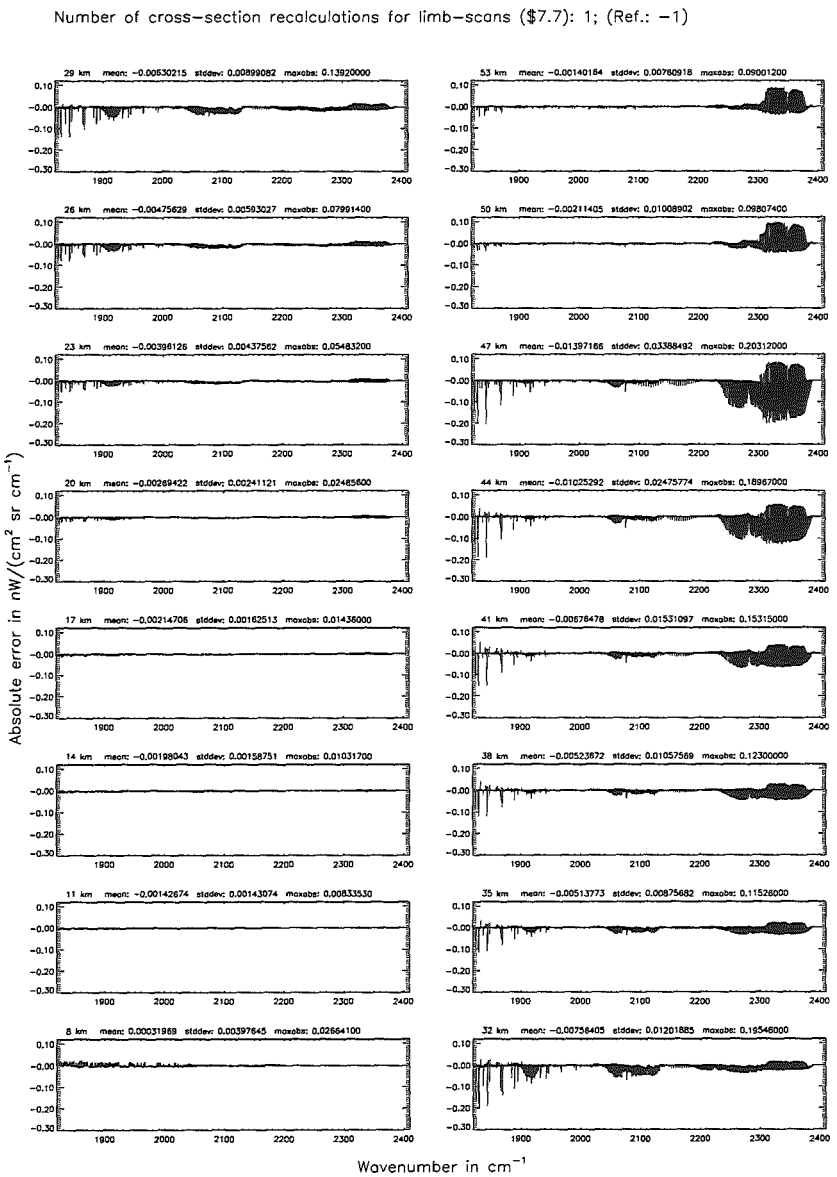


Figure 286: Relative error [%] with respect to MIPAS NESR



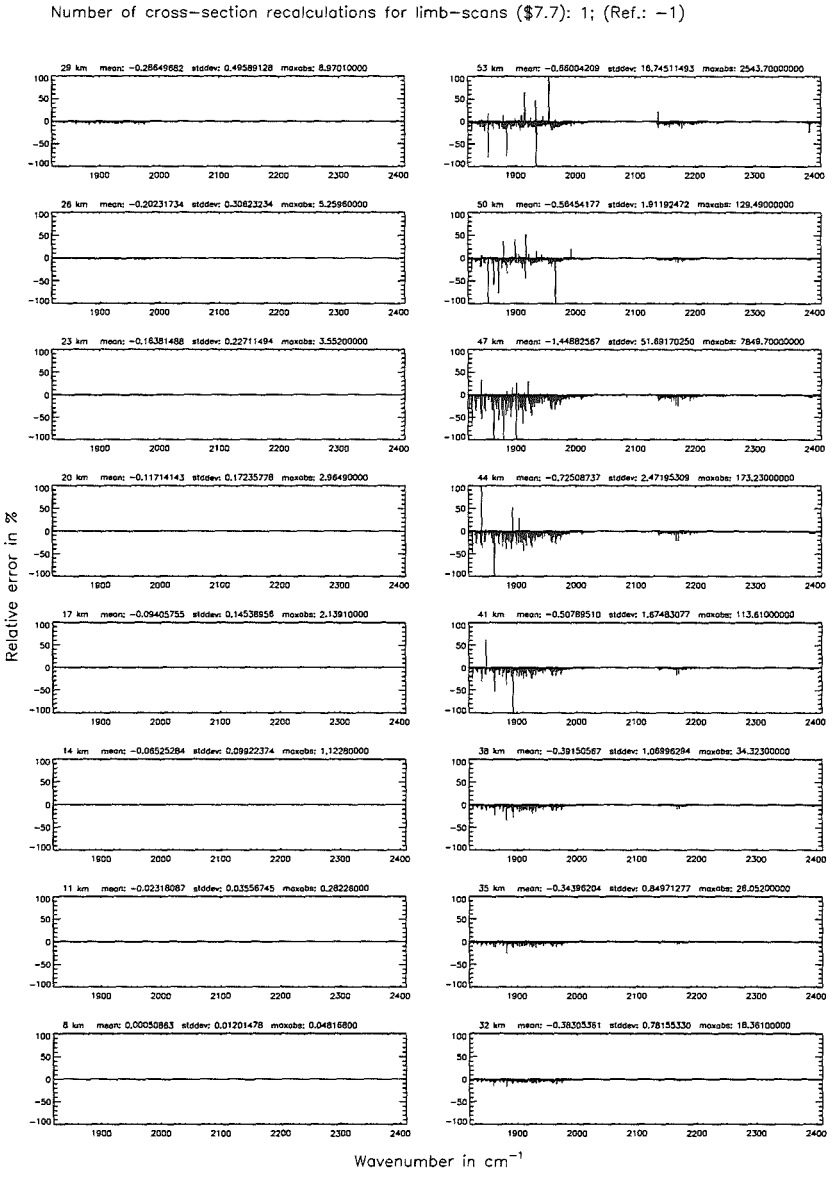


Figure 287: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

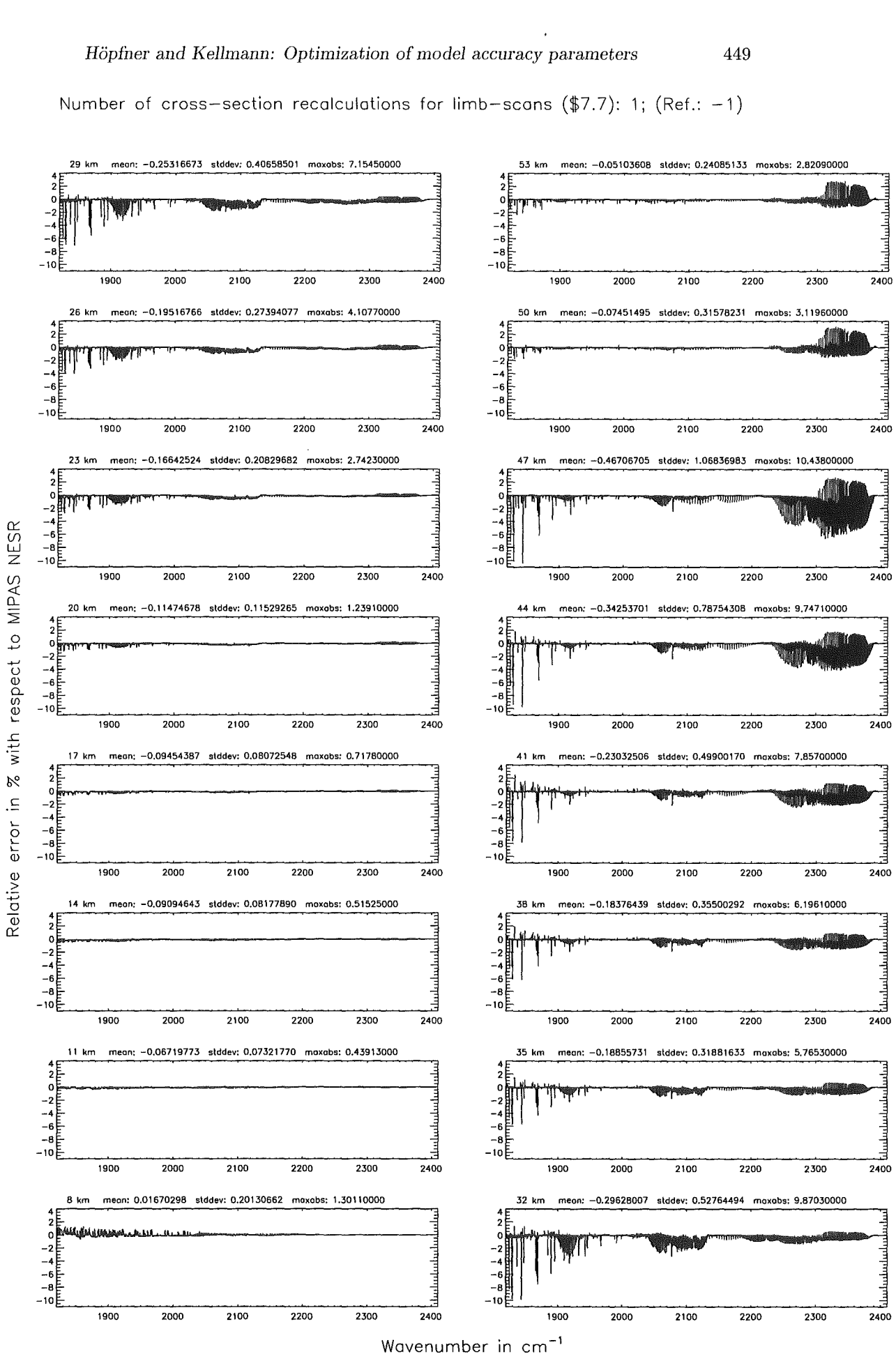


Figure 288: Relative error [%] with respect to MIPAS NESR

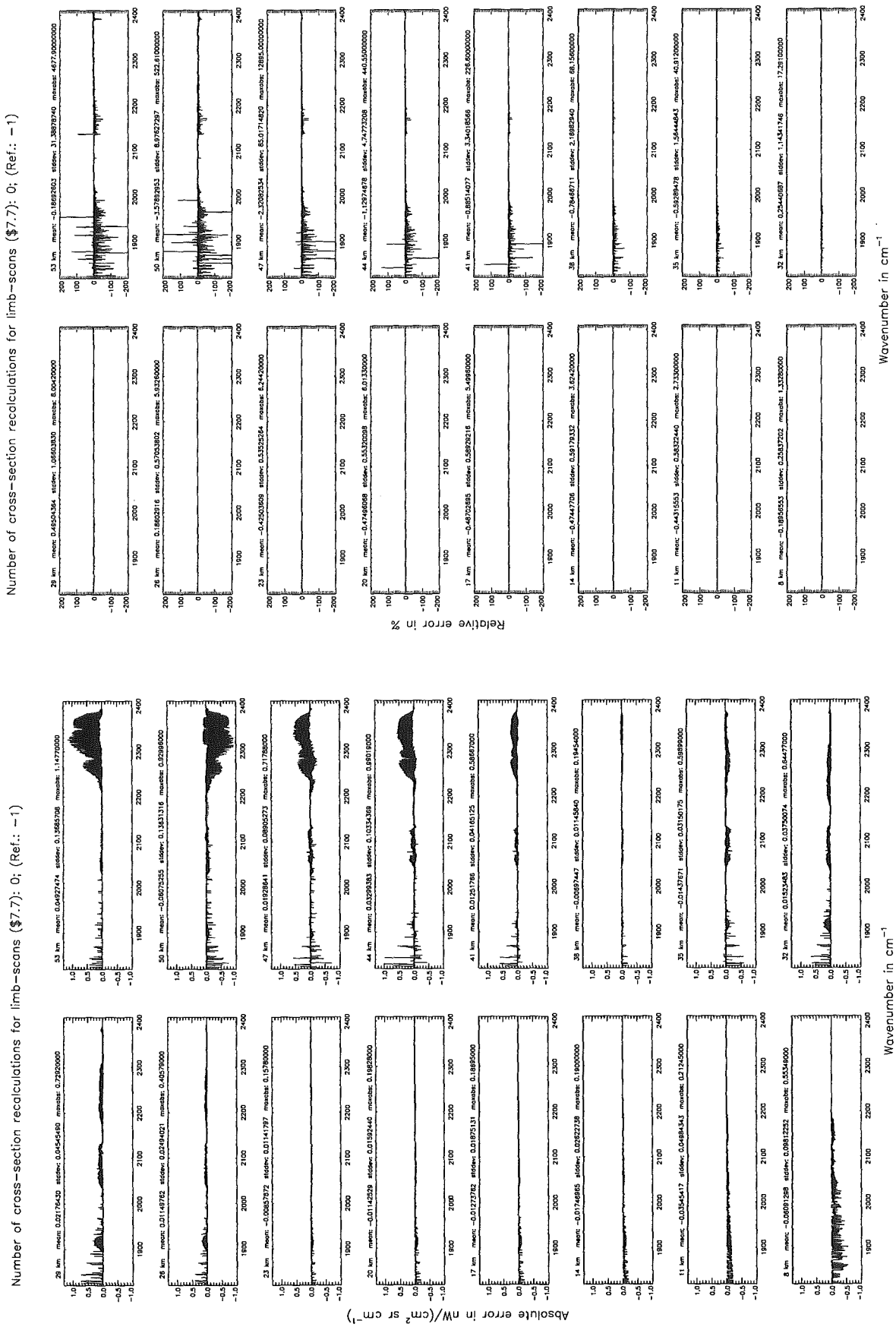


Figure 289: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

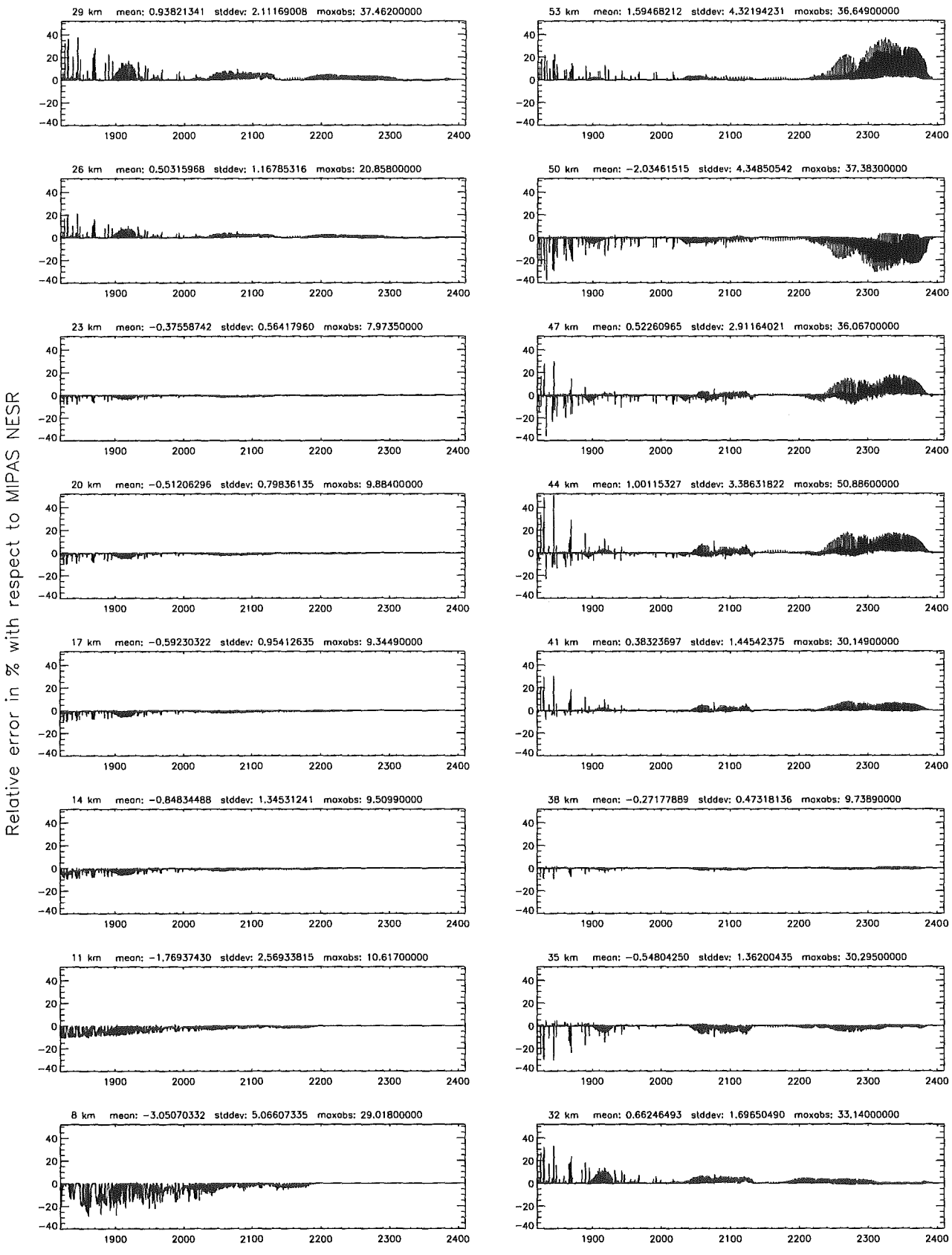
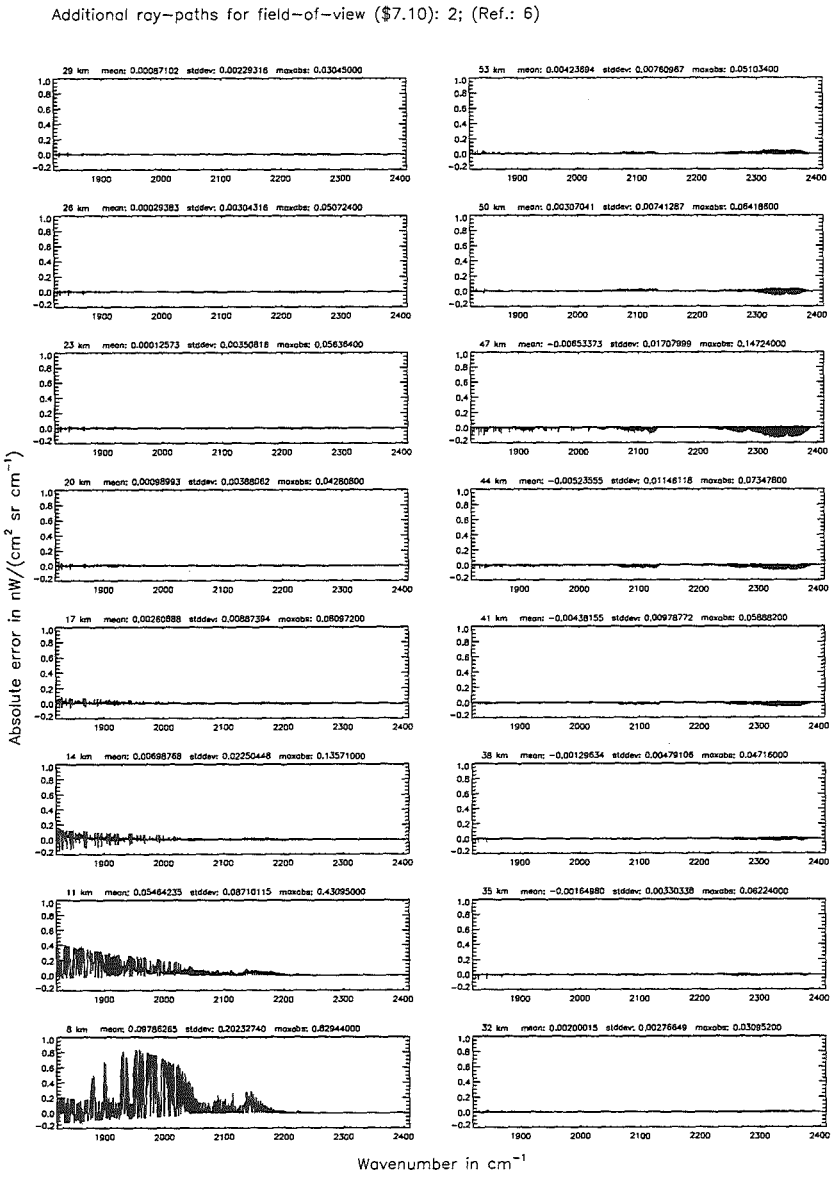


Figure 290: Relative error [%] with respect to MIPAS NESR



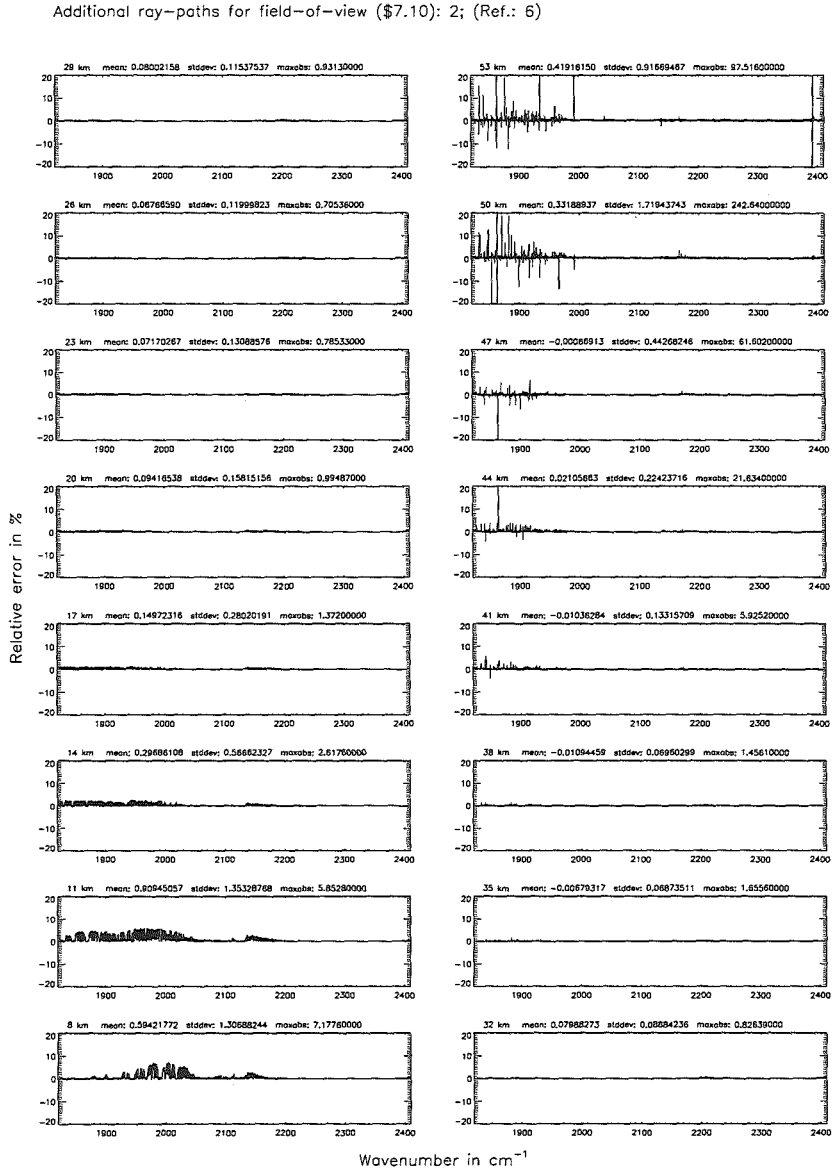


Figure 291: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

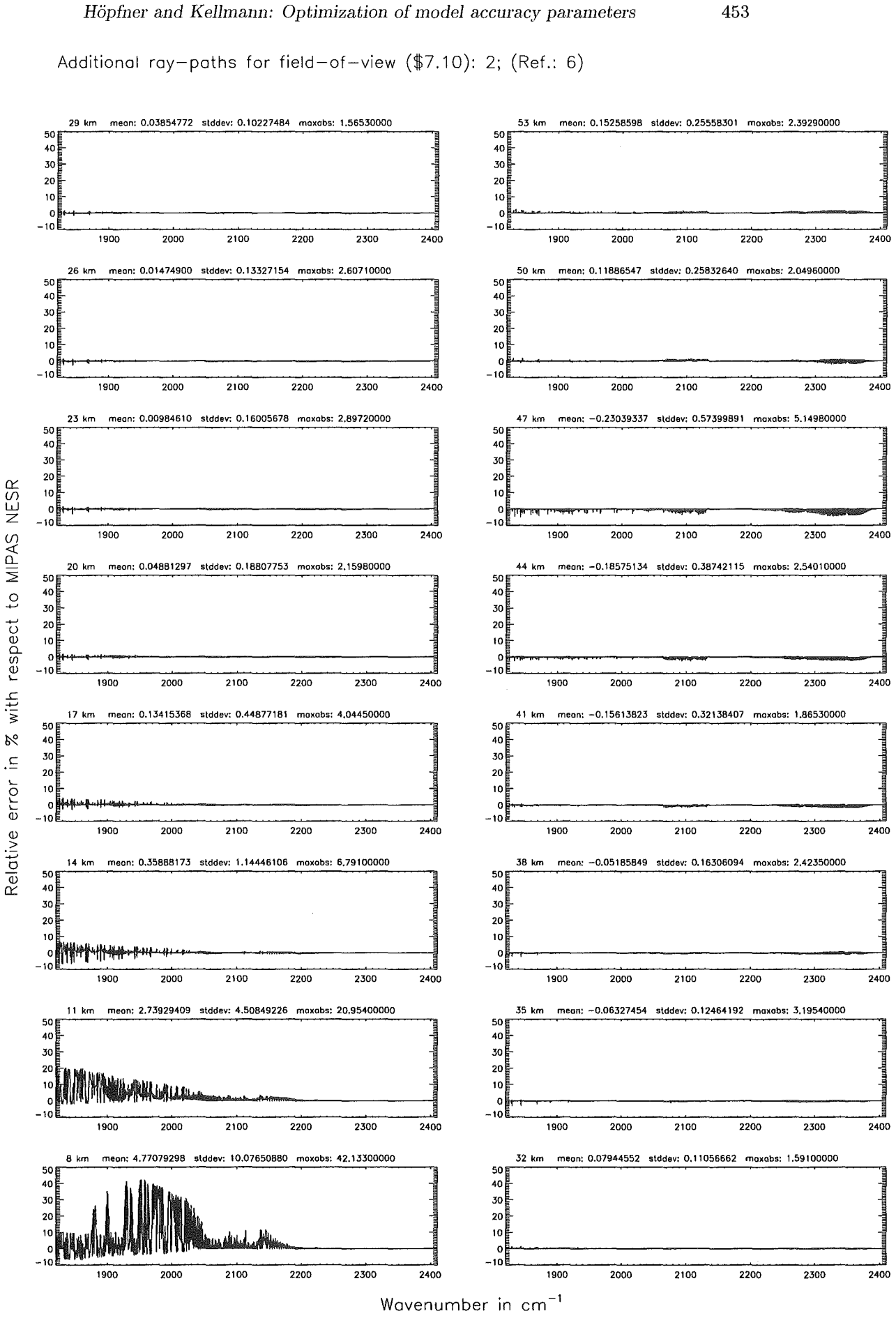
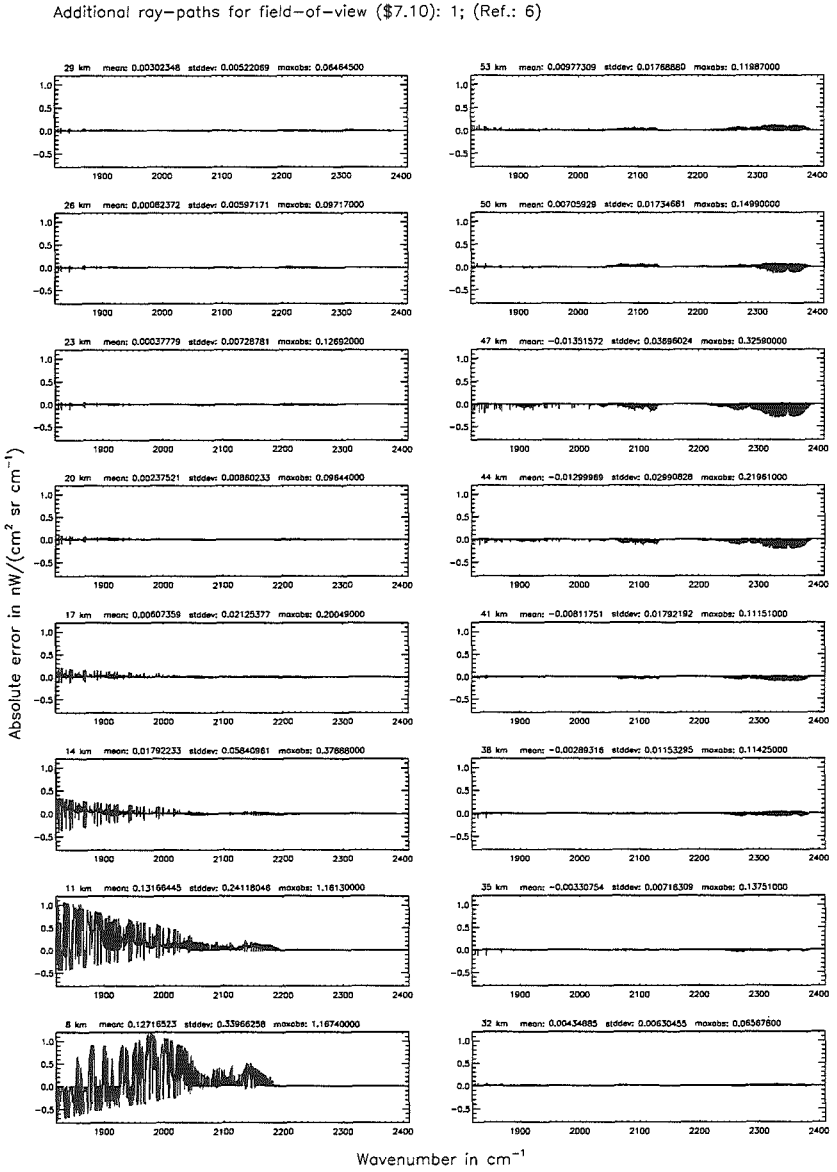


Figure 292: Relative error [%] with respect to MIPAS NESR



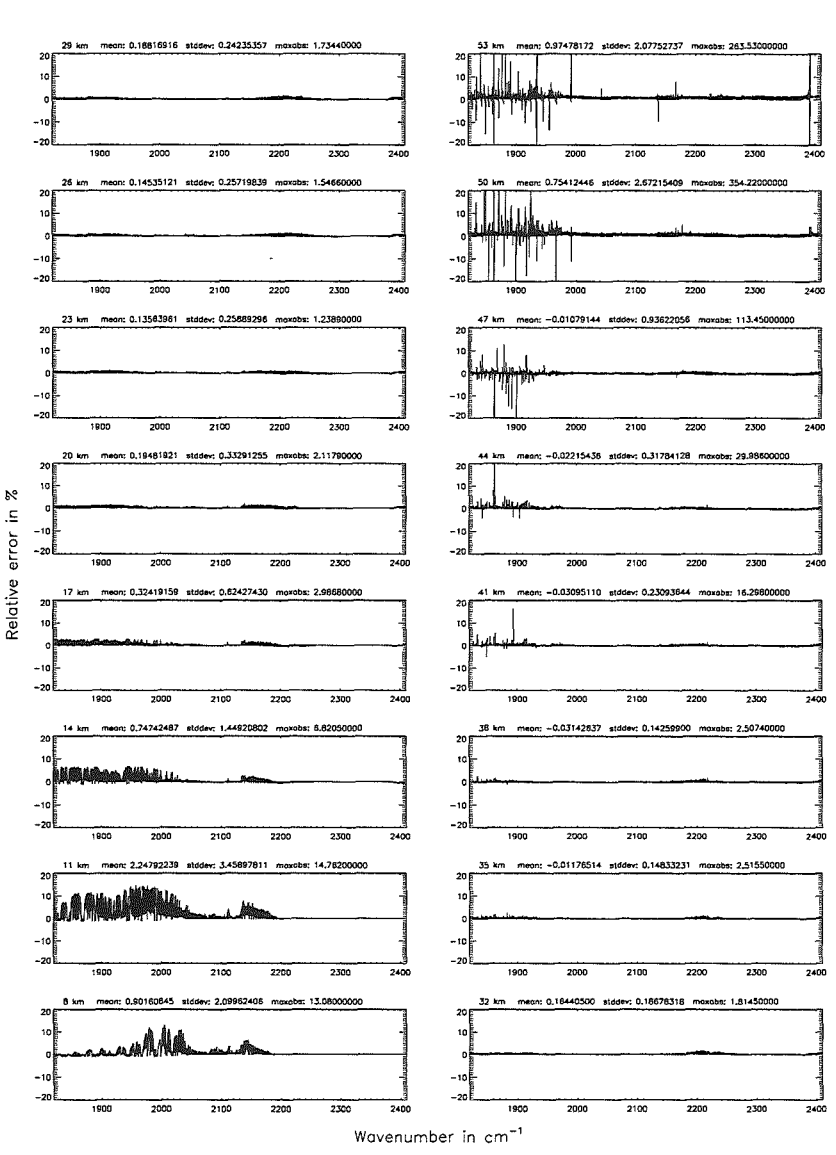


Figure 293: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

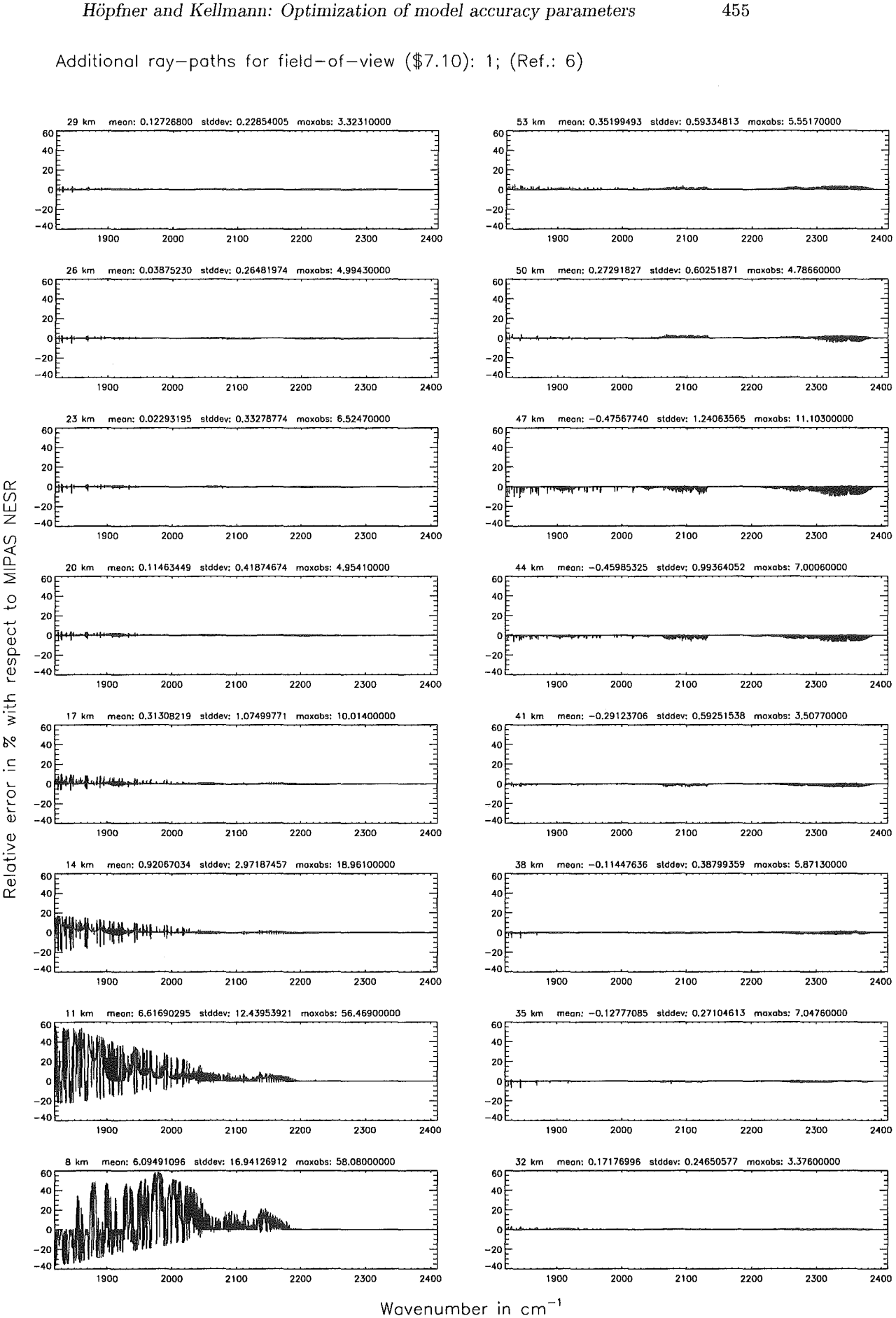


Figure 294: Relative error [%] with respect to MIPAS NESR

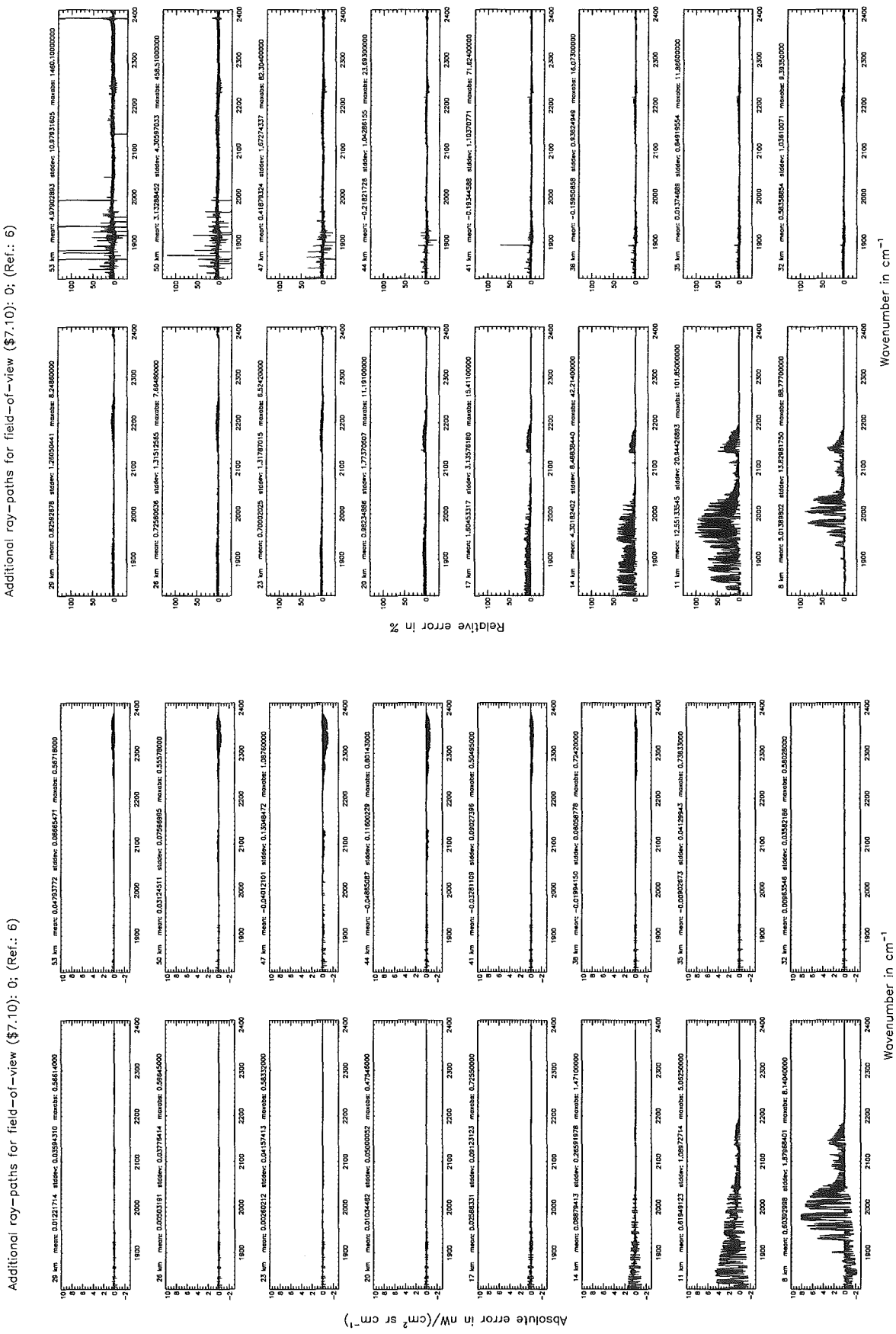


Figure 295: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

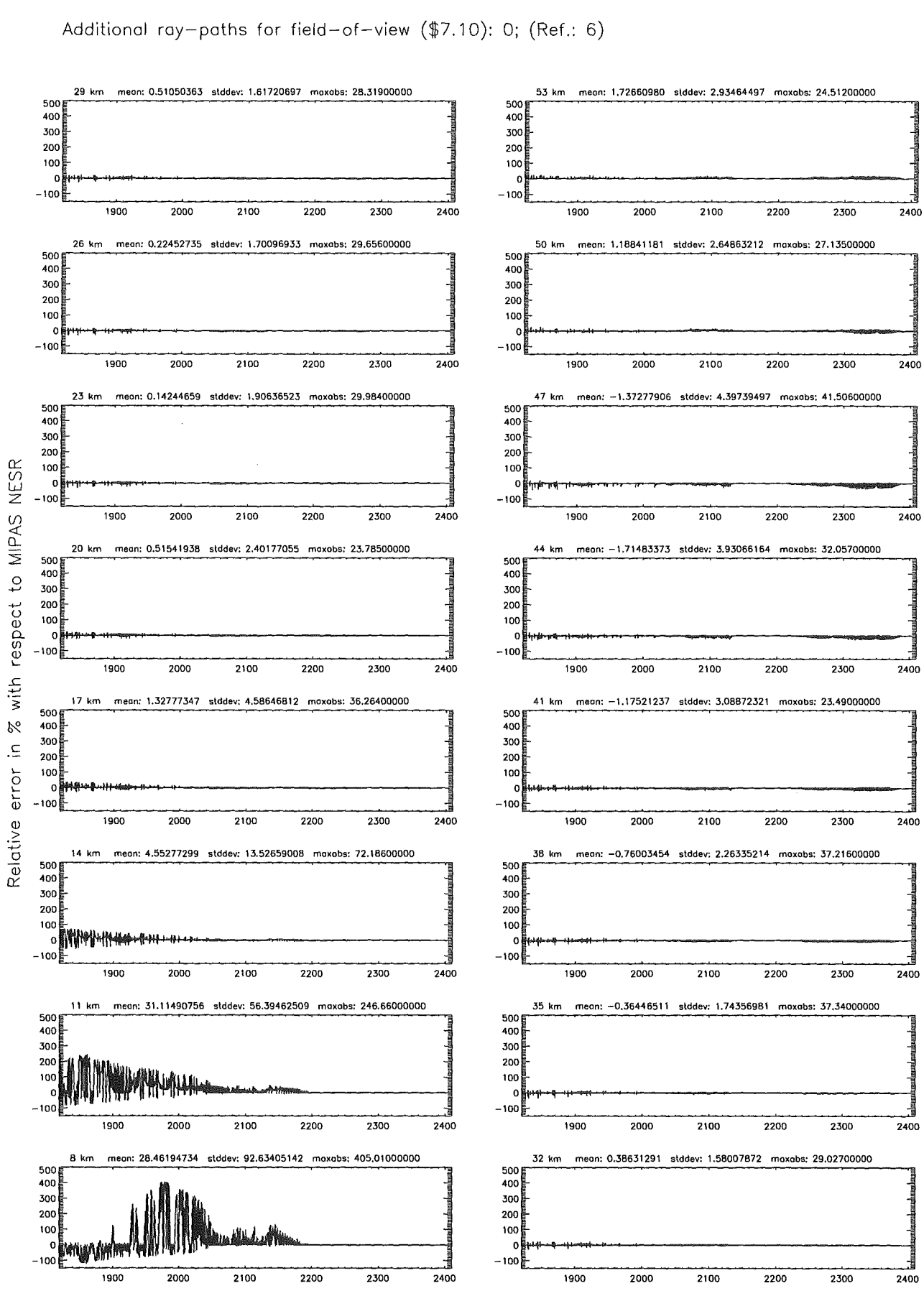


Figure 296: Relative error [%] with respect to MIPAS NESR

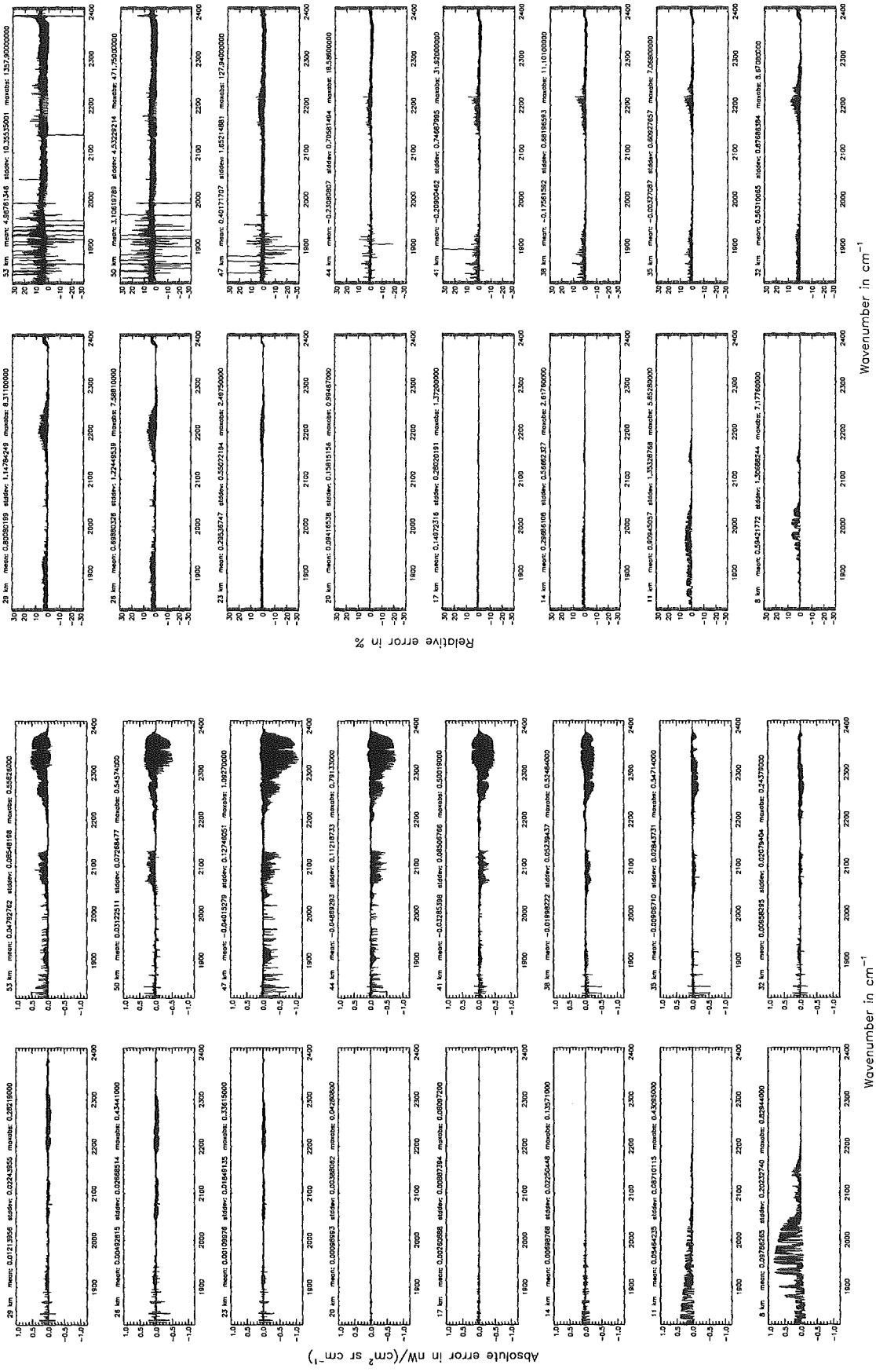


Figure 297: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

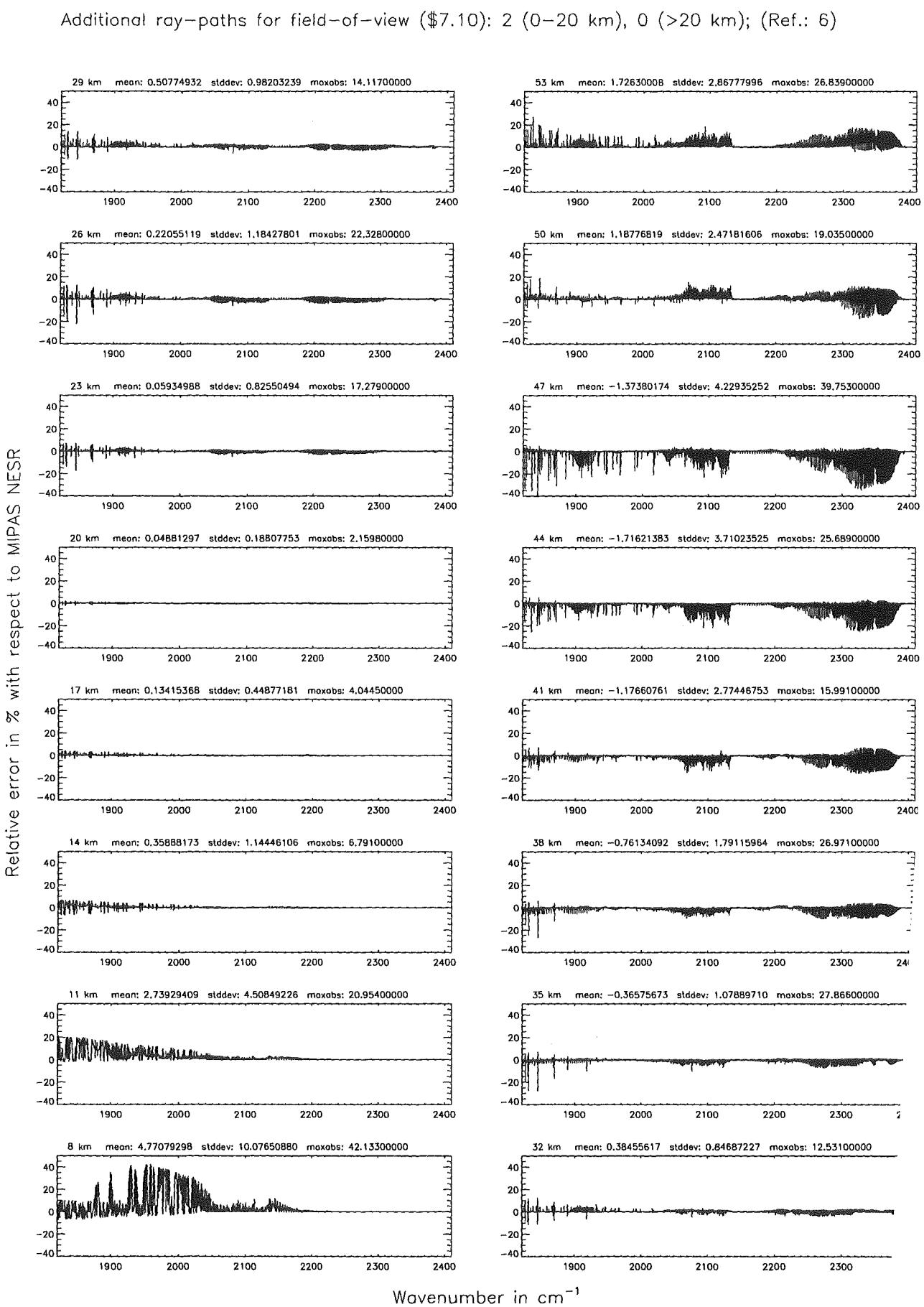


Figure 298: Relative error [%] with respect to MIPAS NESR

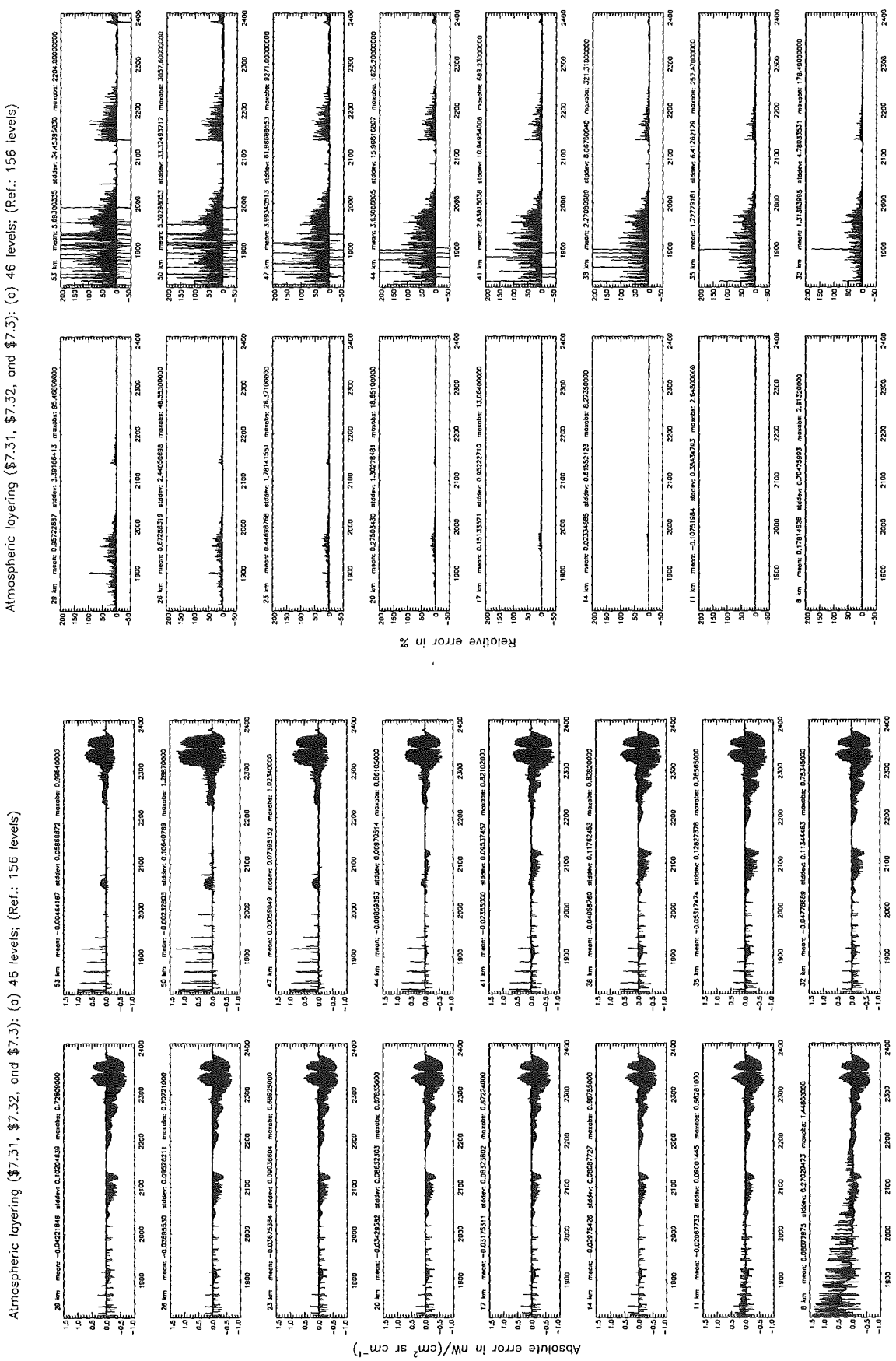


Figure 299: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

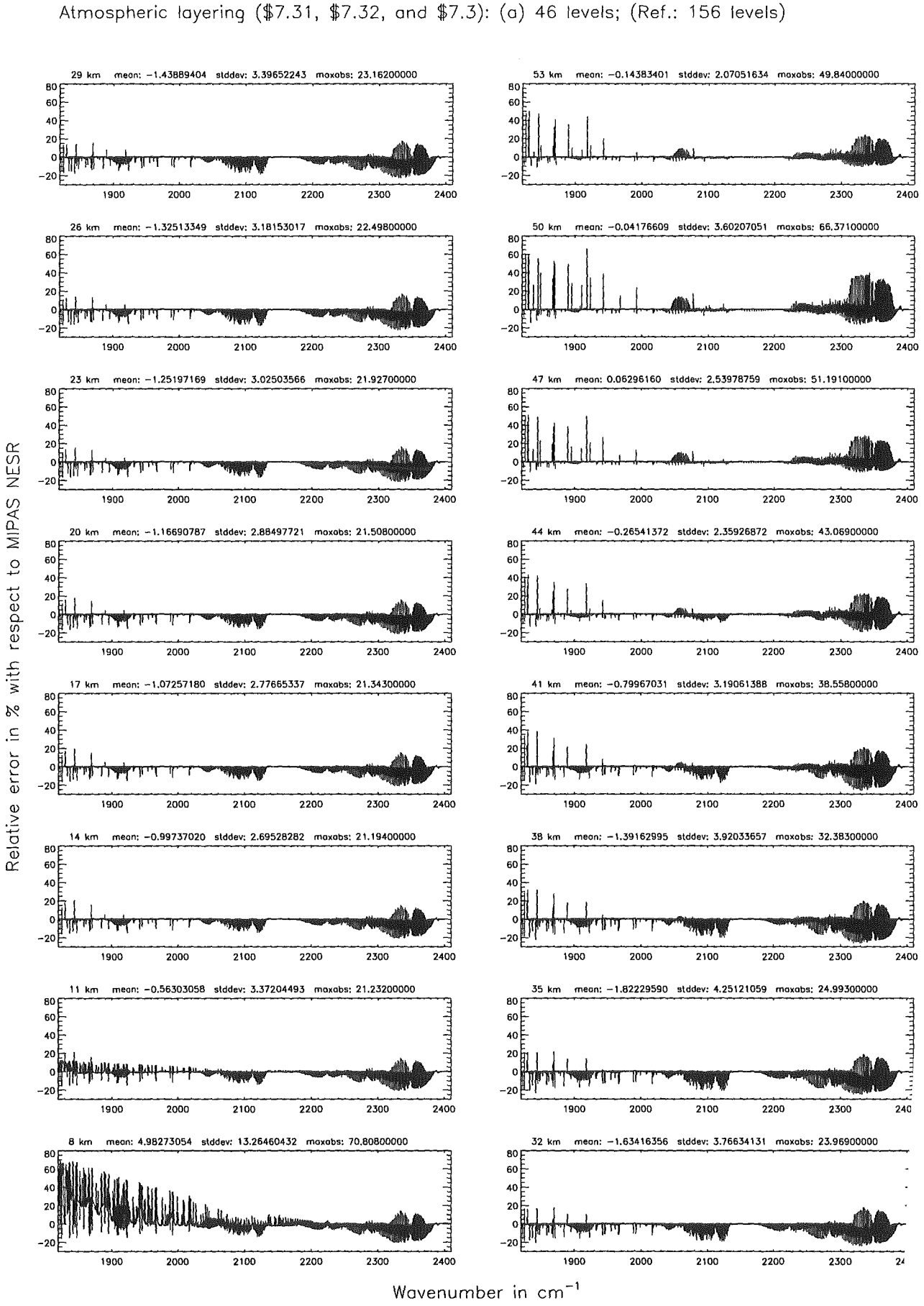
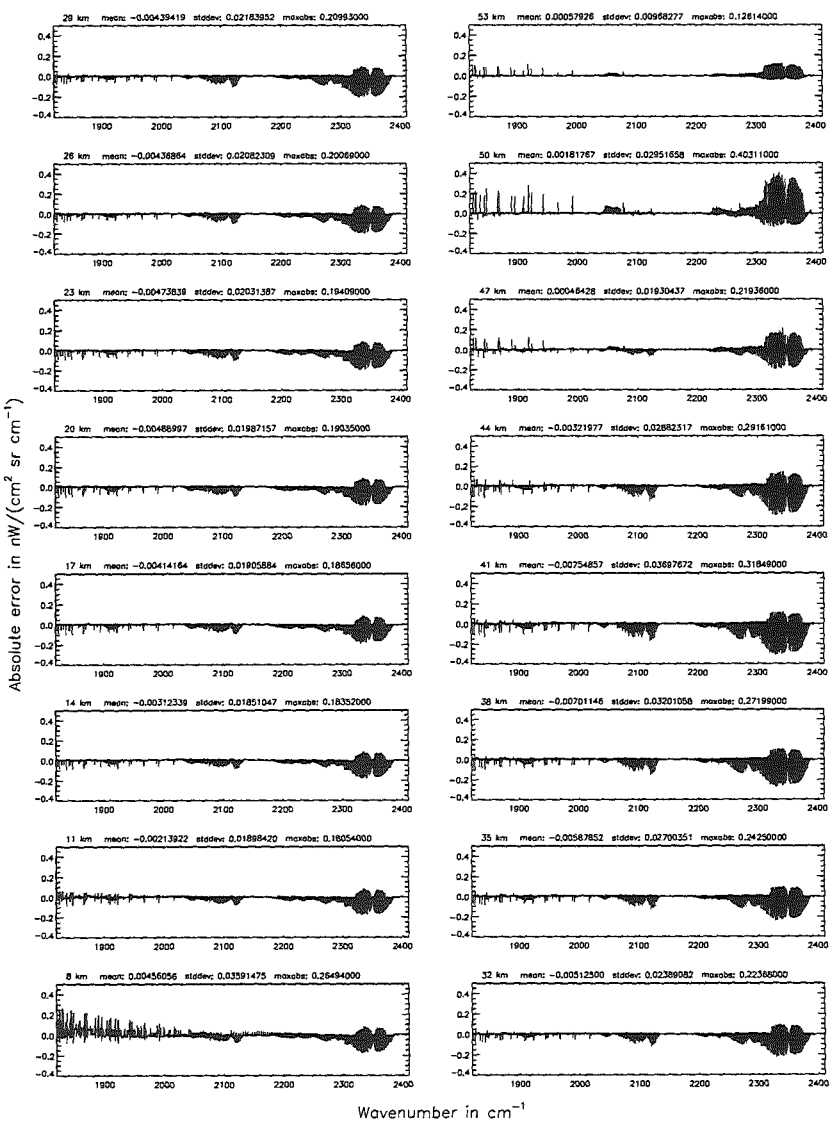


Figure 300: Relative error [%] with respect to MIPAS NESR



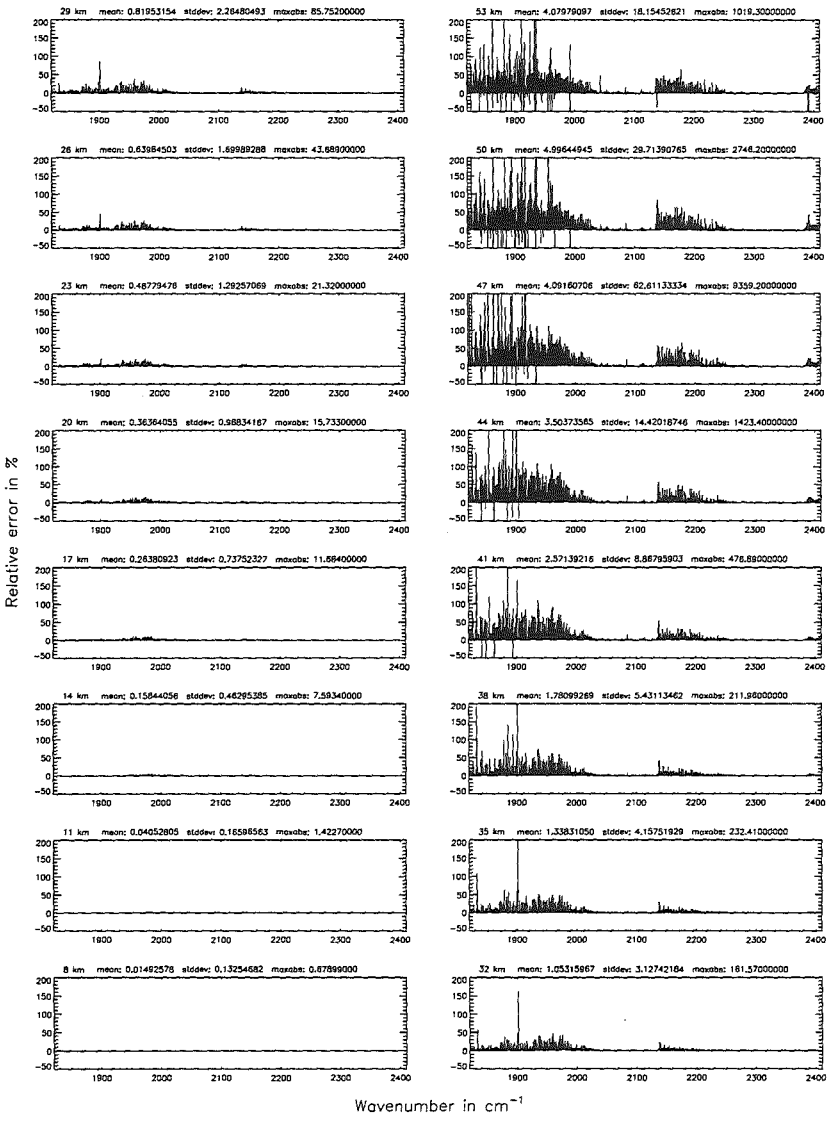


Figure 301: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

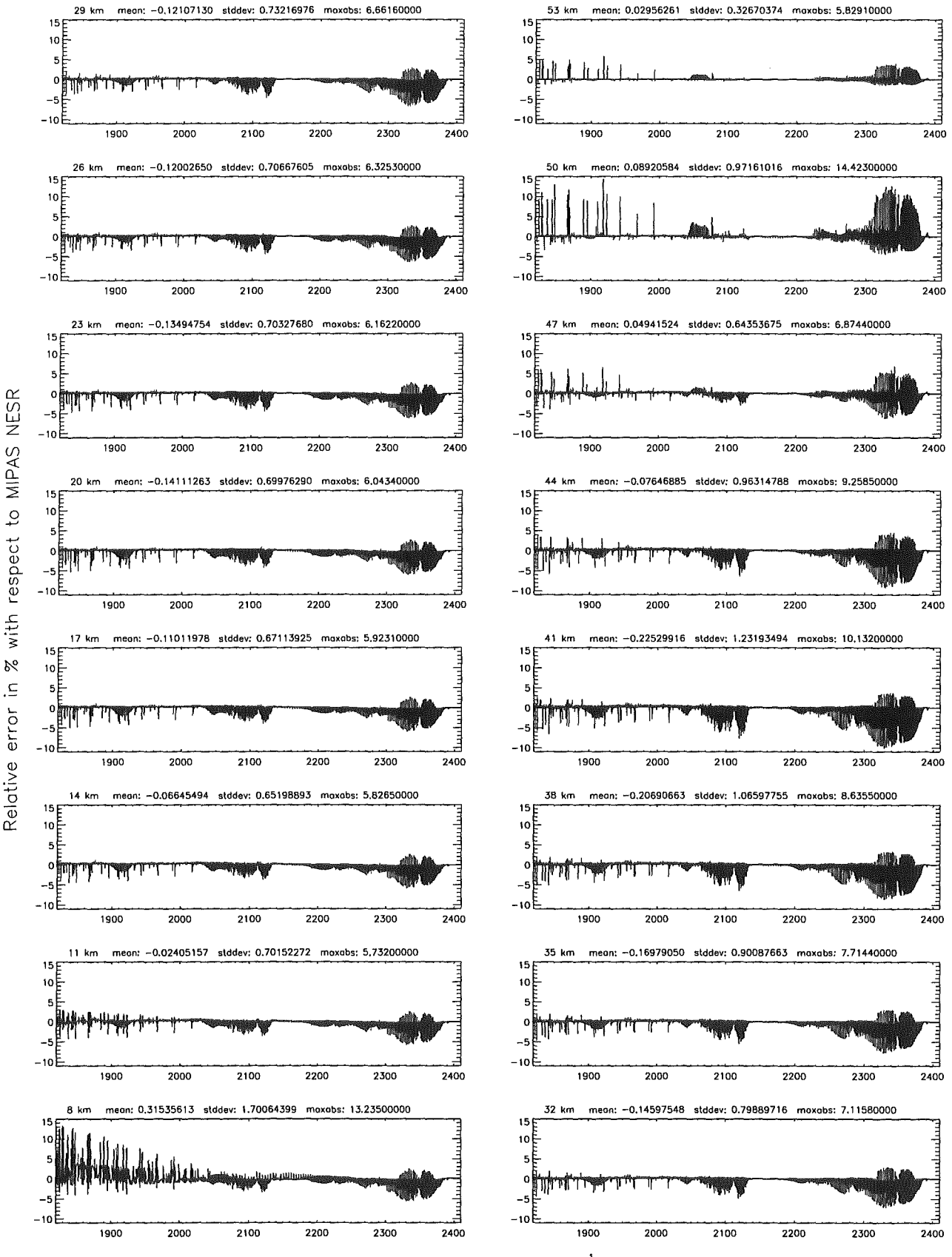
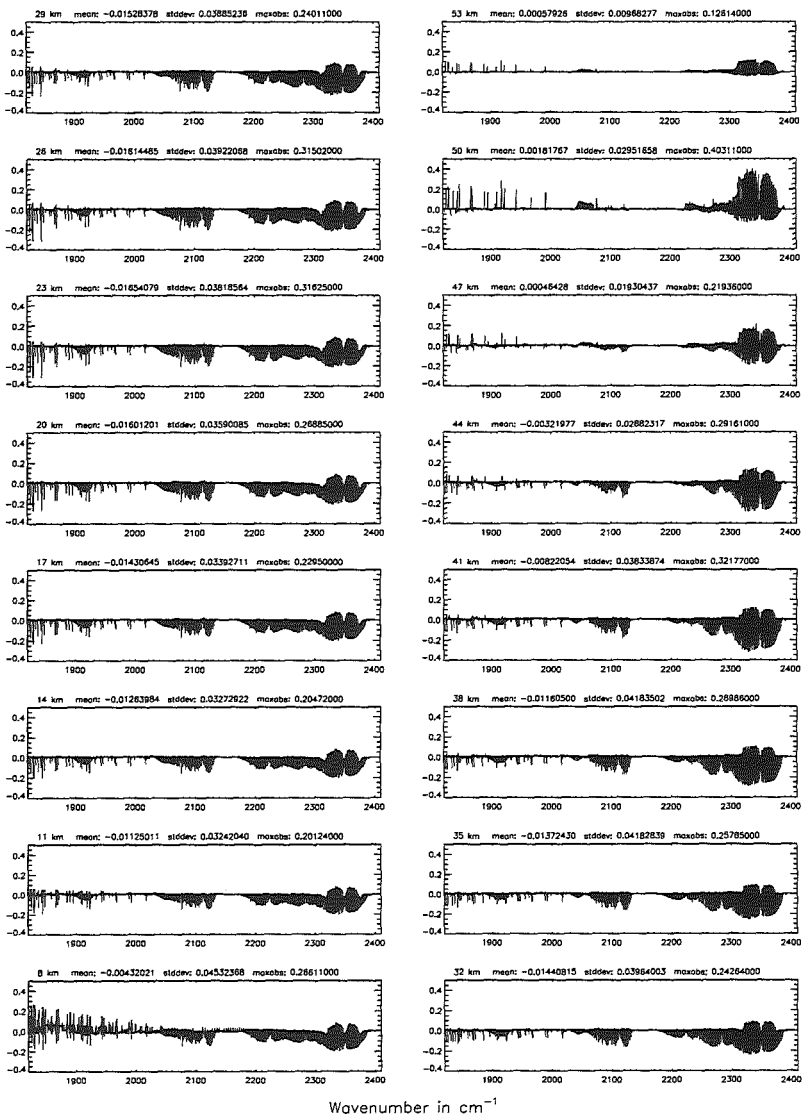


Figure 302: Relative error [%] with respect to MIPAS NESR



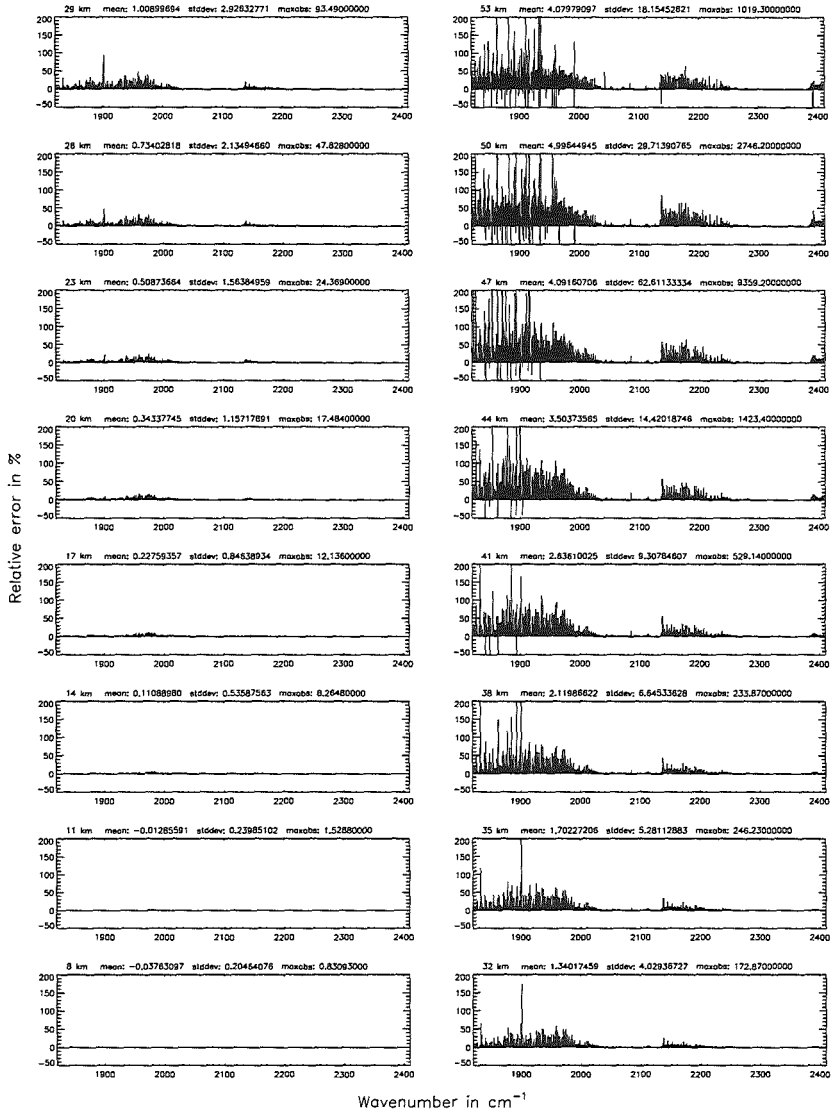


Figure 303: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (c) 66 levels; (Ref.: 156 levels)

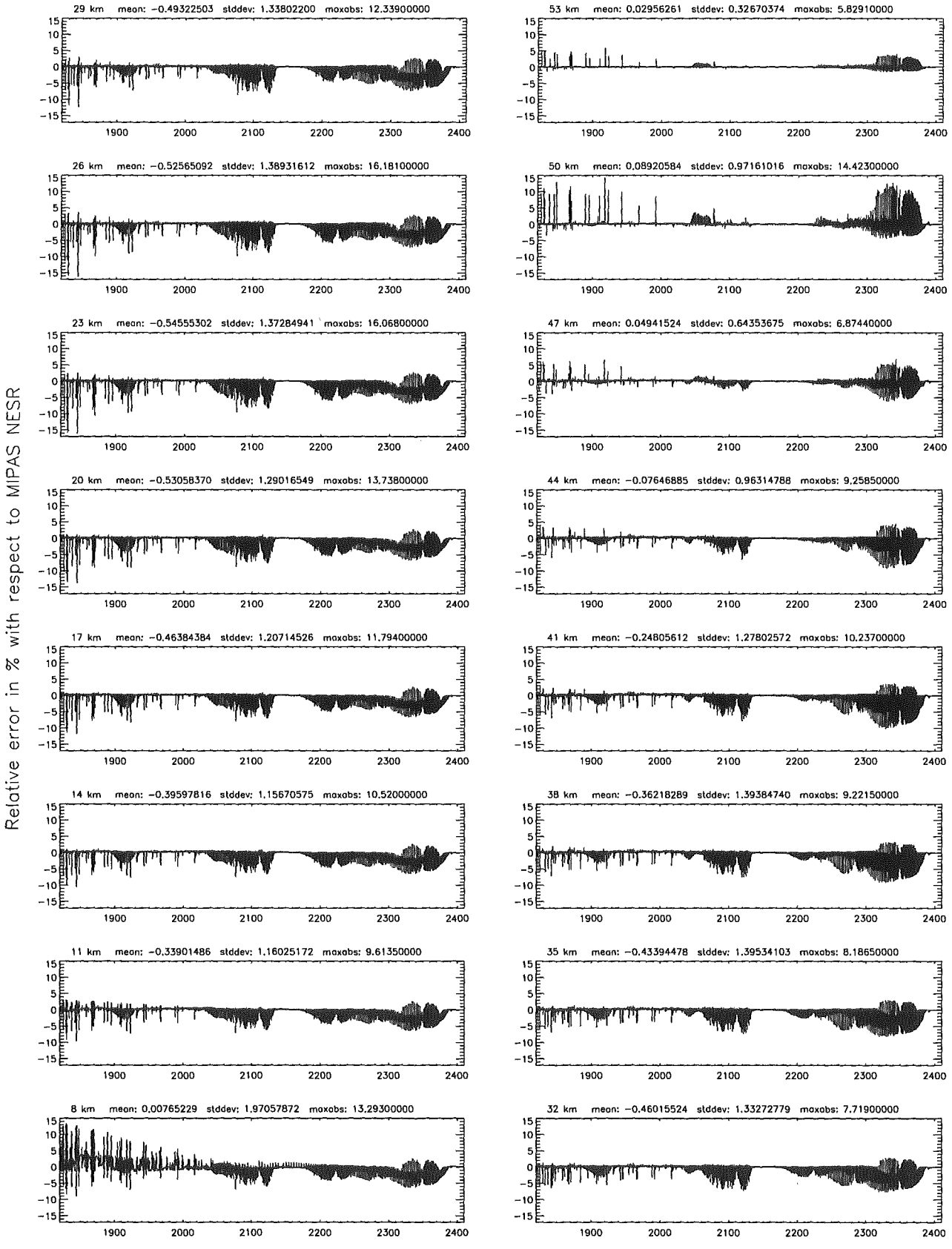
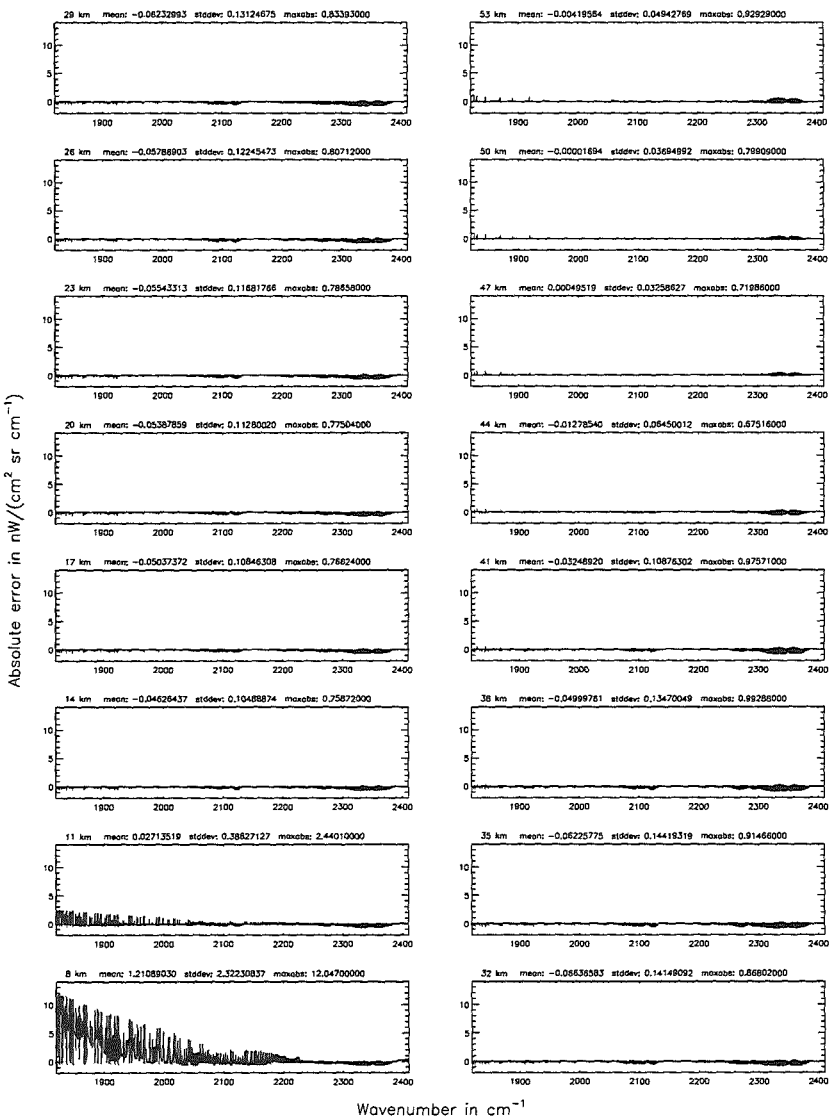


Figure 304: Relative error [%] with respect to MIPAS NESR



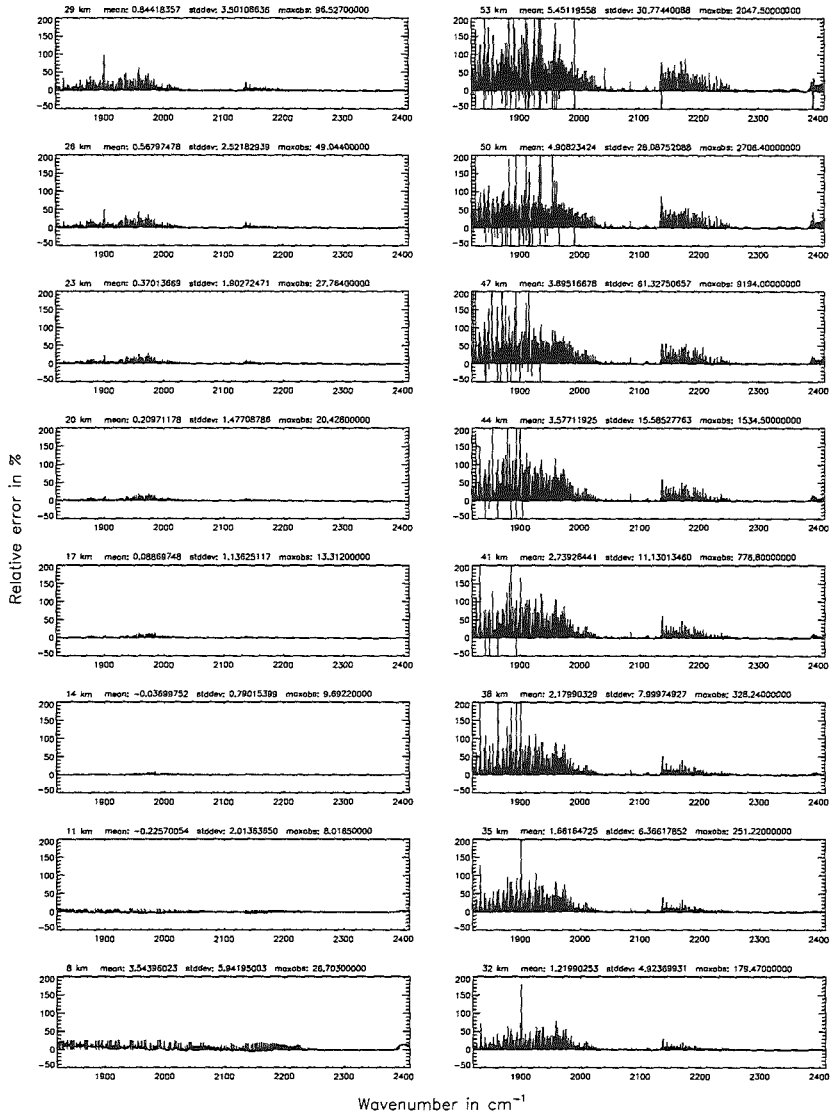


Figure 305: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

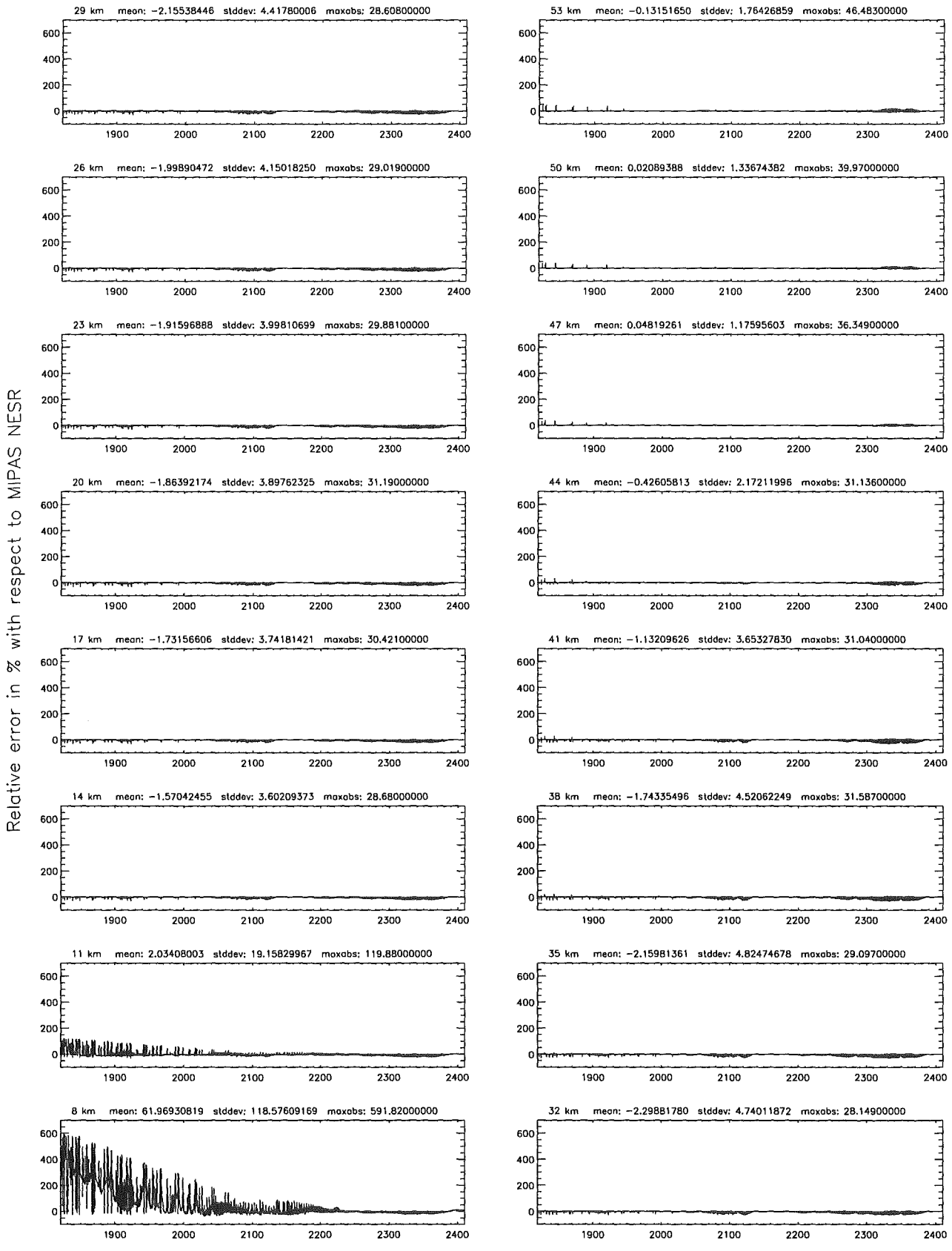


Figure 306: Relative error [%] with respect to MIPAS NESR

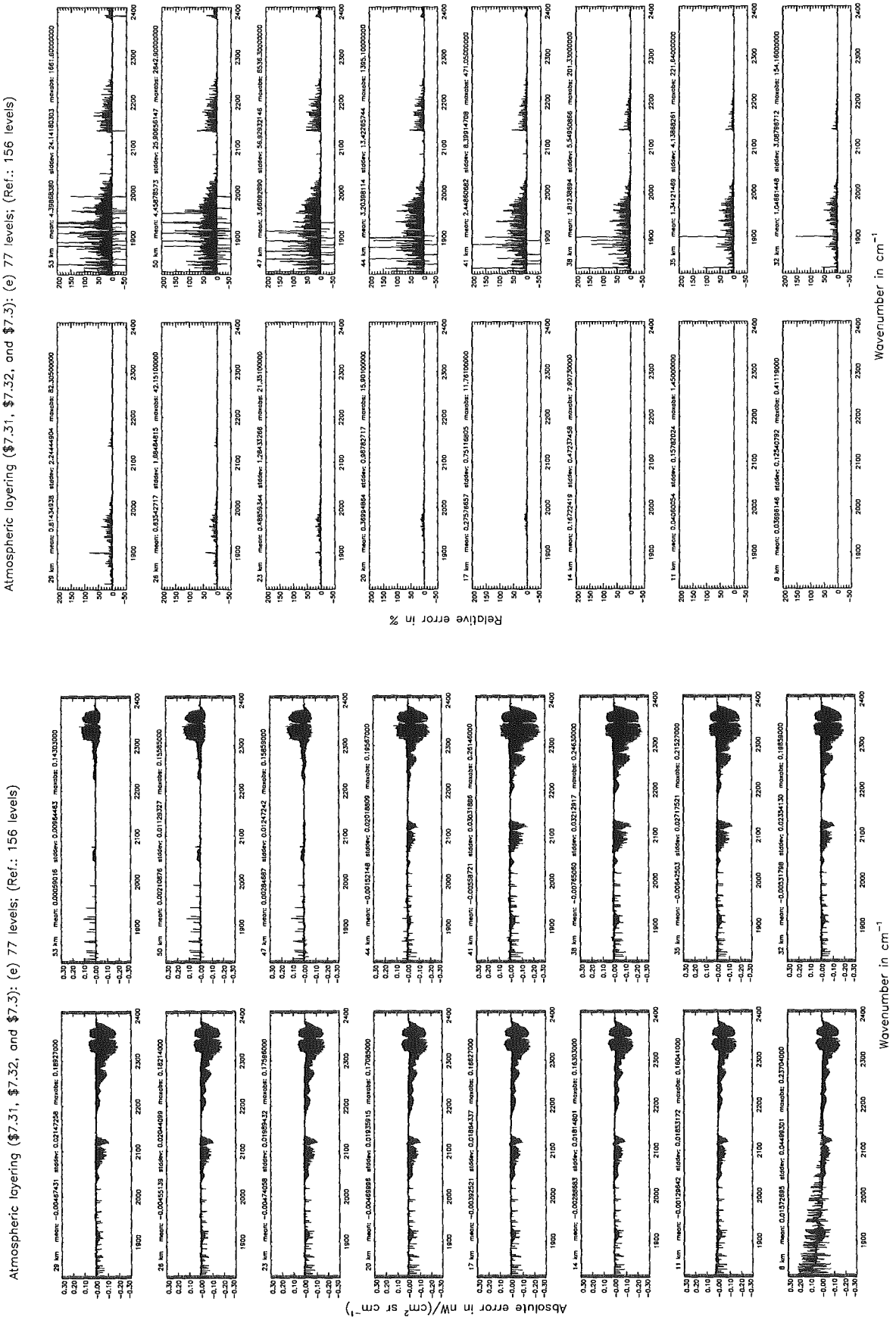


Figure 307: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

Atmospheric layering ($7.31, $7.32, and $7.3): (e) 77 levels; (Ref.: 156 levels)

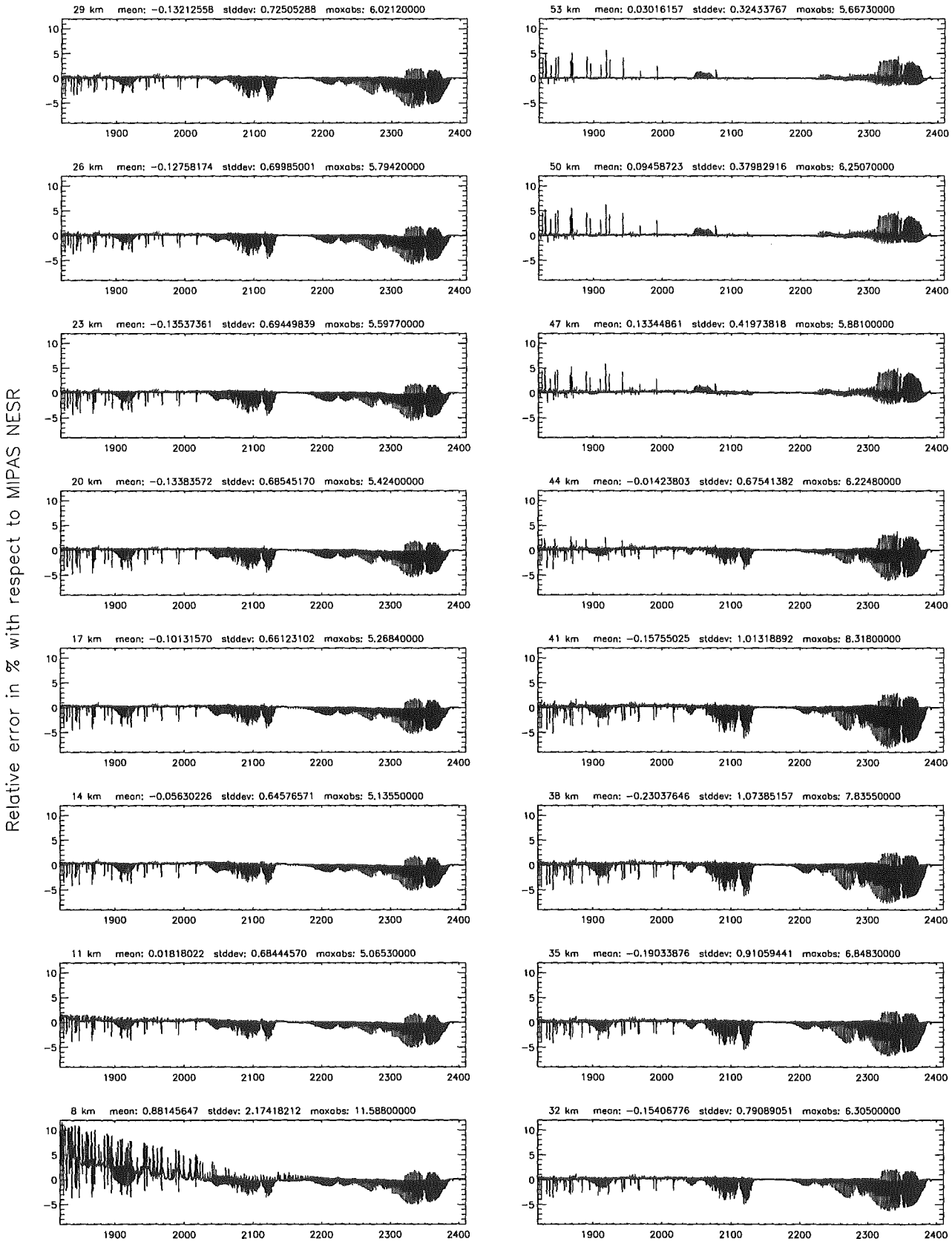


Figure 308: Relative error [%] with respect to MIPAS NESR

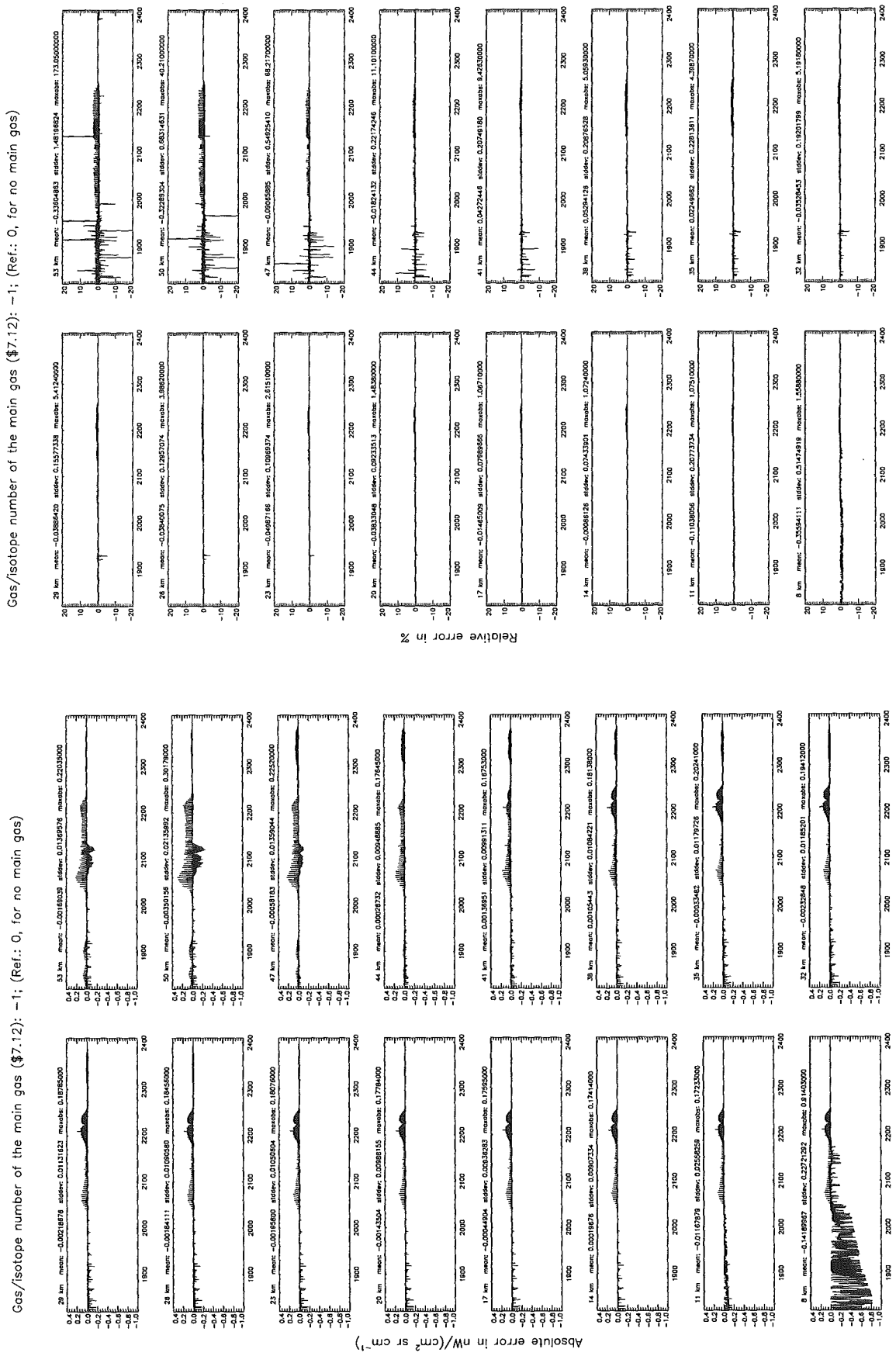


Figure 309: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

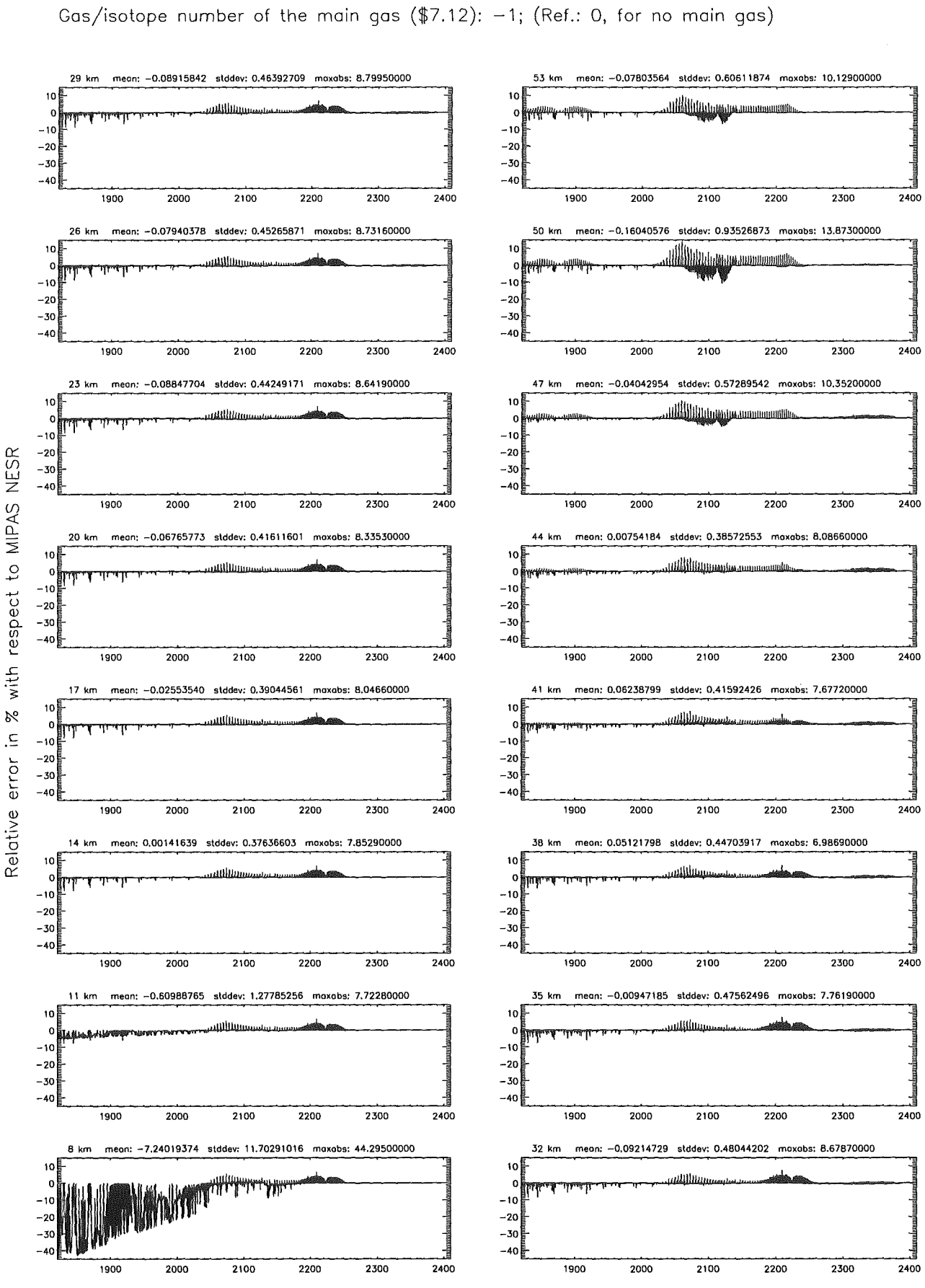
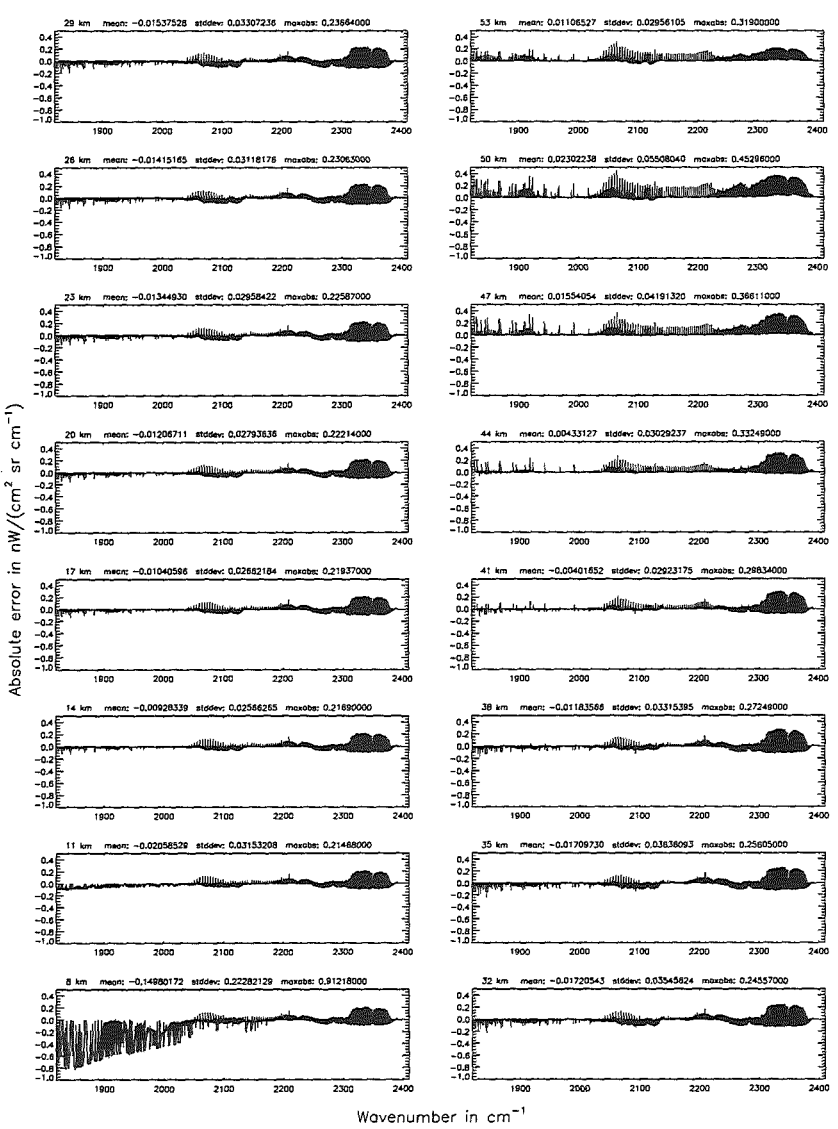


Figure 310: Relative error [%] with respect to MIPAS NESR



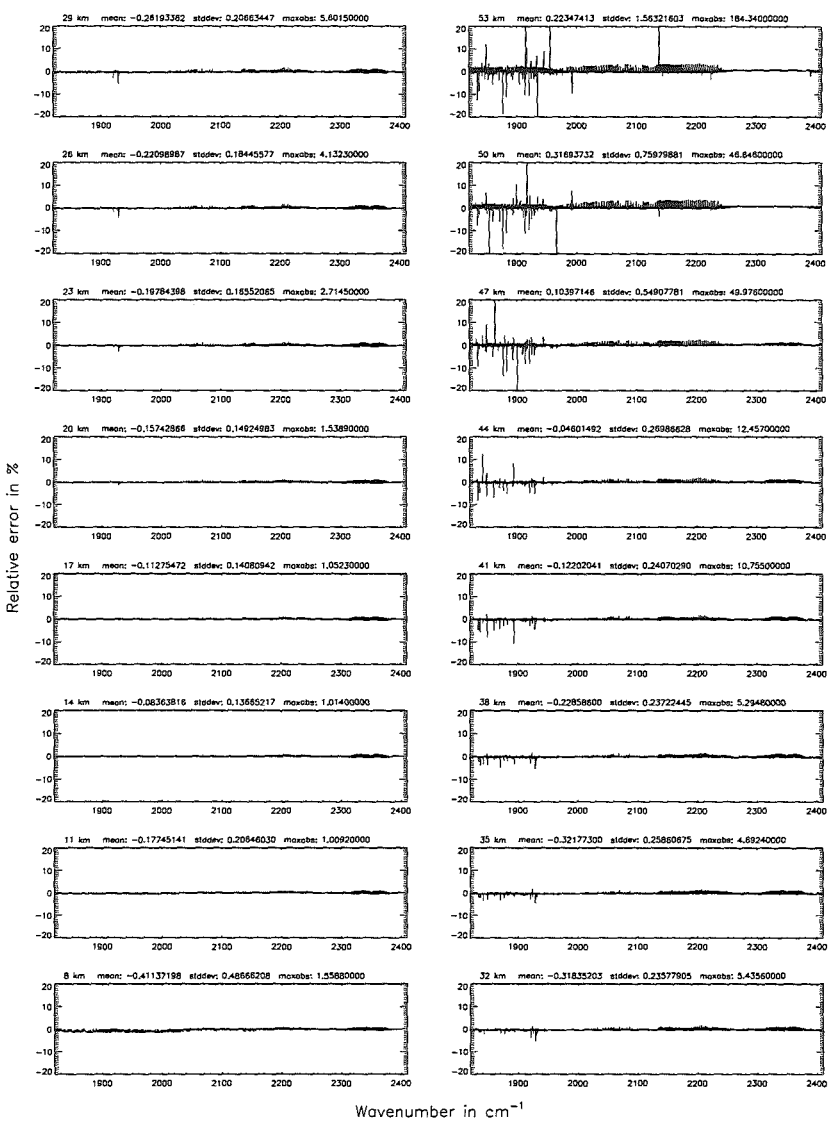


Figure 311: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

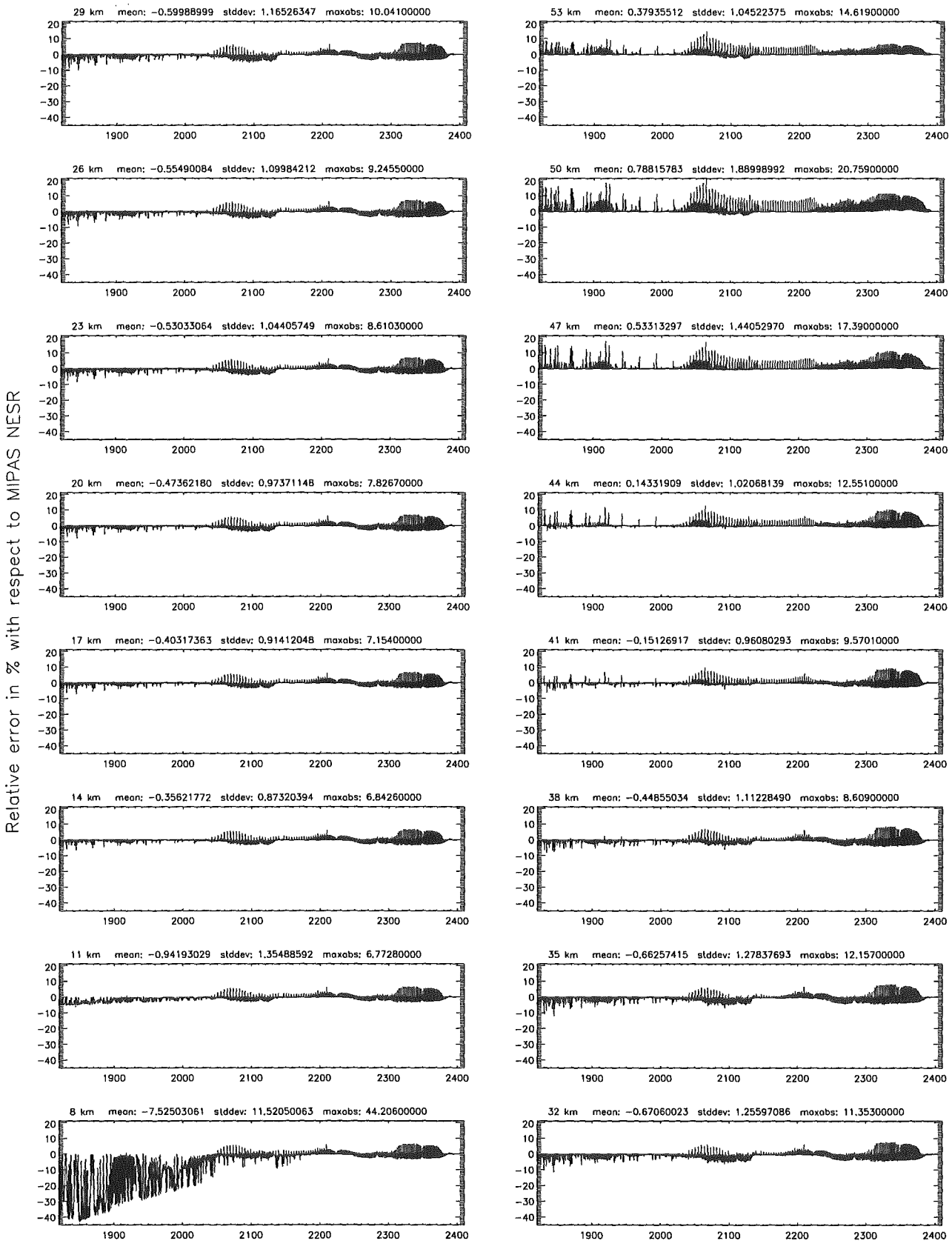


Figure 312: Relative error [%] with respect to MIPAS NESR

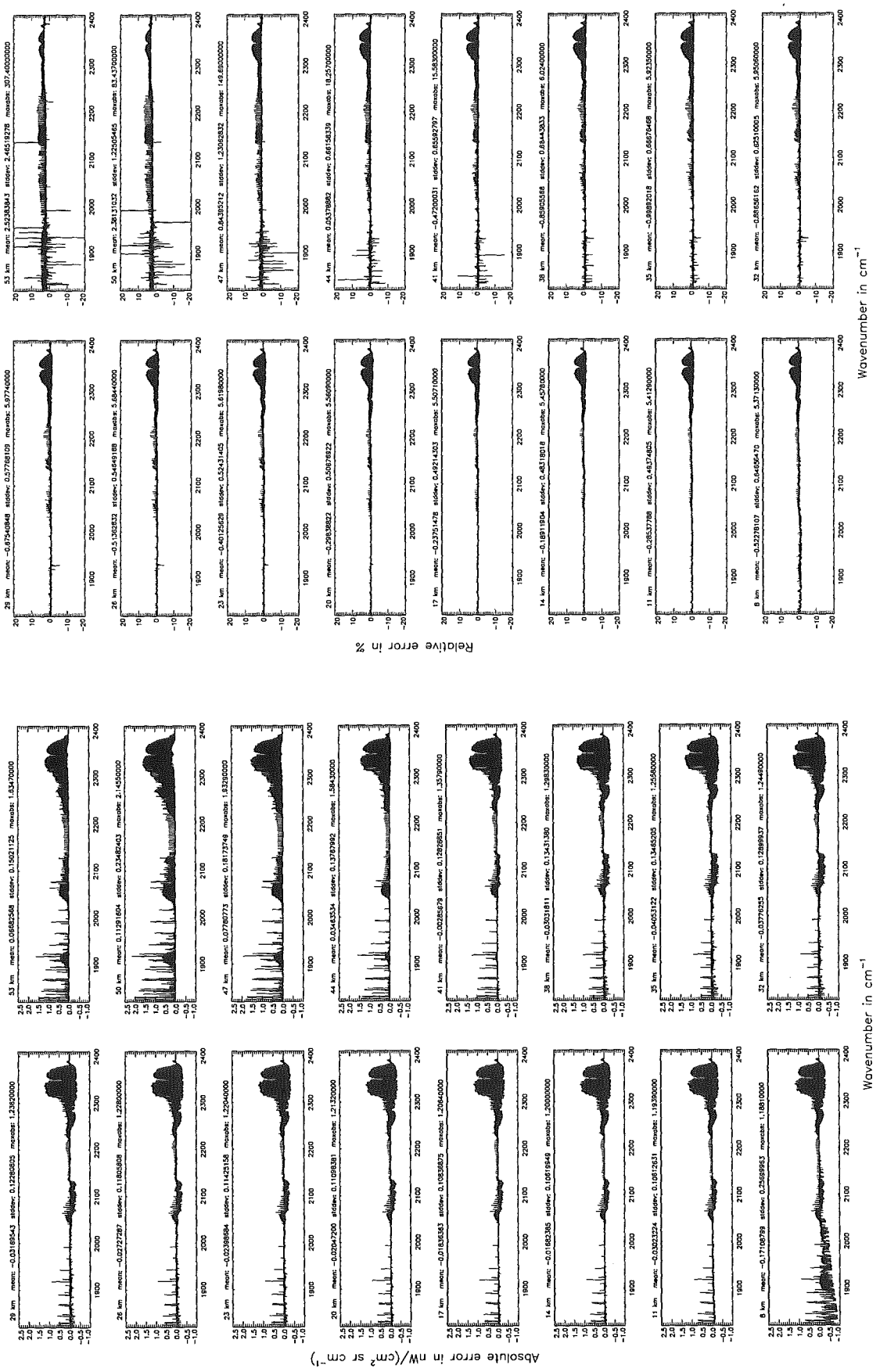


Figure 313: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

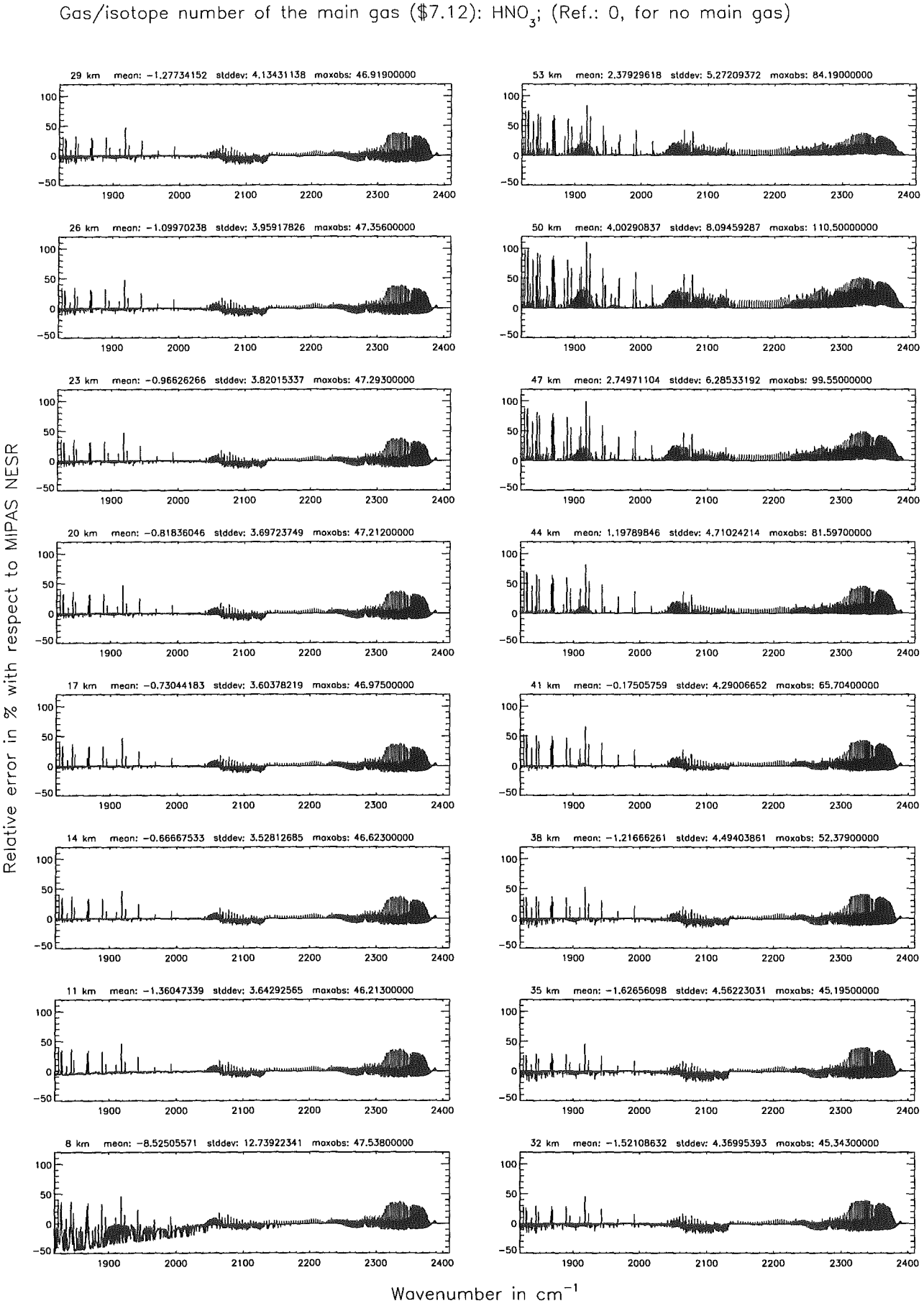


Figure 314: Relative error [%] with respect to MIPAS NESR

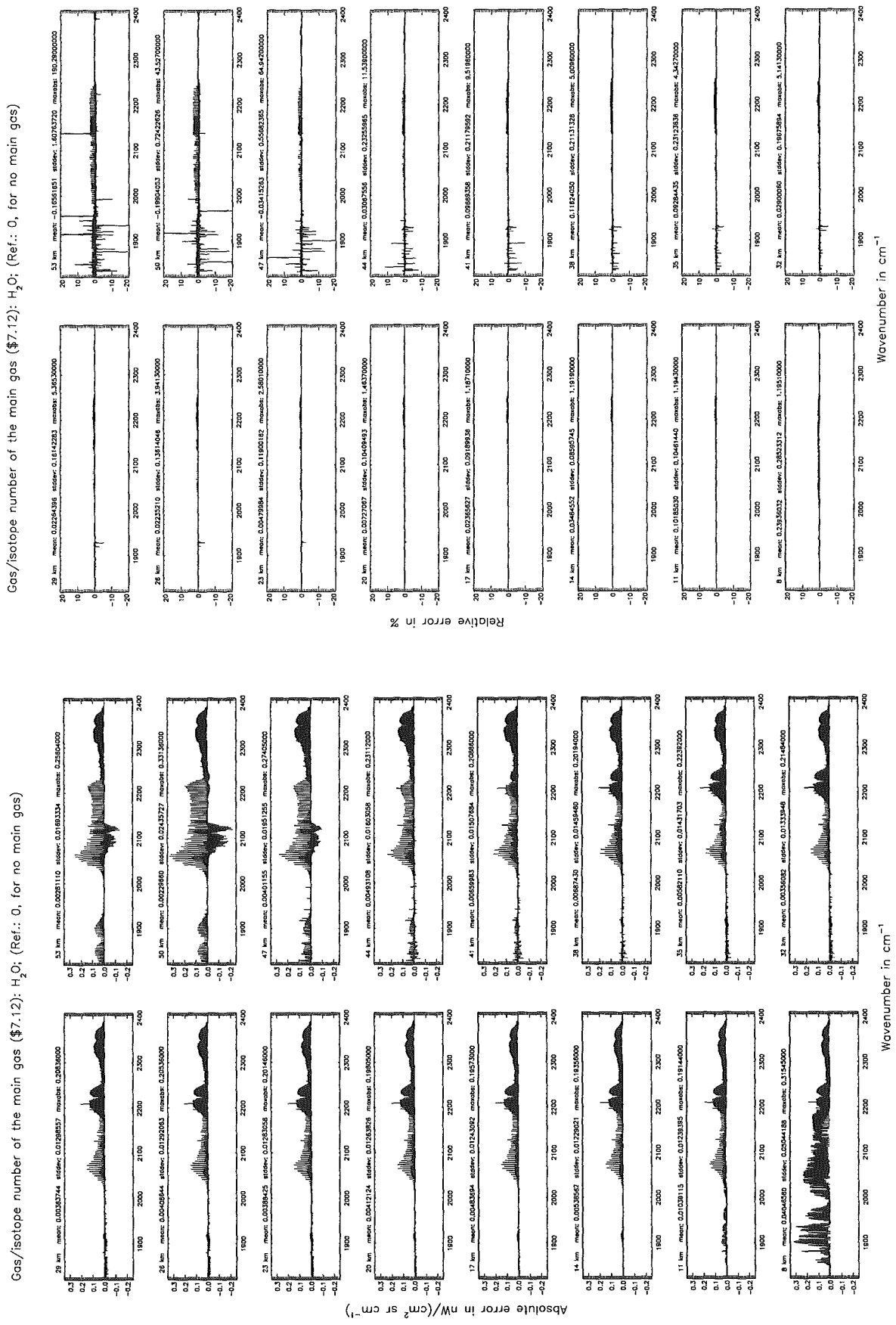


Figure 315: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

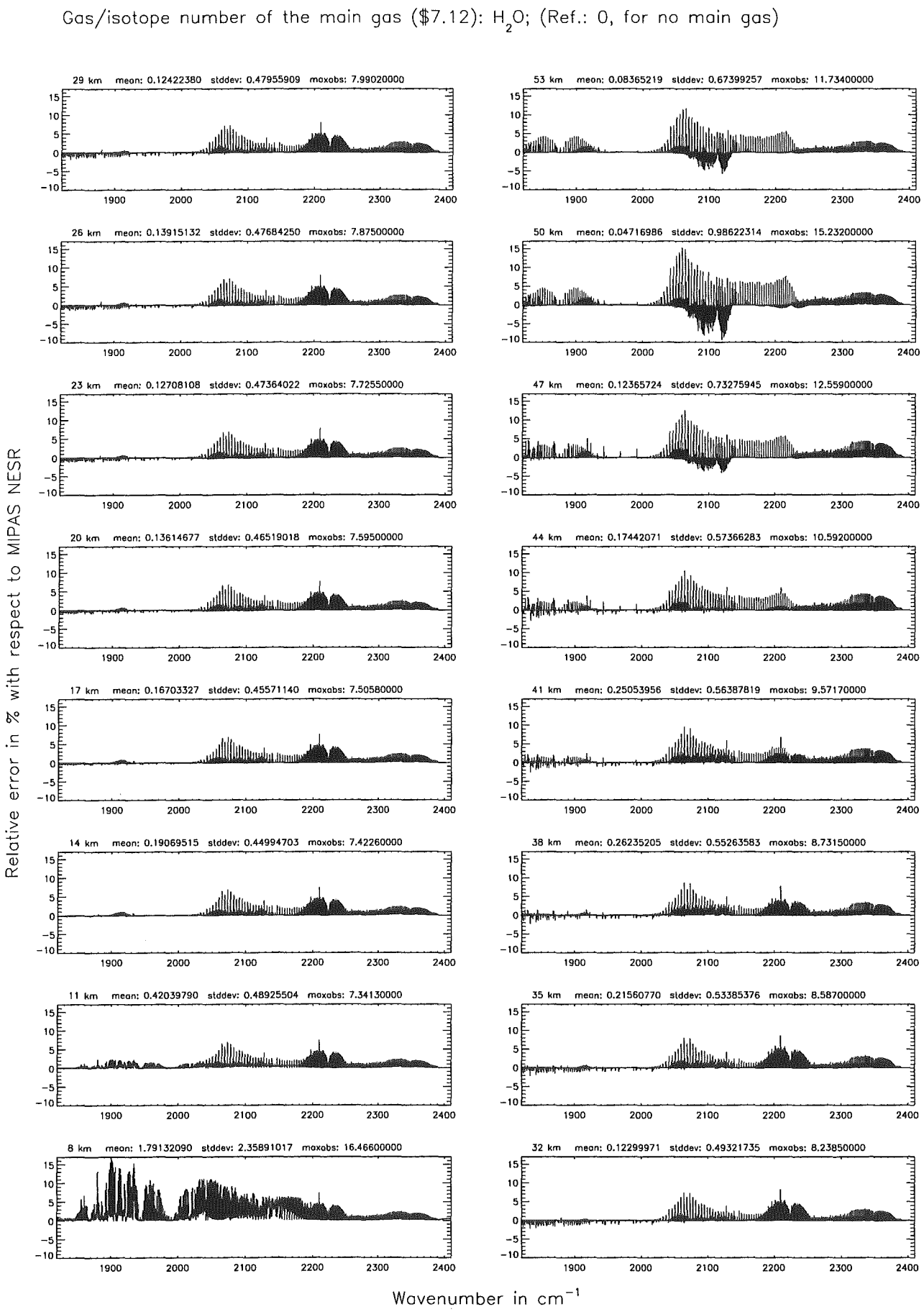


Figure 316: Relative error [%] with respect to MIPAS NESR

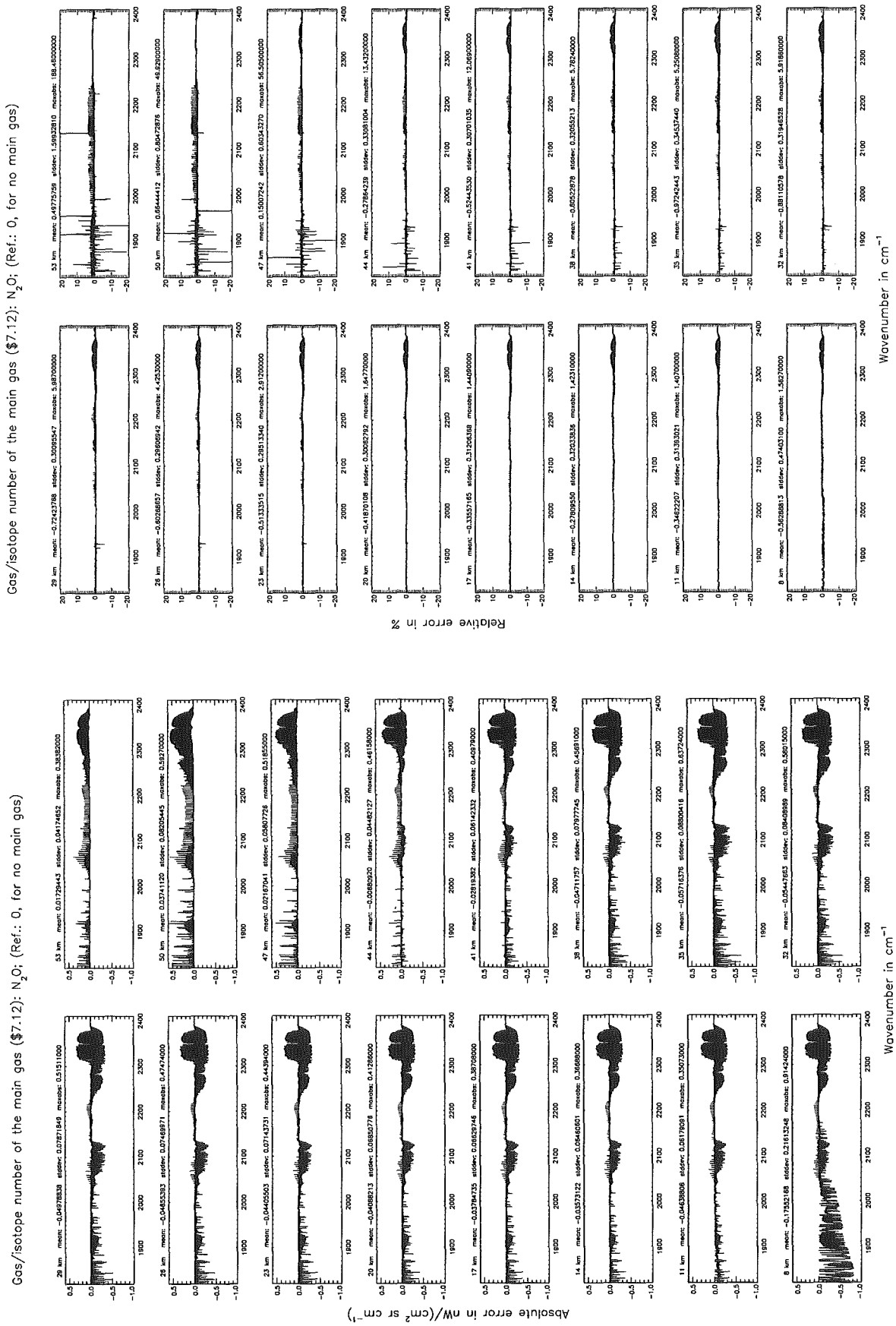


Figure 317: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

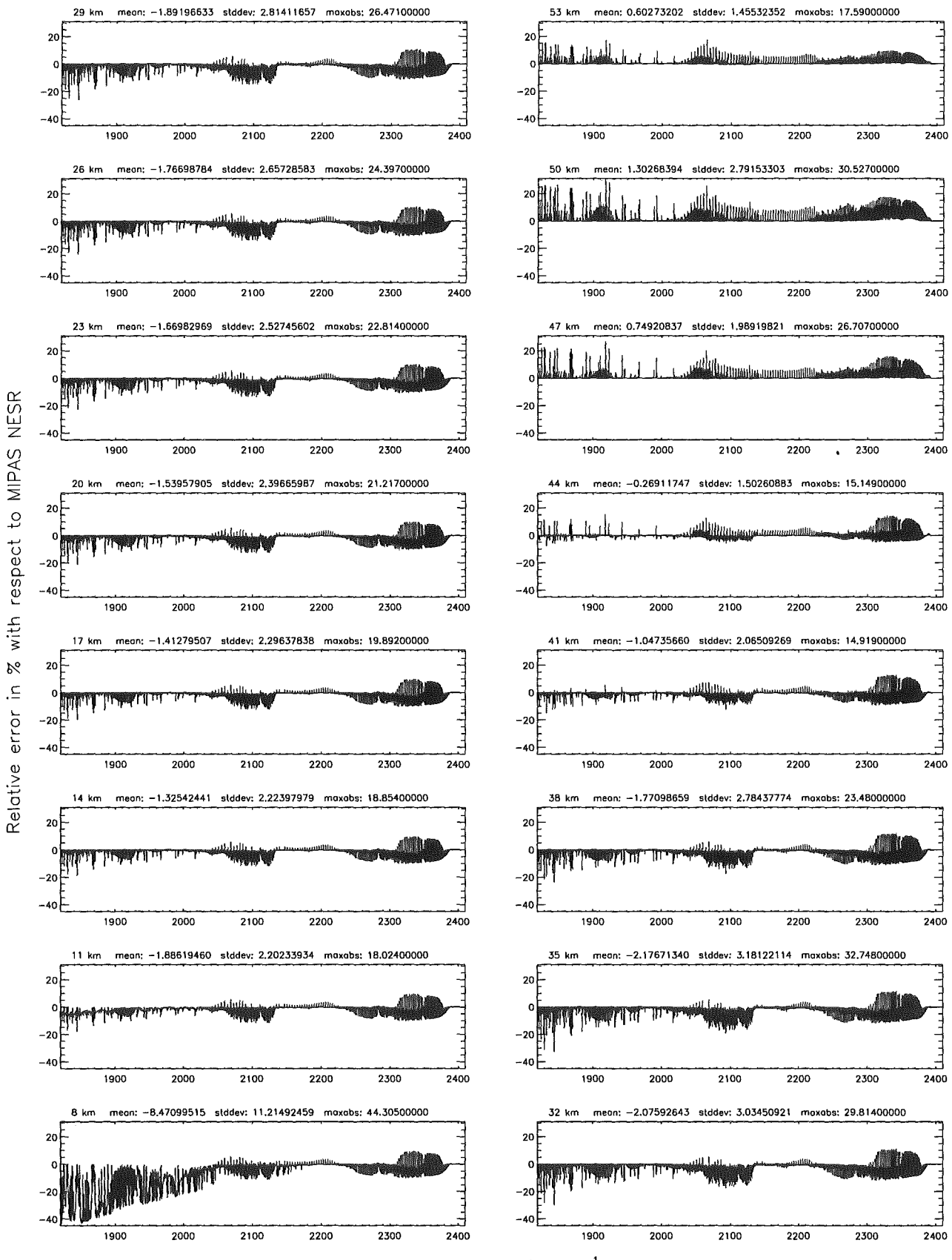


Figure 318: Relative error [%] with respect to MIPAS NESR

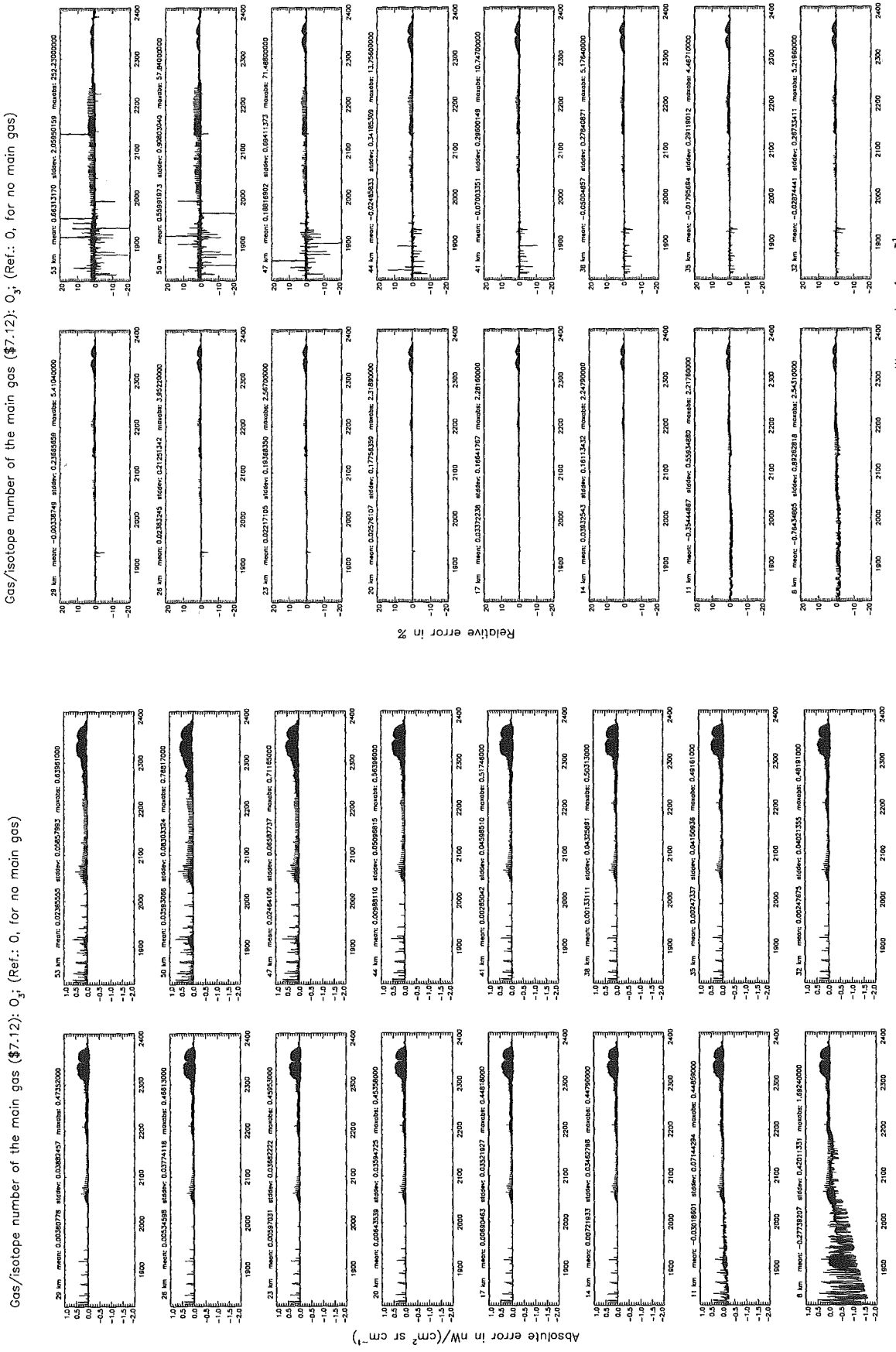


Figure 319: Absolute error [nW/(cm^2 sr cm^{-1})] and relative error [%]

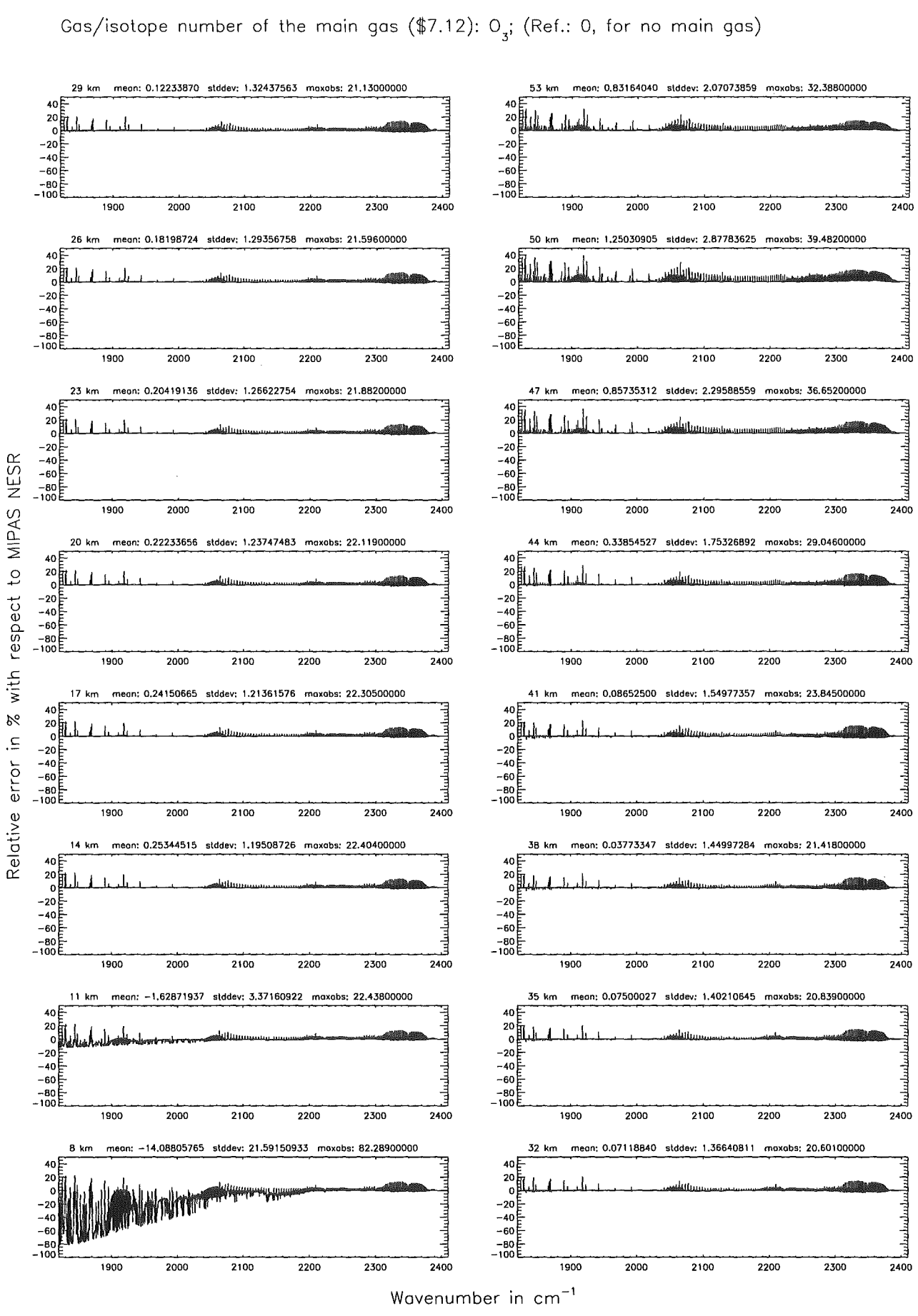


Figure 320: Relative error [%] with respect to MIPAS NESR

Intercomparison of the KOPRA and the RFM radiative transfer codes

N. Glatthor, M. Höpfner, G.P. Stiller,
T. von Clarmann, A. Dudhia, G. Echle,
B. Funke, and F. Hase

Abstract: We discuss the intercomparison between the Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA) and the Reference Forward Model (RFM) codes. The purpose of this intercomparison is to validate the KOPRA algorithm, i.e. to identify and to remove possible errors in the KOPRA (or RFM) code and to quantify the reason of remaining differences. A similar comparison between the MIPAS Optimized Forward Model (OFM) and the RFM has already been performed (Ref. [1]). To be able to relate on these results, the KOPRA validation is similarly organized: We also perform subsequently more complex tests of ray-tracing, integrated column amounts, homogeneous and limb path calculations of unapodized, apodized and field-of-view (FOV) convolved spectra, using the same isolated CO_2 line as well as the same six MIPAS microwindows. Additionally we compare the modeling of CO_2 line-mixing, non-local thermodynamic equilibrium (NLTE), trace gas continua, cross-section spectra and of derivatives of the spectra with respect to atmospheric parameters.
The KOPRA-RFM residuals are below a quarter of the noise-equivalent spectral radiance (NESR) for the isolated CO_2 line as well as for the MIPAS microwindows, i.e. KOPRA fulfills the acceptance criteria requested for the OFM. In most cases the deviations are even clearly below 1 nW/(cm^2 sr cm^{-1}), i.e. more than one order of magnitude below the acceptance threshold. This is valid for unconvolved as well as for ALS (apodized line shape) and FOV convolved spectra. There is also good agreement in modeling of the H_2O-, O_2- and N_2-continua and of CO_2 line-mixing. Larger deviations of up to several nW/(cm^2 sr cm^{-1}) were found for NLTE calculations on the basis of the "default" atmospheric profiles with vertical resolution of 1 or 2.5 km. These differences were found to be due to different layer-averaging of the vibrational temperatures and could be considerably reduced by calculations with a higher vertical resolution of 250 m. Cross-section spectra agree well, if the tabulated data are given independent of pressure, e.g. for $ClONO_2$ and N_2O_5, and cover the atmospheric temperatures. Due to different temperature extrapolation the deviations increase up to 10 nW/(cm^2 sr cm^{-1}) for atmospheric temperatures outside the tabulated range. The RFM is not yet adjusted to cross-sections given for non-equidistant temperatures and for atmospheric pressures, like CFC data in the HITRAN96 database. If these data are used, larger differences arise, e.g. up to 30 nW/(cm^2 sr cm^{-1}) between CFC-12 spectra. Avoidance of interpolation by performing homogeneous path calculations for p,T of one of the tabulated cross-section datasets reduces the deviations to below 0.5 nW/(cm^2 sr cm^{-1}).

1 Introduction

The Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA) has been developed for data analysis of the Michelson Interferometer for Passive Atmospheric Sounding (MIPAS), which will be launched on the polar-orbiting Environmental Satellite 1 (ENVISAT-1). KOPRA is a line-by-line model of high numerical accuracy, which (as far as foreseeable) can handle all atmospheric and instrumental effects necessary for the modeling of the MIPAS spectra, e.g. ray-tracing for an ellipsoidal earth shape including refraction, horizontal atmospheric inhomogeneities, NLTE conditions, CO_2 line-mixing, cross-sections to model the spectra of heavy molecules (CFCs, $ClONO_2$, N_2O_5), varying abundances of isotopomeres, parameterization of the instrumental line shape (ILS), and modeling of the instrumental FOV with regard to refraction.
KOPRA has a flexible structure enabling fast calculations for routine operation but also dedicated scientific studies. Most of the special effects just mentioned can be switched on or off, and the required accuracy for the calculation of the optical depth, for the layering of the atmosphere, for the apodization function and for other

parameters influencing computational speed can be varied by the user. Whereas KOPRA's default accuracy for the calculation of the optical depth is 10^{-5}, the spectra presented here were generally calculated with an accuracy of 10^{-10}. But we will exemplarily show the convergence of the KOPRA spectra towards the RFM results for increasing accuracy. A more detailed description of KOPRA's concept is given in Ref. [2].
The MIPAS Reference Forward Model (RFM) has been derived from GENLN2 to generate "true" reference spectra for validation of the operational codes for the analysis of the MIPAS data (Ref. [3]). Therefore the RFM is designed to perform calculations with high accuracy rather than computational speed.
In this document we compare KOPRA Version 0.4 with RFM Version 3.0 (Version 3.6 for NLTE calculations), which were supplied by the University of Oxford. For readers interested in a more compact presentation of the intercomparison we refer to Ref. [4].

2 Intercomparison Set-Up

2.1 Acceptance Criteria

The forward model errors should be significantly below the uncertainty of the MIPAS experiment and of the retrieval error. We set an acceptance threshold of 25% of noise-equivalent spectral radiance (NESR) for differences between KOPRA and RFM spectra, i.e. approximately
12.5 nW/(cm^2 sr cm^{-1}) in channel A (685 - 970 cm^{-1}),
5 nW/(cm^2 sr cm^{-1}) in channel B (1215 - 1500 cm^{-1}).
1 nW/(cm^2 sr cm^{-1}) in channel D (1820 - 2410 cm^{-1}).

2.2 Data Bases and Assumptions

Generally we use the same databases and make the same assumptions as in the OFM-RFM intercomparison.

Atmosphere

FASCODE 'model6' atmosphere (1976 US Standard); additionally a midlatitude HNO_3 profile of a climatology (Ref. [5]). For LTE calculations the original atmospheric profiles with 50 levels between 0 and 120 km were used, i.e. 1 km vertical spacing up to 25 km, 2.5 km spacing between 25 km and 50 km and 5 km spacing above 50 km. NLTE modeling of CO_2 was performed using profiles with 1 km and 250 m vertical sampling. NLTE-spectra of NO were calculated for an atmosphere extending up to 200 km and 2.5 and 250 m height resolution. For the RFM the Earth's radius of curvature at the tangent point was set to 6367.421 km, whereas KOPRA calculations were performed for an ellipsoidal Earth with a major axis of 6378.137 km, a minor axis of 6356.752 km (WGS84) and southward observation with tangent point at 45°N. To be able to compare both codes using the same limb geometry, a "downgraded" KOPRA-version calculating with a spherical Earth of radius 6367.421 km was additionally created.

Line Data

Line data were taken from the HITRAN96 line data base (Ref. [6]) and total internal partition sums (TIPS) were calculated according to the HITRAN96 TIPS code. For HNO_3, updated TIPS provided by Gamache were used for both codes (Gamache, pers. commun., 1997).

Table 1: CO_2 column amounts U calculated by KOPRA and RFM for different homogeneous atmospheres, path length = 250 km, CO_2 *vmr* = 330 ppmv.

Press. [hPa]	Temp. [K]	U_{KOPRA} [kmole/cm^2]	U_{RFM} [kmole/cm^2]
250.000	220	1.12755×10^{-4}	1.12755×10^{-4}
2.50000	250	9.92241×10^{-7}	9.92241×10^{-7}
0.20000	240	8.26868×10^{-8}	8.26868×10^{-8}
0.01000	200	4.96121×10^{-9}	4.96121×10^{-9}
0.00004	300	1.32299×10^{-11}	1.32299×10^{-11}

Table 2: Path lengths S through the 1976 US Standard Atmosphere calculated by KOPRA and RFM for different tangent heights and spherical Earth, refraction included.

Tght. [km]	S_{KOPRA} [km]	S_{RFM} [km]	Diff. [km]
10	1208.977	1208.975	0.002
20	1139.060	1139.063	-0.003
40	1015.850	1015.850	0.000
60	880.291	880.291	0.000
80	719.302	719.302	0.000

Line Shape

Apodization was performed in spectral space with the Norton-Beer no. 3 (strong) function (Ref. [7]); instrumental self-apodization and other line shape distortions were not considered.

Field-of-View

The FOV was modeled by the trapezoidal vertical response function defined by ESA, which is constant up to ±1.4 km apart from the tangent point and then decreases linearly to zero at ±2.0 km distance.

Interpolation

Both KOPRA and the RFM linearly interpolate log pressure ($\ln p$), temperature T and volume mixing ratio *vmr* between profile levels.

Wavenumber Grids

Two different wavenumber grids were used:

(a) a fine grid with 0.0005 cm^{-1} spacing for unapodized spectra and

(b) a coarse grid with 0.025 cm^{-1} spacing for apodized spectra.

The fine grid is commonly used for middle-atmosphere modeling (Doppler broadening typically > 0.001 cm^{-1}). The coarse grid will be the standard format for the representation of apodized MIPAS spectra.

Table 3: CO_2 column amounts U calculated by KOPRA and RFM for different limb paths through the 1976 US Standard Atmosphere and spherical Earth, refraction included.

Tght. [km]	U_{KOPRA} [kmole/cm^2]	U_{RFM} [kmole/cm^2]	Diff. %
10	1.26011×10^{-4}	1.26011×10^{-4}	0.000
20	2.57658×10^{-5}	2.57661×10^{-5}	-0.001
40	1.23681×10^{-6}	1.23689×10^{-6}	-0.006
60	9.67691×10^{-8}	9.67899×10^{-8}	-0.021
80	5.03587×10^{-9}	5.03652×10^{-9}	-0.013

Table 4: Curtis-Godson pressures calculated by KOPRA and RFM for several segments of a 10 km tangent-height path for N_2O (1976 US Standard Atmosphere). Calculations of both codes for spherical Earth.

Segment [km]	$\bar{p}_{g,KOPRA}$ [hPa]	$\bar{p}_{g,RFM}$ [hPa]	Rel. Diff.
10-11	$2.524114\times10^{+2}$	$2.524111\times10^{+2}$	1.19×10^{-6}
20-21	$5.137634\times10^{+1}$	$5.137519\times10^{+1}$	2.24×10^{-5}
40-42.5	$2.502353\times10^{+0}$	$2.502152\times10^{+0}$	8.03×10^{-5}
95-100	5.482884×10^{-4}	5.481659×10^{-4}	2.23×10^{-4}

Table 5: Curtis-Godson temperatures calculated by KOPRA and RFM for several segments of a 10 km tangent-height path for N_2O (1976 US Standard Atmosphere). Calculations of both codes for spherical Earth.

Segment [km]	$\bar{T}_{g,KOPRA}$ [K]	$\bar{T}_{g,RFM}$ [K]	Diff.
10-11	221.2131	221.2130	0.0001
20-21	217.1286	217.1287	-0.0001
40-42.5	253.3470	253.3484	-0.0014
95-100	191.1491	191.1504	-0.0013

3 Path Lengths and Integrated Column Amounts

First we compared integrated column amounts for homogeneous paths and then the ray-tracing algorithms for limb-paths, i.e. path lengths, integrated column amounts and Curtis-Godson pressures and temperatures. The integrated column amount U_g (kmoles/cm^2) is the primary path integral

$$U_g = \int_{path} \rho_g ds, \tag{1}$$

whereby ρ_g is the molar absorber density (kmoles/cm^3) of gas g and ds the path element. Both KOPRA and the RFM approximate inhomogeneous limb-paths by a number of segments, each having constant absorber amount u_g (kmoles/cm^2), pressure and temperature:

$$u_g = \int_{seg} \rho_g ds, \tag{2}$$

$$\bar{p}_g = \frac{1}{u_g} \int_{seg} p \rho_g ds, \tag{3}$$

$$\bar{T}_g = \frac{1}{u_g} \int_{seg} T \rho_g ds. \tag{4}$$

The weighted quantities $\bar{p}_g$ and $\bar{T}_g$ are the Curtis-Godson pressure and temperature of gas g.

Table 1 shows CO_2 column amounts calculated for homogeneous paths through 5 different atmospheres with constant p, T and vmr (cuvette calculations). In the frame of the representation there is no difference between the KOPRA and RFM results.

To have the same limb geometry for both codes, a "downgraded" KOPRA version calculating with a spherical Earth of radius 6367.421 km was created (change in module param_m.f90). In KOPRA, ray-tracing is performed by calculating the differential displacement of the ray path (Ref. [8]), whereas the RFM uses Snell's law applied to a sphere (Ref. [3]). However, the path lengths through the 1976 US Standard Atmosphere, considering refraction, are in very good agreement (Table 2). For tangent altitudes of 40 km and above they differ by less than 1 m, towards lower altitudes they slightly increase to -3 m at 20 km and to 2 m at 10 km.

The difference between integrated CO_2 column amounts is less than 0.001% for 10 km tangent height and increases to -0.02% at 60 km altitude (Table 3). Similar deviations were found in the OFM-RFM intercomparison. The differences between Curtis-Godson pressures for a 10 km tangent-height path for N_2O increase from 0.0001% for the lowermost segment to 0.02% for the segment from 90 to 95 km (Table 4). The differences between the respective Curtis-Godson temperatures remain below 2×10^{-3} K for each segment (Table 5). We conclude that the differences in path-lengths, column amounts and Curtis-Godson parameters are negligible.

4 Spectral Calculations for Homogeneous Paths

To compare the line shapes unaffected by differences resulting from ray-tracing, we start with homogeneous paths.

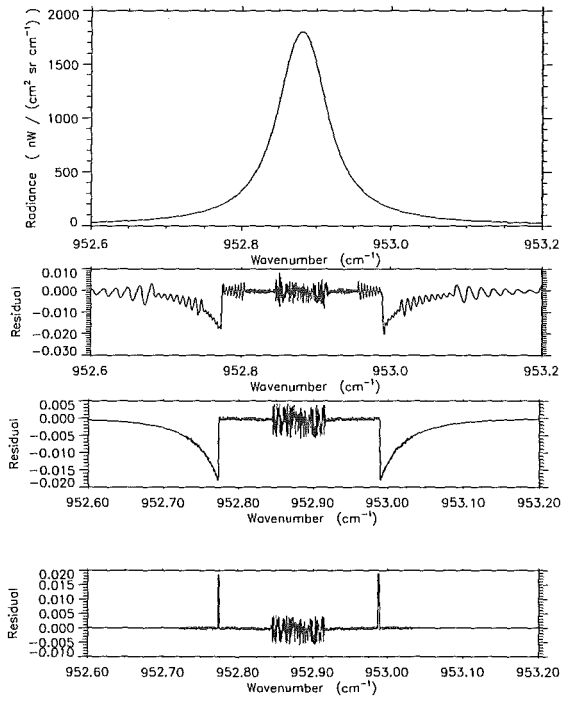


Figure 1: Unapodized radiance spectra of an isolated CO_2 line for homogeneous path conditions of 250 hPa and 220 K (top). KOPRA-RFM differences (in the same units as the radiance spectra) for calculations including χ-factor for KOPRA accuracy of 10^{-5} (second) and 10^{-10} (third) and for calculations without χ-factor (bottom) for KOPRA accuracy of 10^{-10}. The two steps at the wings of the first and second residual and the two spikes in the bottom residual are due to different Humlicek algorithm realizations.

4.1 Voigt Line Shape

Atmospheric lines are generally well described by the Voigt line shape, which is a convolution of the Lorentz and Doppler line shape functions. Both codes evaluate the Voigt line shape either by the original Humlicek algorithm (Ref. [9]), whose accuracy is 10^{-4}, or by modified versions for faster calculations (Ref. [10, 11]). The Humlicek algorithm has four evaluation regions and switches to Doppler line shape near line center and to Lorentzian line shape at the line wings.

Tests of line shape modeling were performed for the same isolated CO_2 line at 952.880858 cm^{-1} and for the same three different regimes as in Ref. [1]:

(a) Lorentz: p=250 hPa, T=220 K (~atmospheric conditions at 10 km altitude),

(b) Mixed: p=2.5 hPa, T=250 K (~40 km altitude),

(c) Doppler: p=0.2 hPa, T=240 K (~60 km altitude).

For each case a CO_2 *vmr* of 330 ppmv (1976 US value) and a pathlength of 250 km (approximately the length of a limb path through a 5 km layer above the tangent point) were assumed. Generally, CO_2-spectra were calculated including χ-factor (cf. Section 6.1), which by default is taken into account for CO_2-line modeling by KOPRA. For RFM calculations the χ-factor can be switched on or off by an external flag. Therefore a modified KOPRA-version was created (change in module abco_m.f90) to compare spectra modeled without χ-factor for the Lorentzian case. Figure 1 shows the resulting KOPRA and RFM spectra[1] (top) and the residual

[1]The deviations between KOPRA and RFM presented here are only visible in the difference

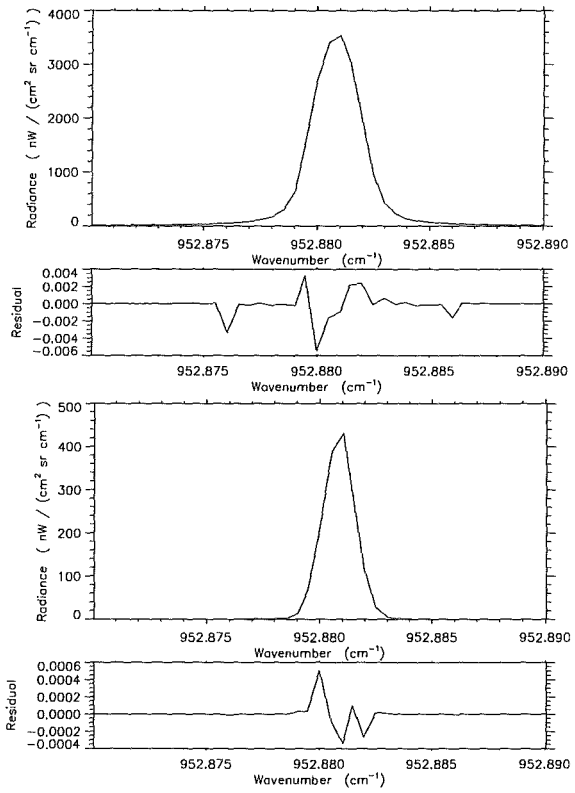


Figure 2: Unapodized radiance spectra and KOPRA-RFM differences for an isolated CO_2 line for homogeneous path conditions of 2.5 hPa and 250 K (top) and of 0.2 hPa and 240 K (bottom). KOPRA calculations of the optical depth with an accuracy of 10^{-10}.

spectra for the Lorentzian case including (second, third) and without χ-factor (bottom), whereby KOPRA's accuracy for the calculation of the optical depth was 10^{-5}, 10^{-10} and 10^{-10}, respectively. The weak oscillations in the case of default accuracy (10^{-5}) are due to quadratic interpolation of the absorption coefficients on the fine grid by KOPRA. In the case of high accuracy KOPRA calculates all absorption coefficients explicitely and the oscillations disappear. The steps of -0.02 nW/(cm^2 sr cm^{-1}) at the wings of the first two residual spectra are due to discontinuities between the different evaluation regions of the fast Humlicek algorithm which was applied by KOPRA to calculate the Voigt profile. When the CO_2 χ-factor is included, the RFM takes the standard Humlicek algorithm, which does not produce these features. However, for default calculations without χ-factor the RFM also uses the fast Humlicek algorithm, which results in a nearly complete compensation of the discontinuities in the lowermost residual. The remaining spikes of 0.02 nW/(cm^2 sr cm^{-1}) are caused by slightly different extensions of the evaluation regions (see also Appendix A). However, the magnitude of the discontinuities remains below 10^{-4}, which is anyway the accuracy limitation of the Humlicek approximation. At the line center, the KOPRA-RFM differences are less than 0.005 nW/(cm^2 sr cm^{-1}) or less than 0.001%.

For the mixed and Doppler cases (Figure 2) the agreement with the RFM is very

spectra.

good, with residuals of less than 0.006 nW/(cm^2 sr cm^{-1}) and less than 0.0006 nW/(cm^2 sr cm^{-1}) at the line center, respectively. The spectral region shown is more confined to the line center than in Figure 1 and does not contain the discontinuities, because the radiances at these locations have nearly decreased to zero. E.g. for the mixed case the radiance at 953 cm^{-1} is below 0.1 nW/(cm^2 sr cm^{-1}) and the step is below 10^{-5} nW/(cm^2 sr cm^{-1}), which again means a relative discontinuity of 10^{-4}.
We conclude that for the Lorentzian case, apart from the steps at the line wings, the errors caused by the line shape approximations of KOPRA are below 0.1% of the acceptance threshold. The discontinuities are below 0.2% of NESR/4. For the mixed case the absolute uncertainties are of the same order, and for the Doppler case even lower by one order of magnitude.

4.2 Line Wing Assumptions

For spectral calculations the RFM covers the microwindow and the surrounding region by a wide-mesh of width 1 cm^{-1} (at integer wavenumbers). Inside the microwindow fine grid calculations are performed for lines centered in the respective and the two neighboring 1 cm^{-1} intervals. The absorption of further remote lines is only calculated at the edges and the centers of the wide-mesh intervals, added up and interpolated to the fine grid. KOPRA has a variable spacing for each line considered with respect to the required accuracy (Ref. [12]). The contributions of the lines are added up and interpolated quadratically to the next finer grid.
Both codes include line wing contributions of lines located in a user-defined region around the microwindow (default value ±25 cm^{-1}). For KOPRA calculations the extension region is related to the microwindow's edges and therefore the same for each line inside the microwindow. This area is further enlarged by the width of the apodization function (cf. Section 5.1.2). The RFM's extension region is applied to each edge of the 1 cm^{-1} intervals. Thus, the concepts are a little different, but approach for small microwindows and low accuracy apodization functions. Both codes can model the contribution of even further remote H_2O, CO_2, N_2 and O_2 lines by pressure and temperature dependent continua.
Line wing modeling for different microwindow extensions was compared for a H_2O-microwindow from 1413.9 to 1416.4 cm^{-1} (cf. Ref. [1]). This microwindow is influenced by the wings of strong H_2O lines just outside the window boundaries. The following atmospheric conditions were chosen: Homogeneous path of 250 km length, $p = 250$ hPa, $T = 220$ K and H_2O *vmr* = 10 ppmv. Five tests are presented:
(a) Using only lines centered inside the microwindow,
(b) Using lines within ±32 cm^{-1} from the microwindow's edges for KOPRA and ±25 cm^{-1} for RFM,
(c) Using lines within ±25 cm^{-1} from the microwindow's edges for both programs,
(d) Same as (c), but area of fine-mesh calculations of the RFM extended to ±5 cm^{-1}.
(e) Same as (c), but additionally H_2O continuum, i.e. contribution of line wings beyond ±25 cm^{-1}, taken into account.
The results of the tests are shown in Figure 3. The spectra of the cases (c) and (d) are not plotted, since they are nearly identical with those of case (b). For better interpolation of Lorentzian line wing contributions of strong H_2O lines outside the microwindow the RFM was run with the inverse quadratic (IQD) option (cf. Ref. [1]). When only lines from inside the microwindow are taken into account (a), the agreement is very good with differences generally smaller than 0.002 nW/(cm^2 sr cm^{-1}). The two spikes of 0.006 nW/(cm^2 sr cm^{-1}) and the smaller spikes at 1414.7 and 1415.2 cm^{-1} are caused by discontinuities in the fast Humlicek algorithms (cf. Section 4.1), which are slightly closer to the line center in the RFM code. For

case (b) there is an offset of 0.25 nW/(cm^2 sr cm^{-1}), because KOPRA adds up line wing contributions of a larger spectral region. At the line centers the offset decreases or disappears due to saturation. This case shows that one has to keep in mind the differences in microwindow extension, when comparing KOPRA and RFM spectra. For exactly the same extension (c) the differences become much smaller. They oscillate around zero with an amplitude of 0.02 nW/(cm^2 sr cm^{-1}) and increase slightly to ±0.05 nW/(cm^2 sr cm^{-1}) at the microwindow's edges. The disappearance of these oscillations at 1414.0, 1414.5 and 1415.0 cm^{-1} and for RFM calculations with an extended fine-mesh area of ±5 cm^{-1} (d) shows, that they are caused by the line wing interpolation of the RFM (see also Appendix A). For case (e) the residuals are nearly the same as for case (c), which means that inclusion of the H_2O continuum causes no additional problems.

5 Spectral Calculations for Limb Paths

Now we compare spectral calculations for atmospheric limb-paths through the 1976 US Standard Atmosphere.

5.1 Isolated CO_2 Line, 10 km Tangent Height

5.1.1 Unconvolved Spectra

For unconvolved spectra the differences are 1-2 orders of magnitude less than NESR/4 (Figure 4). The residuals of 0.35 nW/(cm^2 sr cm^{-1}) at the line center in Figures 4b and 4c are mainly caused by the different Earth shapes assumed, which has the strongest effect at high altitudes far away from the tangent point. Non-standard KOPRA calculations for a spherical Earth reduce the spike to -0.1 nW/(cm^2 sr cm^{-1}) (Figure 4d); the remaining residual is due to slightly different Curtis-Godson temperatures. The weak oscillations in the case of KOPRA calculations with default accuracy (Figure 4b) disappear for higher accuracy (Figures 4c,d). The discontinuities at the line wings are due to different realizations of the Humlicek algorithm (cf. Section 4.1).

5.1.2 ALS Convolution

In this paper the ILS of MIPAS was assumed to be a sinc function (no self-apodization and phase errors). To simulate the apodization of the interferograms measured by MIPAS, the CO_2 line was convolved with the Norton-Beer no. 3 (strong) apodization function. KOPRA can calculate several apodization functions internally or read an external line shape function defined by the user. According to the accuracy chosen, the spectral extension of the strong Norton-Beer apodization function for 20 cm optical path difference is ±0.0755 cm^{-1}, ±0.7005 cm^{-1} or ±7.0005 cm^{-1}, respectively. The apodization function of the RFM is always supplied externally.

KOPRA normalizes the apodization function to its area up to infinity. It does not renormalize the truncated apodization function, which covers slightly less than unit area, because this concept leads to true trace gas column amounts for the retrieval of well resolved lines. The RFM renormalizes the apodization function, which is more adequate for continuum-like signatures. This results in slightly stronger apodized spectral lines, but the differences are small. For example, KOPRA's medium accuracy apodization function covers more than 99.96% of unit area.

To provide exactly the same apodization function for both KOPRA and RFM calculations, the medium accuracy strong Norton-Beer function of KOPRA was truncated to 2001 values (default RFM limitation) and renormalized to unit area

(ils2001.nor). To show the influence of different spectral extension and normalization of the apodization functions, we also present KOPRA spectra convolved with the medium accuracy function, consisting of 2801 spectral points, and with the not renormalized version of the truncated function (ils2001.abs).
Figure 5 shows the results of the ALS convolution. The medium accuracy function, which has a larger spectral extension than the RFM's function, produces a residual of -1.5 nW/(cm^2 sr cm^{-1} at the line center (Figure 5b). For KOPRA calculations with ils2001.abs, which covers slightly less than unit area, the differences are reduced by 50% (Figure 5c). Application of the renormalized apodization function ils2001.nor leads to residuals below 0.03 nW/(cm^2 sr cm^{-1}), i.e. far below 0.1% of NESR/4.

5.1.3 ALS and FOV Convolution

In the case of a vertically extended FOV the spectrum also contains contributions from below and above nominal tangent height, where different atmospheric conditions exist. KOPRA and the RFM model this effect by calculating point-radiance spectra inside and around the vertical FOV (more than 20 possible) and adding up these spectra, weighted by a FOV function. The shape of KOPRA's weighting function can be defined on 20 horizontal strips, the RFM's weighting function at up to 21 vertical grid points.
The RFM-version used for this comparison makes two simplifications in FOV modeling: (a) The vertical extension and the shape of the FOV weighting function is defined in distance (km). This results in a non-physical instrumental aperture dependent on tangent height, when the same weighting function is used for a total limb scan.
(b) Distortion of the FOV due to refraction is not taken into account.
KOPRA's concept is closer to the physical conditions: The FOV weighting function and its partitioning are defined in terms of elevation angle, which leads to a decrease (in distance) of the vertical extension of the FOV with increasing tangent height. Since refraction is also taken into account, the weighting function is additionally distorted.
We compare spectra convolved with the trapezoidal FOV weighting function originally defined by ESA, which is constant until ±1.4 km from nominal tangent height and decreases to zero at ±2 km (cf. Section 2.2). This function can be exactly modeled by the RFM. Due to consideration of refraction, the trapezoid is distorted within KOPRA. Moreover, for nominal instrumental aperture (half angle of 6.1×10^{-4} rad, assessed geometrically without consideration of refraction), KOPRA's FOV is vertically more extended at low tangent heights (e.g. ~4.3 km at 10 km) and less extended at higher altitudes. To obtain nearly consistent conditions for both codes, the FOV half angle of KOPRA was adjusted to 2 km for each tangent height, i.e. to 5.662 $\times10^{-4}$ rad for 10 km. Further, we used a 12-point FOV representation for the RFM, which had been a good compromise between accuracy and computation time in the OFM-RFM intercomparison, and correspondingly represented the FOV of KOPRA by 11 or 13 point-radiance spectra (only odd numbers possible).
As expected, there is poor agreement for KOPRA calculations with the half angle of 6.1 $\times10^{-4}$ rad (Figure 6b). The differences amount to 4 nW/(cm^2 sr cm^{-1}) at the line center and to 11 nW/(cm^2 sr cm^{-1}) at the line wings. KOPRA calculations with the RFM-adjusted half angle result in much better agreement. For 11 point-radiance spectra the difference is reduced to 0.1 nW/(cm^2 sr cm^{-1}) at line center and to 0.4 nW/(cm^2 sr cm^{-1}) at the line wings (Figure 6c). For 13 spectra the residuals at the line wings decrease further to -0.04 nW/(cm^2 sr cm^{-1}) (Figure 6d). This deviation is a factor of 3 more than for ALS convolved spectra, but still only

1% of NESR/4.

5.2 Isolated CO_2 Line, 40 km Tangent Height

The differences in the upper stratosphere are also very small. For unconvolved spectra and KOPRA calculations for an ellipsoidal Earth (Figure 7, top) the residuals are 0.05 nW/(cm^2 sr cm^{-1}) at the line center and -0.1 nW/(cm^2 sr cm^{-1}) at the line wings. KOPRA calculations for a spherical Earth remove the difference at the line center, but the deviation at the line wings remains. After convolution with the renormalized apodization function ils2001.nor (Figure 7, middle) the residuals decrease to -0.01 nW/(cm^2 sr cm^{-1}) at the line center and to 0.005 nW/(cm^2 sr cm^{-1}) at the line wings. FOV-convolution was performed with the trapezoidal weighting function in the same way as outlined in Section 5.1.3, but with a RFM-adjusted half-angle of 6.230×10^{-4} rad for KOPRA calculations. The differences at the line center is 0.02 nW/(cm^2 sr cm^{-1}), whereas the deviations at the line wings are smoothed out (Figure 7, bottom).

5.3 Full Microwindow Calculations

As a next step we performed spectral calculations for the same 6 MIPAS microwindows as in the OFM-RFM study:

(a) PT012A MW, 700.85-701.1 cm^{-1}, CO_2, 40 km,
(b) CH440A MW, 1355.0-1356.75 cm^{-1}, CH_4, 40 km,
(c) N2O20A MW, 1879.35-1880.3 cm^{-1}, N_2O, 10 km,
(d) O3014A MW, 763.5-764.65 cm^{-1}, O_3, 40 km,
(e) HNO08A MW, 888.5-891.2 cm^{-1}, HNO_3, 10 km,
(f) H2O17A MW, 1413.9-1416.4 cm^{-1}, H_2O, 10 km.

Beside the main target gas, the other 5 gases were also considered for each microwindow. Figures 8-13 show the spectra (a) and the differences for unconvolved (b), ALS convolved (c) and ALS and FOV convolved spectra (d) for each case.

5.3.1 Unconvolved Spectra

There is very good agreement between KOPRA and RFM spectra for the CH_4-, N_2O-, O_3-, HNO_3- and H_2O-microwindows (Figures 9b-13b). Due to the different Earth shapes assumed the maximum differences are generally at the line centers, but far below NESR/4. For O_3 they amount to -0.15 nW/(cm^2 sr cm^{-1}) and for CH_4 to 0.03 nW/(cm^2 sr cm^{-1}). These residuals could be further reduced by a factor of 4 by performing KOPRA calculations for a spherical Earth (cf. Section 5.1.1). The deviation of 0.2 nW/(cm^2 sr cm^{-1}) at the left edge of the H_2O-microwindow is caused by the line wing interpolation of the RFM (cf Section 4.2). The systematic offset of about -0.6 nW/(cm^2 sr cm^{-1}) in the CO_2-microwindow (Figure 8b) is not yet understood.

5.3.2 ALS and FOV Convolution

After convolution with the apodization function ils2001.nor the Doppler peaks in the line centers are smoothed out, which in most cases causes a reduction of the KOPRA-RFM residuals (Figures 8c-13c). However, the systematic offset in the p,T-microwindow is not removed (Figure 8c).
Finally the spectra were additionally convolved with the trapezoidal FOV weighting function in the same way as the isolated CO_2 line (Section 5.1.3). For the p,T- and O_3-microwindows the residuals are just as large as for ALS convolved spectra (Figures 8d, 11d). For CH_4 and N_2O they increase by one order of magnitude (Figures

9d, 10d), but are still far below NESR/4. Apart from two regions the agreement becomes only slightly worse for HNO_3 (Figure 12d). The most critical test of FOV calculations is the H_2O-microwindow (Figure 13d), because the H_2O profile has a strong change of gradient near the tangent height of 10 km. The resulting difference spectrum exhibits an offset of -0.1 nW/(cm^2 sr cm^{-1}) at the line wings and residuals of up to 0.35 nW/(cm^2 sr cm^{-1}) near the line centers. The reason of the offset is a slightly wider spacing of the point-radiance spectra calculated by KOPRA in the lower part of the FOV due to consideration of refraction.
The result of the full microwindow calculations is that for unconvolved, ALS convolved as well as for ALS and FOV convolved spectra the KOPRA-RFM differences fulfill the acceptance criteria, i.e. all residuals are well below NESR/4.

6 Special Physical Effects

6.1 CO_2 Line Mixing

Line mixing is the transfer of population between rotational-vibrational molecular states and redistribution of spectral intensity within a band, caused by collisions between the radiating molecule and a broadening gas. This effect is most significant at and near CO_2 Q-branches.
In the region of ± 10 cm^{-1} around the Q-branch head KOPRA models CO_2 line mixing following the approach of Strow et al. (Ref. [13, 14]). The user can either choose direct diagonalization of the relaxation matrix or the Rosenkranz first order approximation (Ref. [15]) and model either Q-branch- or QPR-branch-coupling. The RFM follows the same approach, but considers only Rosenkranz approximation and Q-branch-coupling. Further off the CO_2 Q-branch heads both codes parameterize line mixing and far wing effects by the so-called χ factors.
Test calculations for CO_2 line mixing were performed for limb paths with 10 km tangent height in the spectral region 787.5-797.5 cm^{-1}. This window contains the strong Q-branch at 791.6 cm^{-1}.
To show the magnitude of the effect, Figure 14 contains KOPRA calculations neglecting and including line-mixing (Rosenkranz approximation, Q-branch coupling). The largest difference of more than -600 nW/(cm^2 sr cm^{-1}) (for line mixing: lower values) appears at the left edge of the Q-branch ($\sim$791.2 cm^{-1}).
Figure 15a shows unapodized and ALS and FOV convolved KOPRA and RFM spectra with modeling of line mixing following the Rosenkranz approximation with Q-branch coupling. As a reference for the estimation of the residuals unapodized spectra calculated without line mixing are also presented. When line mixing is neglected, the residuals are very small with maximum differences of up to 0.5 nW/(cm^2 sr cm^{-1}) at the line centers and oscillations of about 0.1 nW/(cm^2 sr cm^{-1}) at the line wings (Figure 15b). As shown in Section 5, the former are mainly due to the different Earth shapes assumed and the latter due to the RFM's line wing interpolation. Inclusion of line mixing (Figures 15c,d) produces residuals very similar to the reference case. This shows that both codes model line mixing the same way. Again, the discontinuity of -0.4 nW/(cm^2 sr cm^{-1}) is probably caused by use of different Humlicek algorithms.

6.2 NLTE Calculations

In the lower and middle stratosphere most trace gases (except e.g. NO) are in local thermodynamic equilibrium (LTE), i.e. the populations of the lower and upper states of a vibrational-rotational transition are determined by the Boltzmann distribution evaluated at the local kinetic temperature. With increasing altitude

collisional energy exchange is no longer sufficient to balance radiative or chemical pumping and the partition of populations is characterized by a temperature different from the local kinetic temperature, i.e. non-local thermodynamic equilibrium (NLTE) conditions develop.

In both codes, deviations from LTE are considered in the absorption cross-sections and in the source function. The RFM models NLTE emissions on the basis of externally supplied vibrational temperature profiles (Ref. [16, 17, 18]), whereas KOPRA can either use external population ratio or vibrational temperature profiles. The two codes handle layer-averaging of the NLTE parameter profiles slightly different. KOPRA calculates Curtis-Godson layer-averages of the vibrational temperatures or population ratios exactly the same way as for kinetic temperatures. The RFM, however, interpolates the vibrational-kinetic temperature difference profile to the Curtis-Godson pressure of the path segment and then adds it to the Curtis-Godson kinetic temperature.

NLTE calculations for CO_2 were performed using temperature and pressure profiles of the 1976 US Standard Atmosphere. The vibrational temperature profiles of CO_2, provided by the Instituto de Astrofisica de Andalucia (IAA), extended up to 120 km with 1 km vertical spacing (Ref. [19]). As a reference, Figure 16a shows unconvolved CO_2 spectra and KOPRA-RFM differences for 10 km tangent height and LTE conditions. The residuals are small (<0.4 nW/(cm^2 sr cm^{-1})) and similar to those of the isolated CO_2-line (cf. Figure 4a). Unapodized NLTE calculations on the 1 km vertical grid exhibit residuals between 0 and 20 nW/(cm^2 sr cm^{-1}) at the line centers and of 2 nW/(cm^2 sr cm^{-1}) at the inner line wings (Figure 16b). The large and irregular residuals at the line centers[2] are caused by different neglection of CO_2-signatures from the uppermost atmosphere. For an atmosphere ending at 80 km altitude the differences between the Doppler kernels are more regular and reduced to ~-1.5 nW/(cm^2 sr cm^{-1}) (Figure 16c), whereas the residuals at the line wings remain the same. Much better agreement is achieved by using vertically highly resolved vibrational temperature profiles, consisting of 321 points with 250 m vertical spacing (Figure 16d). The residuals at the line wings nearly disappear, and the differences at the line centers of 1.2 nW/(cm^2 sr cm^{-1}) are just a factor of three larger than in the LTE-case. This indicates that the residuals for 1 km level-spacing are mainly due to the different averaging schemes of vibrational temperatures, which leads to larger deviations for wider layers.

Another set of NLTE calculations was performed for NO, using (a little unrealistic) *vmr*- and T_{vib}-profiles of the vibrational states 1, 11, 2 and 12 (HITRAN-notation) extending up to 200 km. Pressure and temperature profiles were taken from the MSIS90 model atmosphere for midlatitude March conditions. To investigate the effect of different layer-averaging of T_{vib}, the NLTE calculations were performed with 2.5 and 0.25 km vertical resolution, i.e. for 81 and 801 levels. Figure 17a shows unconvolved NO spectra and residuals for 10 km tangent height, 81 levels and LTE conditions. Like for CO_2, the residuals are very small (<0.004 nW/(cm^2 sr cm^{-1})). For NLTE calculations on the same vertical grid the KOPRA-RFM differences increase by a factor of 100 (Figure 17b). Again, the NLTE-residuals between the strong NO-lines at 1900.08 and at 1900.52 cm^{-1} decrease considerably, when the narrow vertical grid is applied (Figure 17c). The remaining difference at 1900.32 cm^{-1} results from transitions (13 -> 12 in HITRAN notation), for which no upper state vibrational temperature profiles were provided. In this case KOPRA calculates the LTE signature, whereas the RFM neglects the lines totally. When state 13 is formally represented by the vibrational temperatures and energy of state 12 in the input files, these lines also disappear in the KOPRA calculation and the artefact is removed (Figure 17d, only residuals plotted). Now it becomes obvious

[2]this effect does no longer appear for the KOPRA version of July 1999

that the NLTE-residuals are of the same magnitude as the LTE-residuals.

6.3 Continua

In addition to line-by-line absorption, continuum absorption must also be taken into account for strongly absorbing gases in atmospheric infrared spectroscopy. The largest continuum in the atmospheric windows around 4 μm and at 8-12 μm results from water vapor, for which both KOPRA and the RFM use the CKD.21 continuum model (Ref. [20]). They subtract the absorption coefficient evaluated at line center ± 25 cm^{-1} from each line and then add the continuum term. Differences can arise from lines being neglected due to their small contributions. The implementation of the H_2O continuum into the RFM code is discussed more in detail in Ref. [3]. In addition to H_2O, both codes enable calculation of a CO_2 continuum and of the pressure induced O_2 (Ref. [21] and N_2 absorption bands (Ref. [22]) at 1550 cm^{-1} and at 2350 cm^{-1}, respectively.
Figure 18 shows unapodized atmospheric limb spectra for 10 km tangent height. For H_2O the residua are very similar to the case without continuum (Figure 13b), which shows that also for limb paths both codes model the same continuum contribution. For O_2 and N_2 the KOPRA-RFM differences exhibit weak offsets of up to 0.015 and 0.004 nW/(cm^2 sr cm^{-1}), respectively. In addition there are weak residuals of up to 0.03 and -0.002 nW/(cm^2 sr cm^{-1}) at the line centers. All these deviations are far below NESR/4. (CO_2 continuum still to be validated)

6.4 Cross Section Spectra

To model the signature of heavy molecules as CFCs, $ClONO_2$, N_2O_5 and SF_6, for which in most cases line data are not available, both KOPRA and the RFM use laboratory measurements of absorption cross-sections. However, there are differences in data handling. The RFM was designed for HITRAN92 datasets, which are tabulated with equidistant temperature spacing and independently of pressure. Therefore this code interpolates linearly to atmospheric temperature, assuming equidistant T-spacing, but not to atmospheric pressure. Extrapolation beyond the tabulated temperature range is performed using the total internal partition sum. As far as available, KOPRA uses the most recent cross-section data, measured for various non-equidistant temperatures and pressures. Therefore it interpolates in temperature-pressure space, weighting the four (if available) nearest T,p-datasets in each of the quadrants adjacent to the observation by their reciprocal, normalized distance. T,p-extrapolation is performed similarly, using the tabulated data. The $ClONO_2$ cross-sections are interpolated reciprocally in temperature.
First we present $ClONO_2$ and N_2O_5 spectra, for which both codes use the same laboratory measurements of Ballard (Ref. [23]) and of Cantrell (Ref. [24]), respectively. Since these cross-sections are tabualated without specification of pressure, no differences result from KOPRA's pressure interpolation. The tabulated $ClONO_2$ dataset used by KOPRA was interpolated and extrapolated from the original measurements at 296 K and 213 K to 6 temperatures between 190 K and 296 K.
Figure 19 shows unapodized spectra of the ν_4 Q-branch of $ClONO_2$ for 10 km tangent height and a $ClONO_2$-profile measured by MIPAS-B on March 14, 1992 (Ref. [25]). First, the temperature profile of the 1976 US Standard Atmosphere was used. The uppermost KOPRA-RFM residual of Figure 19 exhibits a slightly negative offset at the wings and a positive deviation at the center of the Q-branch. Moreover there are various spikes of up to -40 nW/(cm^2 sr cm^{-1}), due to different wavenumber grids. Whereas the RFM maintained the fine grid, KOPRA changed to 0.008 cm^{-1} spacing. RFM calculations on the same grid remove the spikes (Figure 19, middle). The remaining residuals of -2 nW/(cm^2 sr cm^{-1}) at the wings

and of 8 nW/(cm^2 sr cm^{-1}) at the center of the Q-branch are due to different temperature extrapolation to the minimum of the temperature profile (216.7 K) as well as to different interpolation. However, spectra calculated for one of the temperature values (296 K), at which the cross-sections were measured, exhibit differences below 1.5 nW/(cm^2 sr cm^{-1}). In this case both codes do not interpolate and the agreement becomes as good as for line data gases.

Figure 20 shows unapodized atmospheric limb spectra of the ν_{12} band of N_2O_5 for 10 km tangent height. For the 1976 US Standard temperature profile the KOPRA-radiances are systematically below the RFM-values by about -12%, which means poor agreement (top). This is due to different temperature extrapolation from the "coldest" cross-section dataset (233 K) to the minimum of the temperature profile (216.7 K). For a constant temperature profile of 273 K, which is the temperature of one of the tabulated datasets, the residuals decrease to $\pm$0.04 nW/(cm^2 sr cm^{-1}) or $\pm$0.5% (bottom).

To model CFC-12 (CCl_2F_2) spectra, KOPRA takes cross-sections of Varanasi (Ref. [26]), which are given for various non-equidistant atmospheric temperatures and pressures. The RFM originally used another dataset, which is tabulated without specification of pressure (Ref. [27]). For the intercomparison the Varanasi dataset from the HITRAN96 database, consisting of 15 non-equidistant p,T-points, was supplied for both codes. Due to the non-appropriate interpolation scheme of the RFM, Figure 21 (top) exhibits residuals of up to 30 nW/(cm^2 sr cm^{-1}) even for homogeneous path conditions of 250 km, 250 hPa and 220 K. To avoid interpolation, homogeneous KOPRA and RFM calculations were performed for the tabulated values of 296.2 K and 790.5 Torr (1054 hPa) with 0.008 cm^{-1} resolution. Again, for this p,T-value the differences become smaller than 0.5 nW/(cm^2 sr cm^{-1}) (Figure 21, bottom).

6.4.1 Spectral Derivatives

Additionally to the calculation of the spectrum, KOPRA is able to determine the derivatives of the spectrum with respect to most of the retrieval parameters, e.g. to kinetic and vibrational temperature, volume mixing-ratio, horizontal gradients of temperature and volume mixing-ratio. These derivatives are determined analytically during one forward calculation. Thus time-consuming numerical derivatives, i.e. variation of each parameter and recalculation of the disturbed spectrum, become unnecessary. However to optimize run-time, some simplifications are made. A detailed description of the method is given by Ref. [28]. In this Section we compare analytical with numerical derivatives calculated by KOPRA (cf. Ref. [28]).

Figure 22 shows derivatives of a HNO_3-spectrum with respect to volume-mixing-ratio and to temperature at tangent height (20 km). The *vmr* derivatives differ by less than $\pm$2%, which means very good agreement. The T derivatives exhibit an offset of about -8% and a larger scatter. However, in real retrievals T derivatives of HNO_3 will only be necessary for error estimation. The T derivatives of CO_2 are more important, since this trace gas will be used for p,T-retrievals. Figure 23 shows the derivatives at 10 km and 40 km tangent height. The deviations are more smoothly and smaller, -6% and 4% at the wavenumber regions with highest temperature sensitivity. Figure 24 contains the *vmr* derivatives of a H_2O line at 10 km tangent height. Due to saturation a wide region around line center is no longer sensitive on *vmr* variations (cf. Figure 23, top). The deviations at the maxima of the derivatives are about 6%.

7 Conclusions

The KOPRA-RFM intercomparison shows, that there are no significant differences in ray-tracing, integrated column amounts and Curtis-Godson parameters. The difference spectra are significantly less than NESR/4 for the isolated CO_2 line as well as for six MIPAS microwindows and thus KOPRA fulfills the acceptance criteria. In most cases the deviations are even clearly below 1 nW/(cm^2 sr cm^{-1}), i.e. more than one order of magnitude below the acceptance threshold. This is generally valid for unconvolved as well as for ALS and FOV convolved spectra. There is also good agreement in modeling of the H_2O-, O_2- and N_2-continua and of CO_2 line-mixing. Somewhat larger deviations of several nW/(cm^2 sr cm^{-1}) were found for NLTE calculations with "default" vertical resolution (1 km or 2.5 km), caused by different layer-averaging of the vibrational temperatures. The differences could be considerably reduced by using profiles of higher vertical resolution of 250 m. Cross-section spectra agree well, if the tabulated data are given without specification of pressure, e.g. for $ClONO_2$ and N_2O_5, and cover the atmospheric temperatures. Due to different temperature extrapolation the deviations increase up to 10 nW/(cm^2 sr cm^{-1}) for atmospheric temperatures outside the tabulated range. The RFM is not yet adjusted to cross-sections given for non-equidistant temperature and pressure values, like CFC data in the HITRAN96 database. Thus there are differences of 30 nW/(cm^2 sr cm^{-1}) for e.g. CFC-12 for typical stratospheric conditions. However, the deviations decrease below 0.5 nW/(cm^2 sr cm^{-1}) for homogeneous path calculations using p, T of a tabulated dataset. There is sufficiently good agreement between analytical and numerical derivatives.

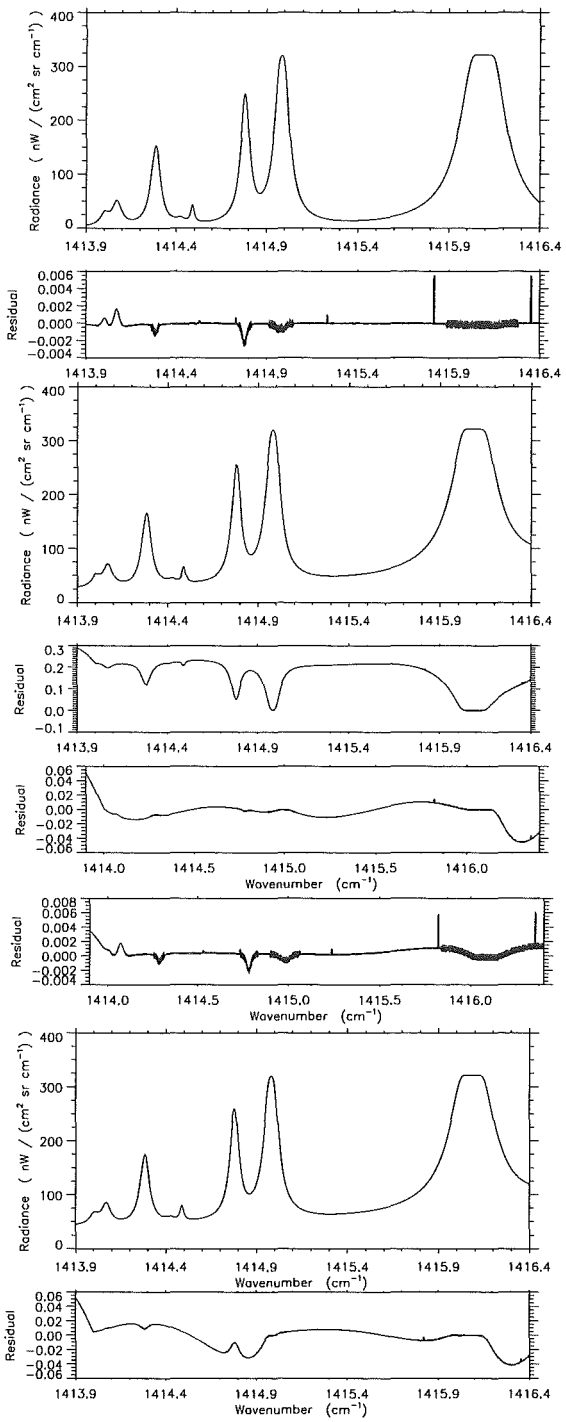


Figure 3: Unapodized H_2O radiance spectra and KOPRA-RFM differences for a homogeneous path (250 hPa, 220 K) and KOPRA accuracy of 10^{-10}. (a) Only lines inside the microwindow considered, (b) additionally lines within $\pm 32 cm^{-1}$ (KOPRA) and $\pm 25 cm^{-1}$ (RFM) from the microwindow's edges considered, (c) for both codes lines within ± 25 cm^{-1} considered, (d) same as (c) but RFM's fine-mesh extended to ± 5 cm^{-1}, (e) same as (c) but additionally H_2O continuum considered.

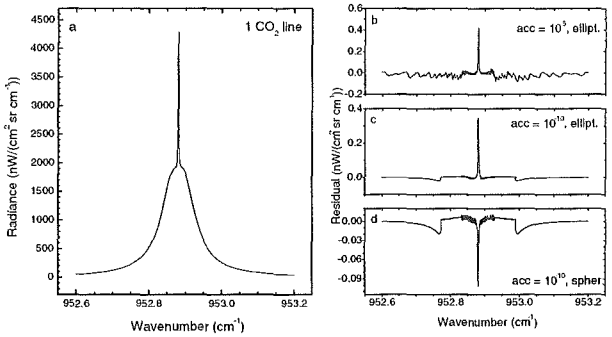


Figure 4: (a) Isolated CO_2 line, unapodized, 10 km tangent height; residuals for KOPRA calculations for ellipsoidal Earth and an accuracy of (b) 10^{-5} and (c) of 10^{-10}, (d) for KOPRA calculations for spherical Earth and an accuracy of 10^{-10}.

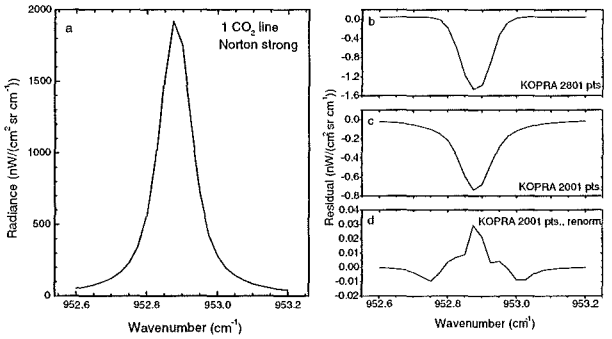


Figure 5: (a) Isolated CO_2 line, convolved with Norton-Beer strong function, 10 km tangent height; residuals for (b) different spectral extensions of the apodization functions (RFM 2001, KOPRA 2801 spectral points), (c) both functions with 2001 points, (d) KOPRA's apodization function additionally renormalized as by the RFM.

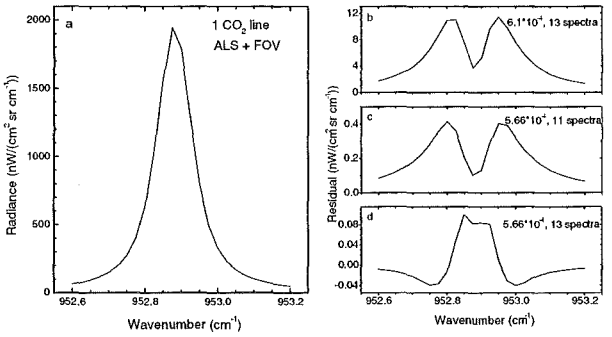


Figure 6: (a) Isolated CO_2 line, ALS and FOV convolved, 10 km tangent height; RFM calculations with 12 point-radiance spectra. Residuals for KOPRA calculations for nominal FOV half-angle of 6.1×10^{-4} rad and 13 spectra (b), for RFM-adjusted half-angle of FOV extension and 11 spectra (c), for RFM-adjusted half-angle of FOV extension and 13 spectra (d).

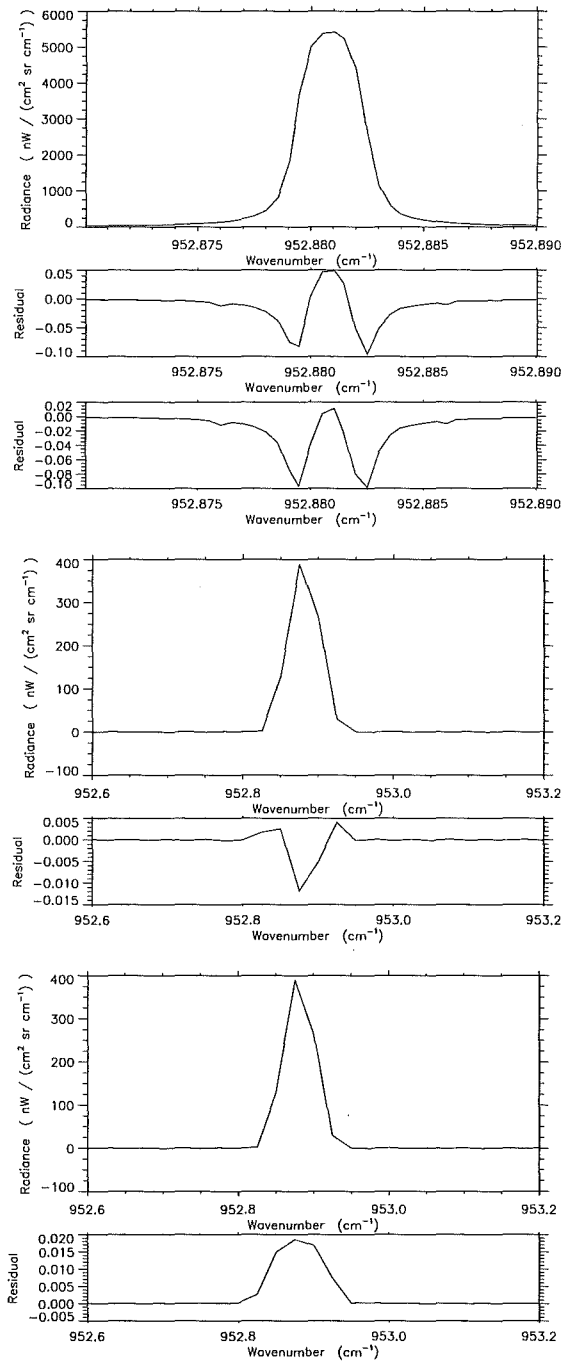


Figure 7: Isolated CO_2 line, 40 km tangent height, KOPRA's accuracy 10^{-10}; top: unconvolved spectrum, differences for KOPRA calculations for ellipsoidal Earth and spherical Earth; middle: ALS convolution (ils2001.nor); bottom: ALS and FOV convolution, FOV-modeling with 12 (RFM) and 13 (KOPRA) point-radiance spectra, KOPRA's FOV RFM-adjusted.

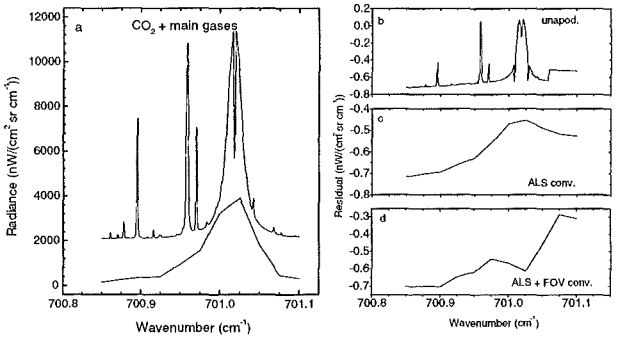


Figure 8: (a) p,T-microwindow, 40 km tangent height, unapodized (offset 2000 nW/(cm^2 sr cm^{-1})) and ALS+FOV convolved spectra; residuals for (b) unapodized spectra, (c) ALS convolved spectra, (d) ALS and FOV convolved spectra, KOPRA calculations for RFM-adjusted FOV extension.

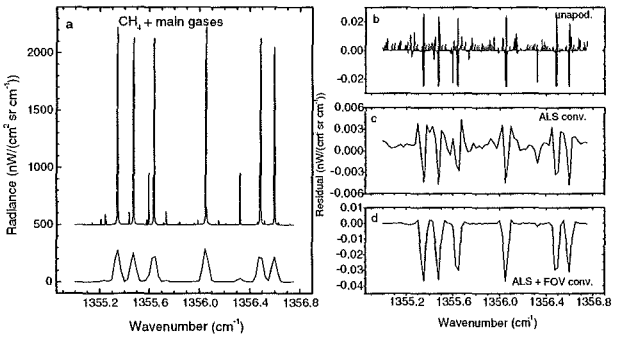


Figure 9: Same as Figure 8, but for a CH_4-microwindow, 40 km tangent height, unapodized spectra shifted by 500 nW/(cm^2 sr cm^{-1}).

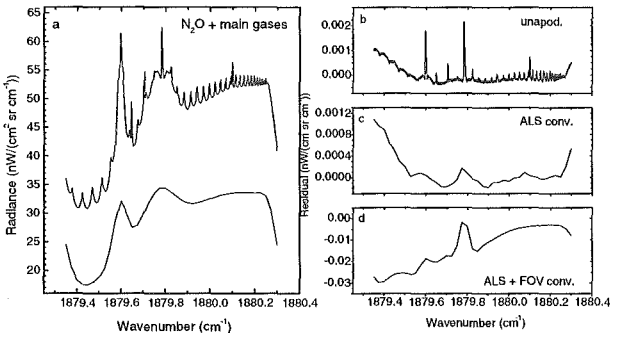


Figure 10: Same as Figure 8, but for a N_2O-microwindow, 10 km tangent height, unapodized spectra shifted by 20 nW/(cm^2 sr cm^{-1}).

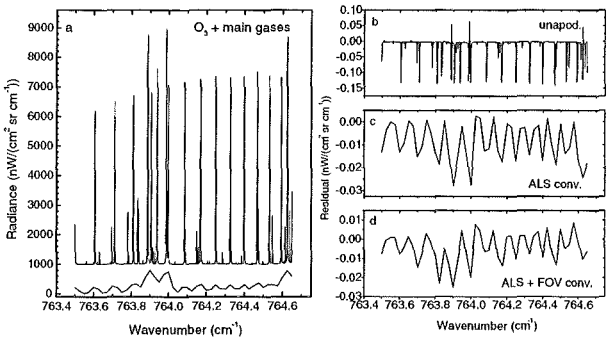


Figure 11: Same as Figure 8, but for an O_3-microwindow, 40 km tangent height, unapodized spectra shifted by 1000 nW/(cm^2 sr cm^{-1}).

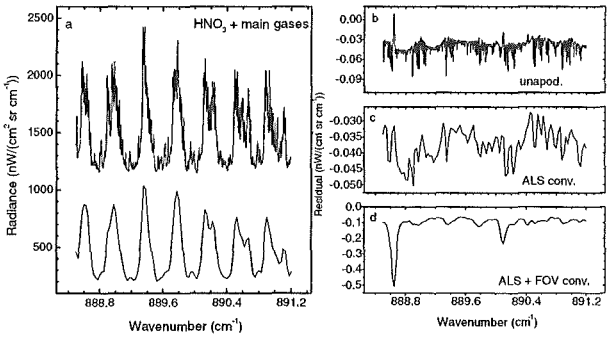


Figure 12: Same as Figure 8, but for a HNO_3-microwindow, 10 km tangent height, unapodized spectra shifted by 1000 nW/(cm^2 sr cm^{-1}).

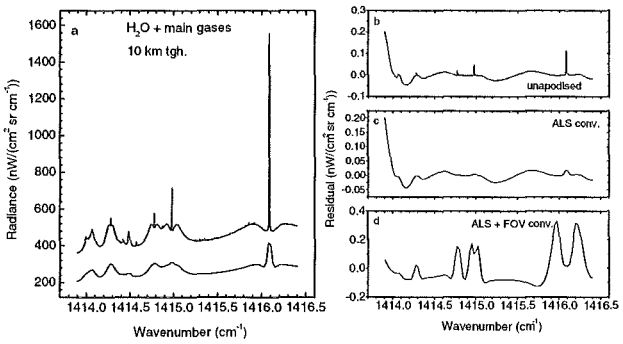


Figure 13: Same as Figure 8, but for a H_2O-microwindow, 10 km tangent height, unapodized spectra shifted by 200 nW/(cm^2 sr cm^{-1}).

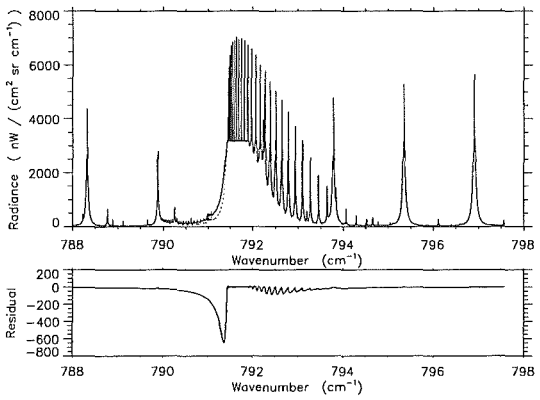


Figure 14: Unapodized KOPRA spectra containing the CO_2-Q-branch at 791.6 cm^{-1}, 10 km tangent height; differences between KOPRA calculations neglecting line-mixing and including line-mixing (Rosenkranz approximation and Q-branch coupling).

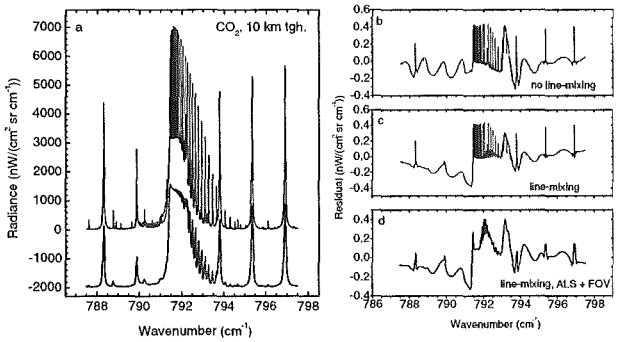


Figure 15: (a) CO_2-microwindow containing the Q-branch at 791.6 cm^{-1}, unapodized and ALS+FOV convolved spectra (shifted by -2000nW/(cm^2 sr cm^{-1}). Residuals for (b) unapodized spectra, no line-mixing, (c) unapodized spectra, line-mixing with Rosenkranz approximation, (d) ALS and FOV convolved spectra, line-mixing with Rosenkranz approximation.

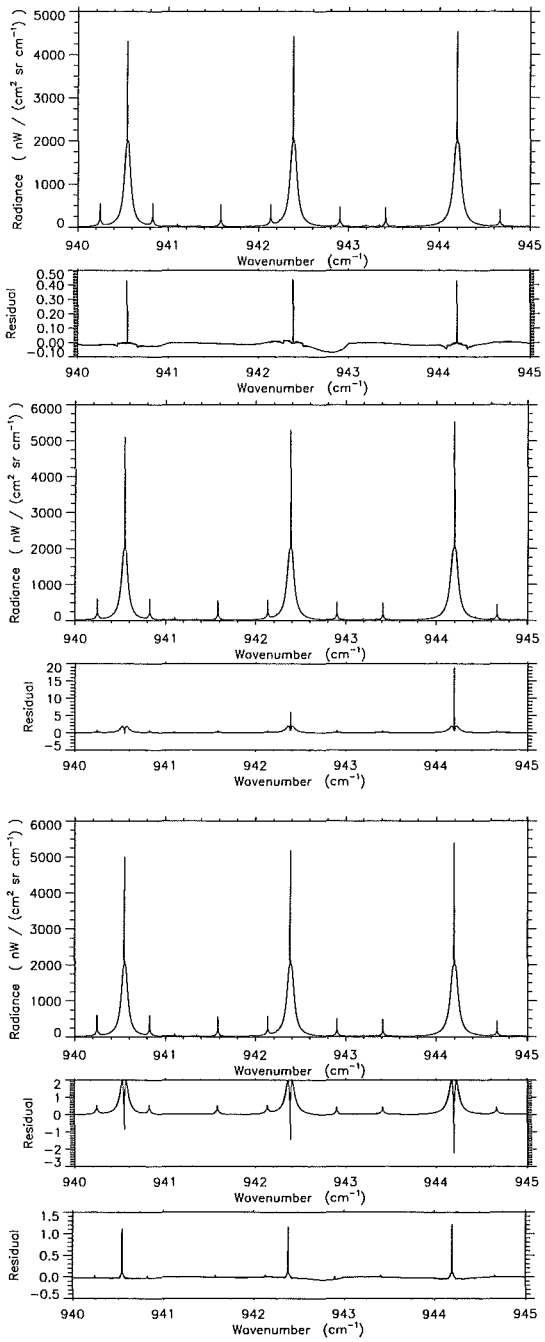


Figure 16: CO_2-microwindow, 10 km tangent height, unapodized spectra and residuals for (a) LTE-conditions (as reference) and atmospheric profiles with 1 km vertical resolution up to 120 km, (b) NLTE-calculations for 1 km spacing up to 120 km (also T_{vib}), (c) NLTE-calculations for 1 km spacing up to 80 km and (d) NLTE-calculations for 250 m spacing up to 80 km (only residuals shown).

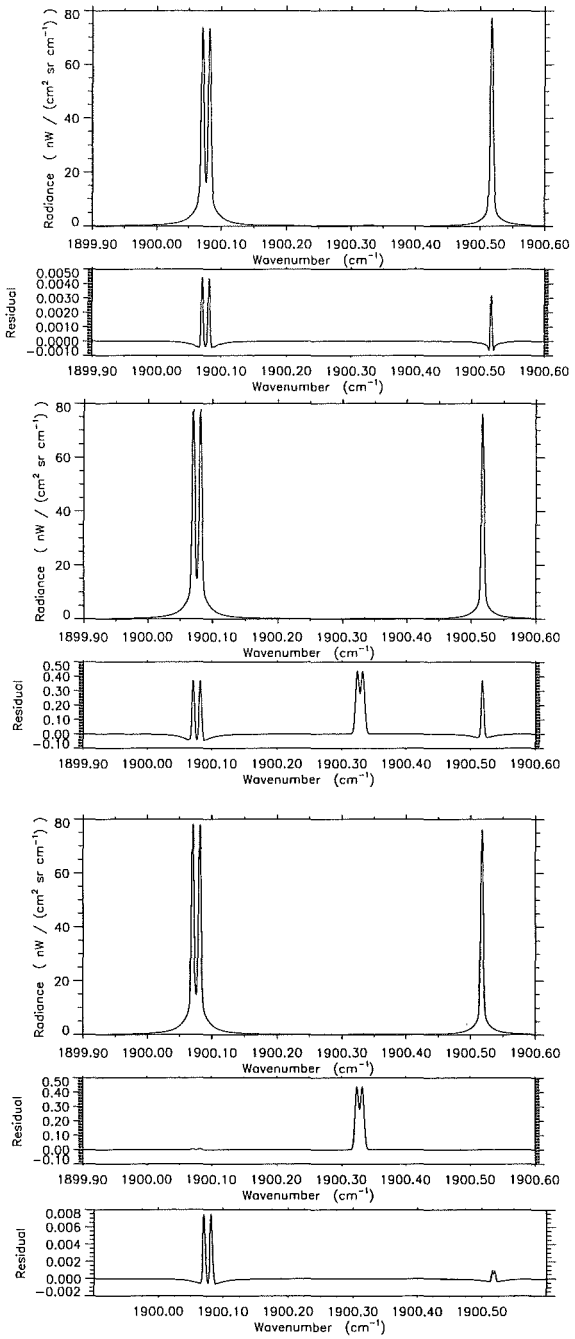


Figure 17: NO-microwindow, 10 km tangent height, unapodized spectra and residuals for (a) LTE-conditions and atmospheric profiles with 2.5 km spacing up to 200 km (as reference), (b) NLTE-calculations for 2.5 km spacing up to 200 km (also T_{vib}), (c) NLTE-calculations for 250 m spacing up to 200 km and (d) same as in (c), but lines at 1900.32 cm^{-1} also taken into account by RFM (only residual shown).

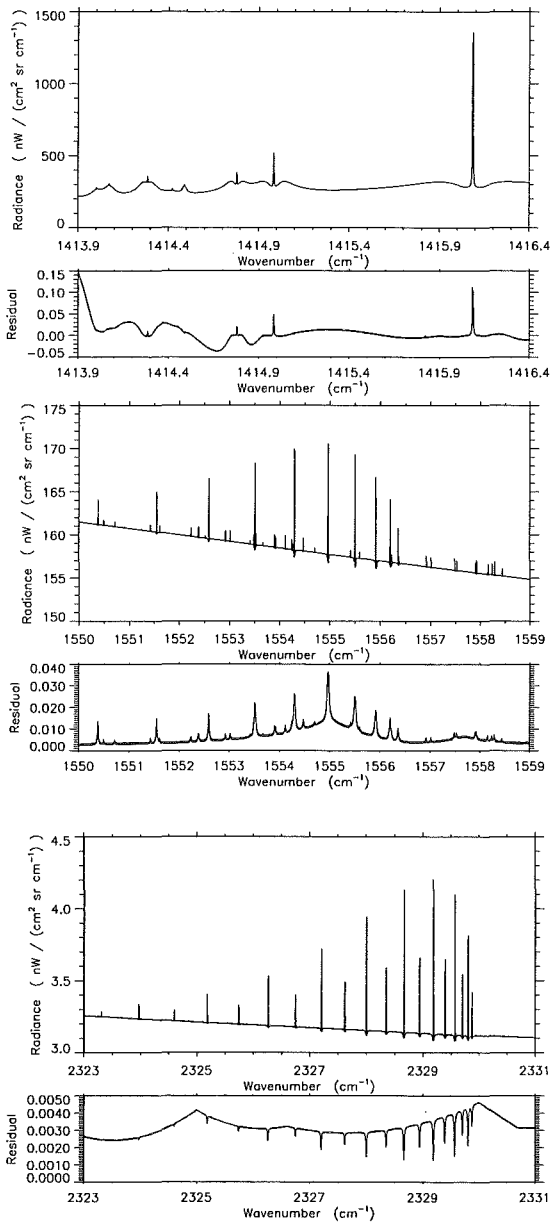


Figure 18: Unapodized spectra for 10 km tangent height, top: H_2O-microwindow including the H_2O-continuum, middle: O_2-microwindow with O_2-continuum, bottom: N_2-microwindow with N_2-continuum.

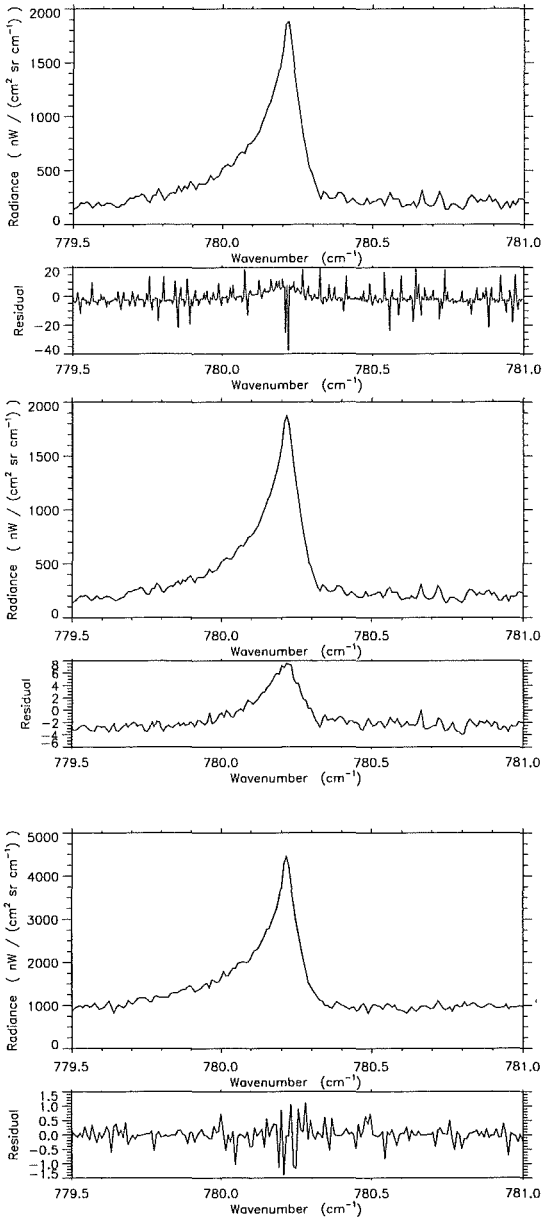


Figure 19: Unapodized $ClONO_2$ cross-section spectra for 10 km tangent height and KOPRA-RFM deviations. Top: different wavenumber grids (KOPRA 0.008 cm^{-1}, RFM 0.0005 cm^{-1}), middle: both codes same wavenumber grids (0.008 cm^{-1}), bottom: both codes same wavenumber grids and isothermal atmosphere (296 K).

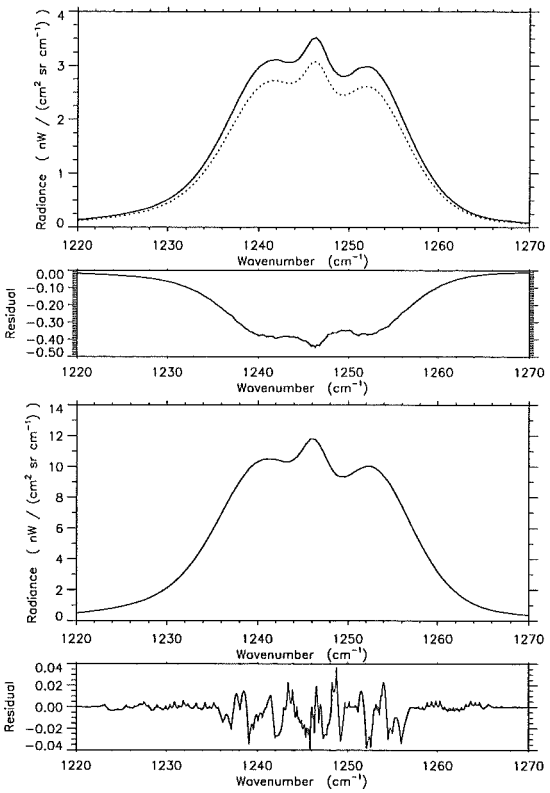


Figure 20: Unapodized N_2O_5 cross-section spectra for 10 km tangent height and KOPRA-RFM deviations for (top) 1976 US Standard Atmosphere and (bottom) isothermal atmosphere (273 K).

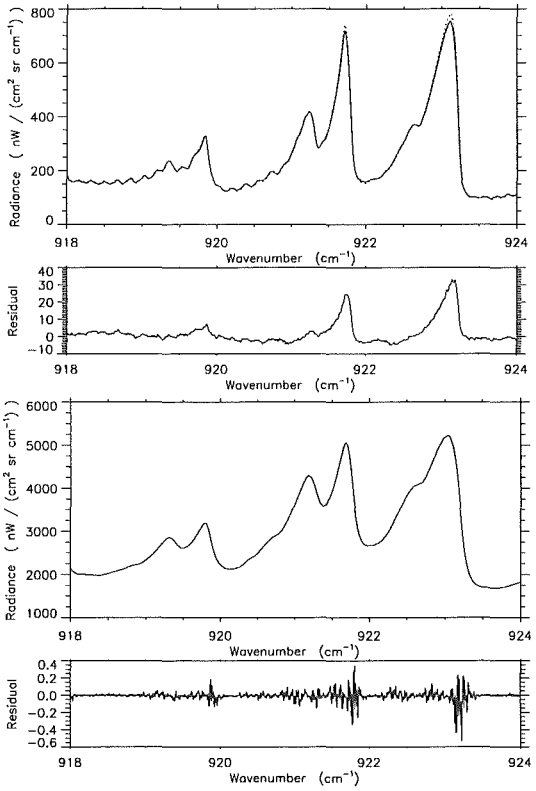


Figure 21: Unapodized CFC-12 (CCl_2F_2) cross-section spectra for homogeneous path conditions (250 km). Top: KOPRA and RFM calculations for 250 hPa and 220 K, bottom: calculations for 1054 hPa and 296.2 K.

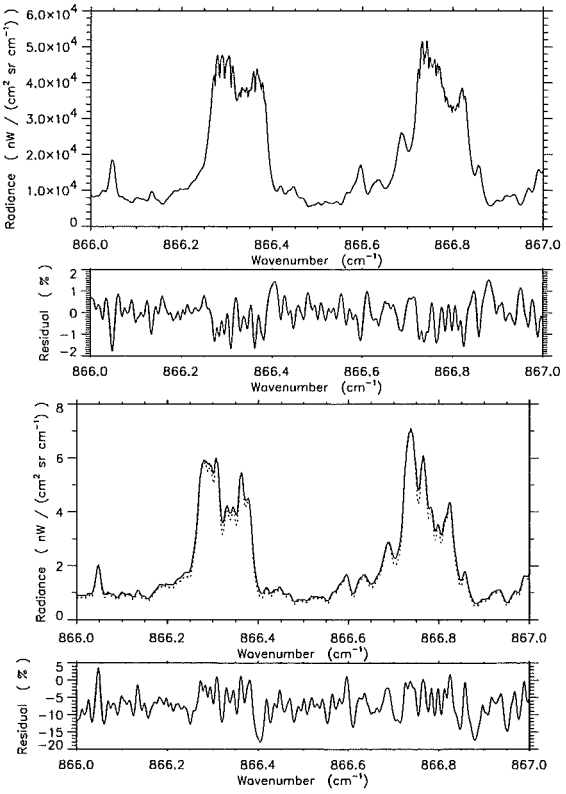


Figure 22: Analytical and numerical derivatives of an unapodized HNO_3-spectrum with respect to volume-mixing ratio (top) and temperature (bottom) at tangent height (20 km) and differences (numerical - analytical).

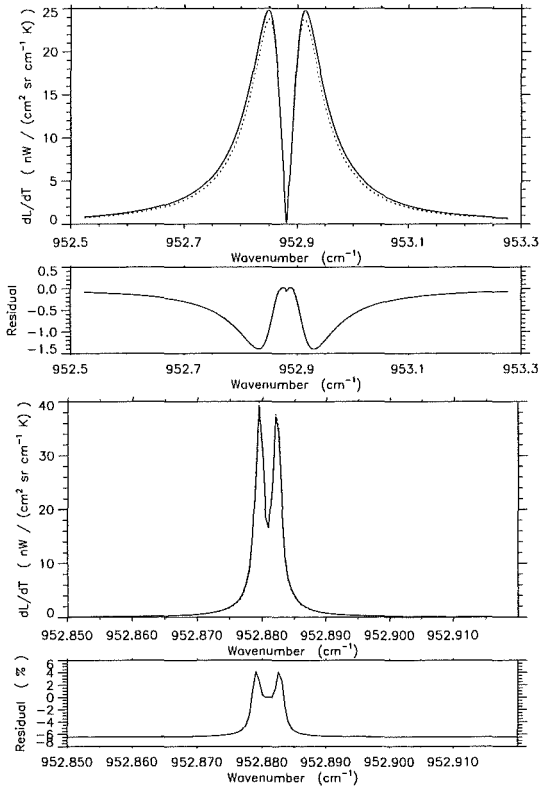


Figure 23: Analytical and numerical derivatives of unapodized CO_2-spectra with respect to temperature at tangent height of 10 km (top) and 40 km (bottom) and differences (numerical - analytical).

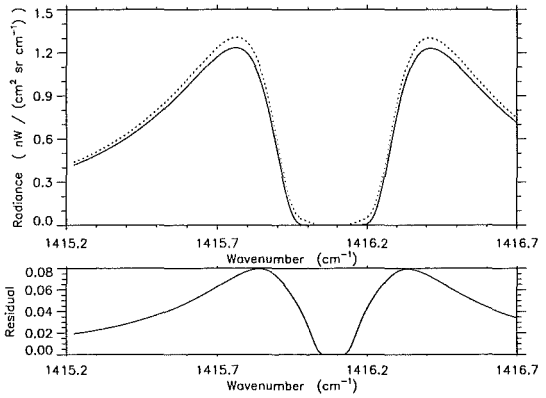


Figure 24: Analytical and numerical derivative of an unapodized H_2O-spectrum with respect to volume-mixing ratio at tangent height (10 km) and differences (numerical - analytical).

Appendix A

RFM- and KOPRA-Modifications to minimize CO_2-Residuals

In Figure 25 we show unapodized CO_2-lines for 10 km tangent height and summarize the reductions of KOPRA-RFM residuals, which can be obtained by slight modifications of the two codes. The first residual spectrum results from calculations using the standard versions of the codes and contains several spikes, oscillations and discontinuities. Although these features are very small compared to NESR/4 (12.5 nW/(cm^2 sr cm^{-1})), we tried to find the reason of the differences. The oscillations at e.g. 940.8 and 942.8 cm^{-1} are due to the wide-mesh interpolation of the RFM and can be removed by extending the RFM's fine-mesh area to ± 5 cm^{-1} (second residual spectrum). The spikes are for the most part caused by the different Earth shapes assumed (KOPRA: ellipsoidal Earth, RFM: spherical Earth) and are reduced from 0.35 to -0.1 nW/(cm^2 sr cm^{-1}) by a modified KOPRA version calculating with a spherical Earth shape (third residual). The discontinuities close to the spikes result from different Humlicek-algorithm realizations. Whereas KOPRA uses a fast version for calculations including χ-factor, the RFM takes the standard version. For simulations without χ-factor the RFM also applies the fast version and the discontinuities nearly disappear (fourth residual). The remaining small positive spikes are due to slightly different extensions of the evaluation regions of the fast Humlicek-algorithm.

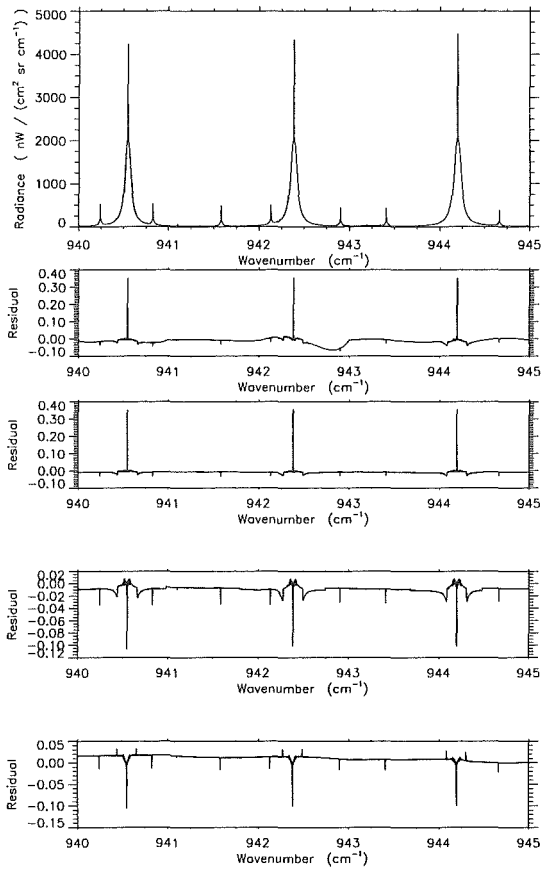


Figure 25: Unapodized CO_2-spectra, 10 km tangent height. Residuals for calculations (a) with χ-factor, standard program versions, (b) with χ-factor, RFM's fine-mesh extended to ± 5 cm^{-1}, (c) with χ-factor, RFM's fine-mesh extended to ± 5 cm^{-1}, KOPRA calculations for spherical Earth, (d) same as (c), but without χ-factor.

Appendix B

FOV-Modeling by KOPRA

Figure 26 contains ALS and FOV convolved H_2O-spectra calculated by KOPRA for 10 km tangent height. The residuals result from FOV-modeling by different numbers of point-radiance spectra. The upper plot contains the reference spectrum modeled by 39 and a spectrum modeled by just 3 point-radiance spectra. The respective difference spectrum (second plot) exhibits systematic deviations of up to -8 nW/(cm^2 sr cm^{-1}) at the line wings. The additional plots show the deviations between the reference case and spectra modeled by 9, 13 and 30 point-radiance spectra. The residuals successively decrease to 0.02 nW/(cm^2 sr cm^{-1}) in the lowermost plot. For FOV-modeling by 13 spectra, as performed in the KOPRA-RFM intercomparison, the differences amount to $\pm$0.4 nW/(cm^2 sr cm^{-1}).

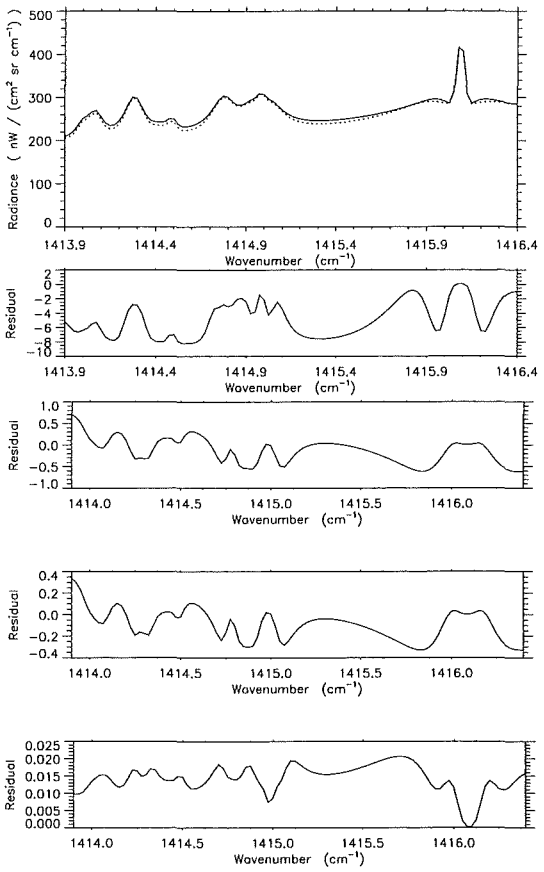


Figure 26: ALS and FOV convolved H_2O-spectra calculated by KOPRA for 10 km tangent height. A reference spectrum (—) was created using 39 point-radiance spectra. The second spectrum (···) and the first residual result from FOV-modeling by 3, the additional residuals from FOV-modeling by 9, 13 and 30 point-radiance spectra.

Bibliography

[1] A. Dudhia and P. Raspollini, "Results of intercomparisons of the OFM and the RFM code PO-TN-OXF-GS-0007," tech. rep., Oxford University and IROE, December, 9, 1996. Combination of Task Reports on Task 2.1 (PO-SW-ESA-GS-0315) and Task 3.3 (PO-SW-ESA-GS-00323) of ESA Contract 12055/96/NL/CN.

[2] G. P. Stiller, M. Höpfner, M. Kuntz, T. von Clarmann, G. Echle, H. Fischer, B. Funke, N. Glatthor, F. Hase, and S. Zorn, "The Karlsruhe Optimized and Precise Radiative transfer Algorithm (KOPRA): Realization, model error assessment and a posteriori justification," in *Proc. European Symposium on Atmospheric Measurements from Space, ESAMS'99, 18–22 Jan 1999, Noordwijk*, pp. 749–756, European Space Agency, ESTEC, Noordwijk, The Netherlands, 1999.

[3] P. Morris, A. Darbyshire, and A. Dudhia, "MIPAS–development of a reference forward algorithm for the simulation of MIPAS atmospheric limb emission spectra – detailed design document," tech. rep., European Space Agency, 1997. Report of ESA Contract 11886/96/NL/GS.

[4] N. Glatthor, M. Höpfner, G. P. Stiller, T. von Clarmann, A. Dudhia, G. Echle, B. Funke, and F. Hase, "Intercomparison of the KOPRA and the RFM radiative transfer codes," in *Proc. European Symposium on Atmospheric Measurements from Space, ESAMS'99, 18–22 Jan 1999, Noordwijk*, pp. 757–764, European Space Agency, ESTEC, Noordwijk, The Netherlands, 1999.

[5] G. Echle, H. Oelhaf, and A. Wegner, "Measurement of atmospheric parameters with MIPAS," tech. rep., European Space Agency, December 1992. Final Report of ESA Contract 9597/91/NL/SF.

[6] L. S. Rothman, C. P. Rinsland, A. Goldman, T. Massie, D. P. Edwards, J.-M. Flaud, A. Perrin, C. Camy-Peyret, V. Dana, J.-Y. Mandin, J. Schroeder, A. McCann, R. R. Gamache, R. B. Wattson, K. Yoshino, K. V. Chance, K. W. Jucks, L. R. Brown, V. Nemtchinov, and P. Varanasi, "The HITRAN molecular spectroscopic database and HAWKS (HITRAN atmospheric workstation): 1996 edition," *J. Quant. Spectrosc. Radiat. Transfer* **60**(5), pp. 665–710, 1998.

[7] H. Norton and R. Beer, "New apodizing functions for Fourier spectrometry," *J. Opt. Soc. Am* **66**(3), pp. 259–264, 1976. (Errata J. Opt. Soc. Am., 67, 419,1977).

[8] F. Hase and M. Höpfner, "Atmospheric ray path modeling for radiative transfer algorithms," *Appl. Opt.* **38**(15), pp. 3129–3133, 1999.

[9] J. Humlicek, "Optimized computation of the Voigt and complex probability functions," *J. Quant. Spectrosc. Radiat. Transfer* **27**(4), pp. 437–444, 1982.

[10] M. Kuntz, "A new implementation of the Humlicek algorithm for the calculation of the Voigt profile function," *J. Quant. Spectrosc. Radiat. Transfer* **57**(6), pp. 819–824, 1997.

[11] R. J. Wells, "Rapid approximation to the Voigt/Faddeeva function and its derivatives," *J. Quant. Spectrosc. Radiat. Transfer* **62**(1), pp. 29–48, 1999.

[12] M. Kuntz and M. Höpfner, "Efficient line–by–line calculation of absorption coefficients," *J. Quant. Spectrosc. Radiat. Transfer* **63**(1), pp. 97–114, 1999.

[13] L. L. Strow and D. Reuter, "Effect of line mixing on atmospheric brightness temperatures near 15 μm," *Appl. Opt.* **27**(5), pp. 872–878, 1988.

[14] L. L. Strow, D. C. Tobin, and S. E. Hannon, "A compilation of first–order line–mixing coefficients for CO_2 Q–branches," *J. Quant. Spectrosc. Radiat. Transfer* **52**(3/4), pp. 281–294, 1994.

[15] P. W. Rosenkranz, "Shape of the 5 mm oxygen band in the atmosphere," *IEEE Transactions on Antennas and Propagation* **AP–23**(4), pp. 498–506, 1975.

[16] D. P. Edwards, M. López-Puertas, and M. Á. López-Valverde, "Non–local thermodynamic equilibrium studies of the 15 μm bands of CO_2 for atmospheric remote sensing," *J. Geophys. Res.* **98**(D8), pp. 14,955–14,977, 1993.

[17] D. P. Edwards, M. López-Puertas, and M. G. Mlynczak, "Non–local thermodynamic equilibrium limb radiance from O_3 and CO_2 in the 9–11 μm spectral region," *J. Quant. Spectrosc. Radiat. Transfer* **52**(3–4), pp. 389–407, 1994.

[18] D. P. Edwards, M. López-Puertas, and R. R. Gamache, "The non–LTE correction to the vibrational component of the internal partition sum for atmospheric calculation," *J. Quant. Spectrosc. Radiat. Transfer* **59**(3–5), pp. 423–436, 1998.

[19] M. López-Puertas, G. Zaragoza, M. Á. López-Valverde, F. J. Martín-Torres, G. M. Shved, R. O. Manuilowa, A. A. Kutepov, O. Gusev, T. von Clarmann, A. Linden, G. Stiller, A. Wegner, H. Oelhaf, D. P. Edwards, and J.-M. Flaud, "Non–local thermodynamic equilibrium limb radiances for the MIPAS instrument on ENVISAT–1," *J. Quant. Spectrosc. Radiat. Transfer* **59**(3–5), pp. 377–403, 1998.

[20] S. A. Clough, F. X. Kneizys, and R. W. Davies, "Line shape and the water vapor continuum," *Atmospheric Research* **23**, pp. 229–241, 1989.

[21] F. Thibault, V. Menoux, R. Le Doucen, L. Rosenmann, J.-M. Hartmann, and C. Boulet, "Infrared collision–induced absorption by O_2 near 6.4 μm for atmospheric applications: measurements and empirical modeling," *Appl. Opt.* **36**(3), pp. 563–567, 1997.

[22] W. J. Lafferty, A. M. Solodov, A. Weber, W. B. Olson, and J.-M. Hartmann, "Infrared collision–induced absorption by N_2 near 4.3 μm for atmospheric applications: measurements and empirical modeling," *Appl. Opt.* **35**(30), pp. 5911–5917, 1996.

[23] J. Ballard, W. B. Johnston, M. R. Gunson, and P. T. Wassell, "Absolute absorption coefficients of $ClONO_2$ infrared bands at stratospheric temperatures," *J. Geophys. Res.* **93**(D2), pp. 1659–1665, 1988.

[24] C. A. Cantrell, J. A. Davidson, A. H. McDaniel, R. E. Shetter, and J. G. Calvert, "Infrared absorption cross sections for N_2O_5," *Chem. Phys. Lett.* **148**, pp. 358–363, 1988.

[25] T. von Clarmann, H. Fischer, F. Friedl-Vallon, A. Linden, H. Oelhaf, C. Piesch, M. Seefeldner, and W. Völker, "Retrieval of stratospheric O_3, HNO_3 and $ClONO_2$ profiles from 1992 MIPAS–B limb emission spectra: Method, results and error analysis," *J. Geophys. Res.* **98**(D11), pp. 20,495–20,506, 1993.

[26] P. Varanasi and V. Nemtchinov, "Thermal infrared absorption coefficients of CFC–12 at atmospheric conditions," *J. Quant. Spectrosc. Radiat. Transfer* **51**(5), pp. 679–687, 1994.

[27] S. T. Massie, A. Goldman, A. H. McDaniel, C. A. Cantrell, J. A. Davidson, R. E. Shetter, and J. G. Calvert, "Temperature dependent infrared cross sections for CFC–11, CFC–12, CFC–13, CFC–14, CFC–22, CFC–113, CFC–114, and CFC–115," tech. rep., NCAR, 1991. Technical Note, TN–358+STR.

[28] M. Höpfner, G. P. Stiller, M. Kuntz, T. von Clarmann, G. Echle, B. Funke, N. Glatthor, F. Hase, H. Kemnitzer, and S. Zorn, "The Karlsruhe optimized and precise radiative transfer algorithm. Part II: Interface to retrieval applications," in *Optical Remote Sensing of the Atmosphere and Clouds, Beijing, China, 15–17 September 1998*, J. Wang, B. Wu, T. Ogawa, and Z. Guan, eds., *Proc. SPIE* **3501**, pp. 186–195, 1998.

Overall retrieval error budget and a posteriori justification of modeling choices

G.P. Stiller

Abstract: In this paper we present the assessment of individual forward model errors by disregarding a number of atmospheric properties and processes KOPRA is able to handle with, as well as an over-all error budget with respect to these parameters. We demonstrate that a number of specific features of KOPRA are capable to improve the retrieval accuracy in MIPAS/ENVISAT data analysis compared to approaches which neglect the modeling of these atmospheric and/or instrumental properties.

1 Introduction

KOPRA has been designed to allow modeling of a number of atmospheric properties and processes relevant for the altitude range the MIPAS mission will cover, as well as the instrument specific details of MIPAS itself. Although we decided to cover all these aspects, we are aware that some of them may be disregarded in one or the other situation of data analysis without too serious loss of accuracy. However, in order not to rely on *ad hoc* decisions and introduce retrieval errors due to simplified assumptions, a study has been performed to quantify the relevance of these effects. We have assessed the forward modeling error and its mapping on the retrieval error for a number of KOPRA specific features like modeling of the Earth's oblateness, temperature and vmr horizontal inhomogeneities, neglection of NLTE effects and line-mixing, insufficient modeling of MIPAS' field of view and instrumental line shape, and others. Based on this study, *a posteriori* decisions on the relevance of these effects and thus on the possibility to simplify the retrieval cases, have become possible.

2 Modeling error related to raytracing in a non-spherical atmosphere

In contrast to most other radiative transfer models, KOPRA makes use of the approach of differential displacement of the ray path[1] which assumes linear variation of the refractive index within each incremental step of integration. Contrary to the implementation of Snell's Law for spherical media, this approach can easily be extended for application to an ellipsoidal earth shape. For our implementation, the WGS84 reference ellipsoid has been used. It was shown that the spherical approximation of the earth, making use of the local curvature of the WGS84 reference ellipsoid, is sufficient as long as measurement geometry is characterized by the observer altitude and the tangent altitude (see Part IV: 'Atmospheric raypath modeling for radiative transfer algorithms' and Table 1). However, earth shape considerations may be of major importance if the measurement geometry is characterized by the observer altitude and the elevation angle of the measurement. This is in particular true if the tangent point is a fit parameter rather than a constant, and the latitude of the tangent point is changing from iteration to iteration of a pointing retrieval. This is illustrated by Table 2.

Table 1: Path lengths and CO_2 column amounts as calculated by KOPRA using the WGS84 reference ellipsoid, for spherical earth, and assuming local curvature radius of the WGS84 reference ellipsoid. Calculations were made for various observer positions and viewing directions (azimuth = 0^o means pointing southward), and for 8 km tangent height. The tangent height is assumed to be known.

	Latitude[^{o}]/ Azimuth[^{o}]	0/0	0/90	45/0	90/0
Path length [km]	WGS 84	2446.6	2455.0	2452.8	2459.1
	Spherical	2452.8	2452.8	2452.8	2452.8
	Spherical, curvature radius of WGS 84 at tangent point	2446.4	2455.0	2452.8	2459.2
$CO_2[10^{23}$molec/cm$^2]$	WGS 84	2.3051	2.3135	2.3113	2.3177
	Spherical	2.3114	2.3114	2.3114	2.3114
	Spherical, curvature radius of WGS 84 at tangent point	2.3051	2.3135	2.3114	2.3178

We conclude from this comparison that for the application as planned, namely the retrieval of elevation angles from measured spectra for various geolocations, and the modeling of the earth's ellipticity is mandatory.

Table 2: Path lengths, CO_2 column amounts, and tangent heights as calculated by KOPRA using the WGS84 reference ellipsoid, and for spherical earth. Calculations were made for various observer positions and viewing directions (azimuth = 0^o means pointing southward). Observer altitude is 800 km, nadir angle is 62.7675^o

	Latitude[^{o}]/ Azimuth[^{o}]	0/0	0/90	45/0	90/0
Path length [km]	WGS 84	2449.3	2541.6	2487.7	2424.0
	Spherical	2514.6	2514.6	2514.6	2514.6
CO_2 [10^{23}molec/cm^2]	WGS 84	2.3101	4.3490	3.1580	4.3020
	Spherical	3.6880	3.6880	3.6880	3.6880
Tangent height [km]	WGS 84	7.992	2.914	5.537	0.922
	Spherical	4.290	4.290	4.290	4.290

3 Mapping of radiative transfer modeling errors on the retrieval error of trace species

3.1 Method

In the following we present the estimation of the mapping of radiative transfer modeling errors on the retrieval of target species, the comparison to the random retrieval error due to measurement noise, and the assessment of the expected overall retrieval error which results from these individual uncertainties. For estimation of retrieval errors we follow the scheme proposed by Ref. [2] who apply linear theory which usually is considered sufficient for error propagation problems in this context[3]. The mapping of measurement noise, represented by the measurement covariance matrix $\mathbf{S}_y$, on the retrieval, represented by the retrieval covariance matrix $\mathbf{S}_x$ is estimated as

$$\mathbf{S}_x = (\mathbf{K}^\tau\mathbf{K})^{-1}\mathbf{K}^\tau\mathbf{S}_y\mathbf{K}(\mathbf{K}^\tau\mathbf{K})^{-1} \tag{1}$$

where $\mathbf{K}$ is the Jacobian matrix, evaluated as reported by Ref. [4]. The Jacobian matrix contains the derivatives of the spectral signal with respect to the fit parameters. τ denotes a transposed matrix.

Model errors and parameter errors are characterized by correlations in the spectral domain. The retrieval errors $\Delta x_{i,j}$ of the fit parameter x_i due to error source j are approximated linearily

$$\Delta X_j = \begin{pmatrix} \Delta x_{1,j} \\ \Delta x_{2,j} \\ \vdots \\ \Delta x_{i_{max},j} \end{pmatrix} = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T(Y_{error,j} - Y_{true}). \quad (2)$$

Y_{true} is a synthetic spectrum modeled with KOPRA including all available physics, while $Y_{error,j}$ is a spectrum calculated under neglection of physical effect j.

The total retrieval error Δx_i of the target quantity x_i then is calculated as

$$\Delta x_i = \sqrt{\sigma_i^2 + (\sum_{l=1}^{l_{max}} \Delta x_{i,l})^2 + \sum_{j=1}^{j_{max}} (\Delta x_{i,j})^2} \quad (3)$$

where σ_i^2 is the i^{th} diagonal element of $\mathbf{S}_x$. Errors labeled l are assumed correlated among each other and therefore are added linearily while errors labeled j are assumed uncorrelated among each other and therefore are added quadratically.

3.2 Case Studies

In order to demonstrate the impact of disregarding atmospheric and radiative properties on the spectrum and the related retrieval error we performed a number of case studies. The spectral ranges selected for the case studies are somewhat arbitrary, as they do not all represent optimized so-called microwindows for the analysis of MIPAS data. They were selected as they appeared to be suitable to demonstrate the effects under investigation. Dedicated studies for selection of optimized microwindows for the MIPAS experiments are underway and are beyond the scope of this study[5]. In all cases we generated a "perfect" reference spectrum including all effects considered to be relevant for the correct modeling of the radiance (Y_{true} in Eq. 2). The Jacobians were provided simultaneously with the forward calculation as described in Ref [4]. Besides the target species, an empirical background continuum radiation was an additional fit parameter. A number of less perfect "approximation" spectra were generated, each of them excluding one of the effects under investigation ($Y_{error,j}$ in Eq. 2).

Effects under investigation were line mixing, NLTE, horizontal inhomogeneities of temperature, pressure, and volume mixing ratio (vmr) profiles of target and interfering species, isotopic abundance profiles deviating from the altitude-constant values used in the HITRAN compilation, and the MIPAS specific FOV and ILS modeling. The apodized noise levels assumed were 3.035×10^{-8} W/(cm^2 sr cm^{-1}) and 2.549×10^{-9} W/(cm^2 sr cm^{-1}) for the filter ranges A (685-970 cm^{-1}) and D (1820-2410 cm^{-1}), respectively, according to predictions as provided by ESA[6].

The resulting individual errors and the total error of the retrieved target quantity then were estimated by application of the method outlined above. For the evaluation of the total error, errors due to neglection of NLTE and line-mixing and improper FOV and ILS modeling have been considered correlated, and therefore added linearily, while errors due to horizontal inhomogeneities and isotopic abundances have been considered uncorrelated.

All reference spectra were modeled for the 1976 US Standard atmosphere[7], and trace gas abundance vertical profiles were provided by Echle et al.[8]. The vibrational temperature profiles were from Kerridge and coworkers for O_3, and López-Puertas and coworkers (H_2O, CO_2, and N_2O), both as reported in Ref. [5]. For NO, vibrational temperature profiles from Ref. [9] were used. Profiles of volume

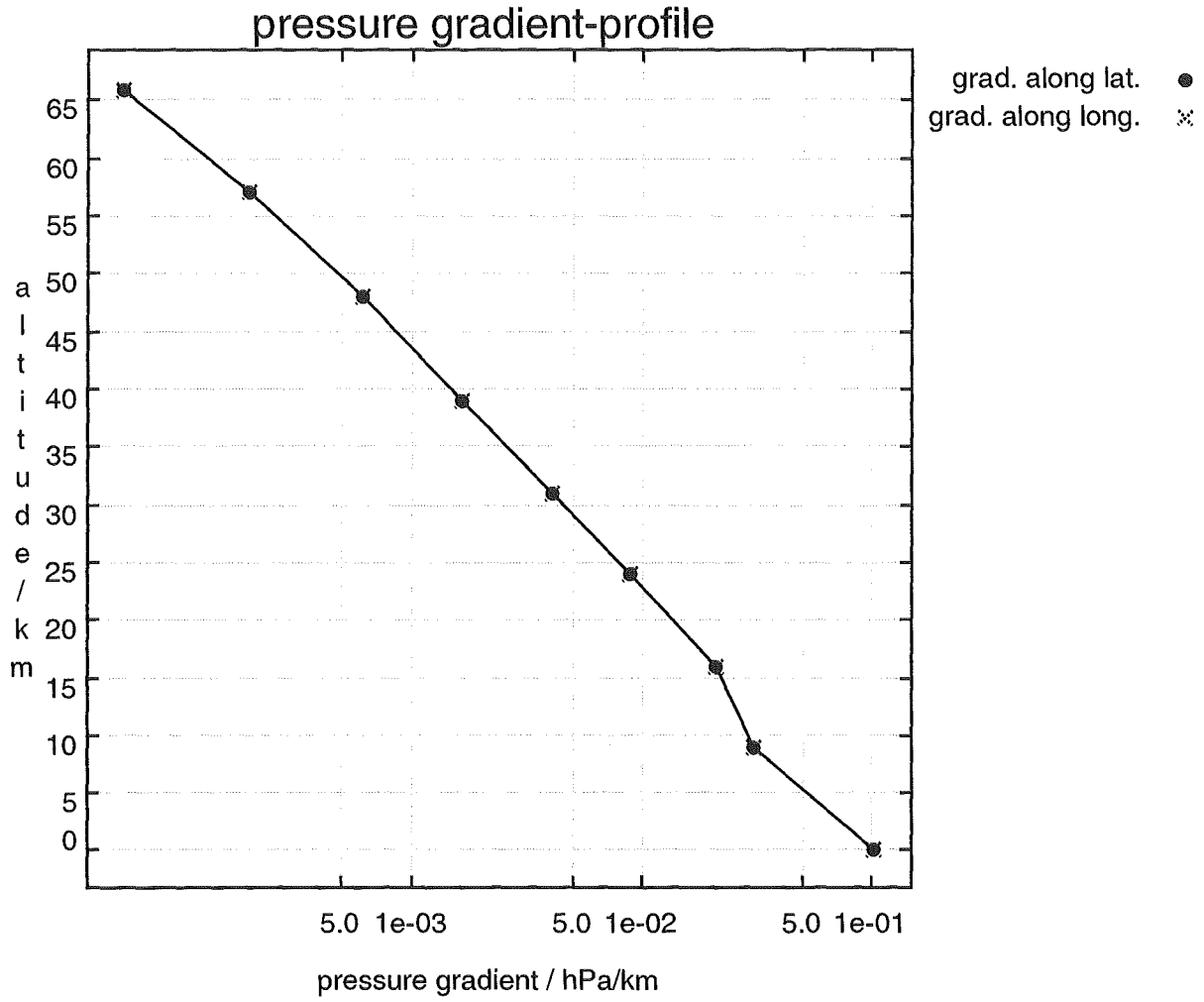


Figure 1: Horizontal inhomogeneities for pressure

mixing ratio gradients were used as estimates of maximum values to be measured by MIPAS[10].

We considered horizontal inhomogeneities for pressure (see Fig. 1), temperature (3 K/ 100 km for all altitudes), and H_2O, O_3, HNO_3, $ClONO_2$, and NO_2 (see Figs. 2 to 6).

Dedicated vertical profiles for isotopomeric abundances were compiled from the literature for water vapor[11] and ozone[12].

For the approximation spectra, line-mixing, NLTE, and horizontal inhomogeneities were disregarded. Instead of the isotopomeric profiles above, the altitude-constant HITRAN standard values were used. The ILS was modeled for a circular FOV and assuming perfect performance of the instrument (no modulation loss, no phase error). Instead of integration over the FOV, only radiance along the center beam was modeled.

Results of the retrieval error assessment are presented in Table 3.

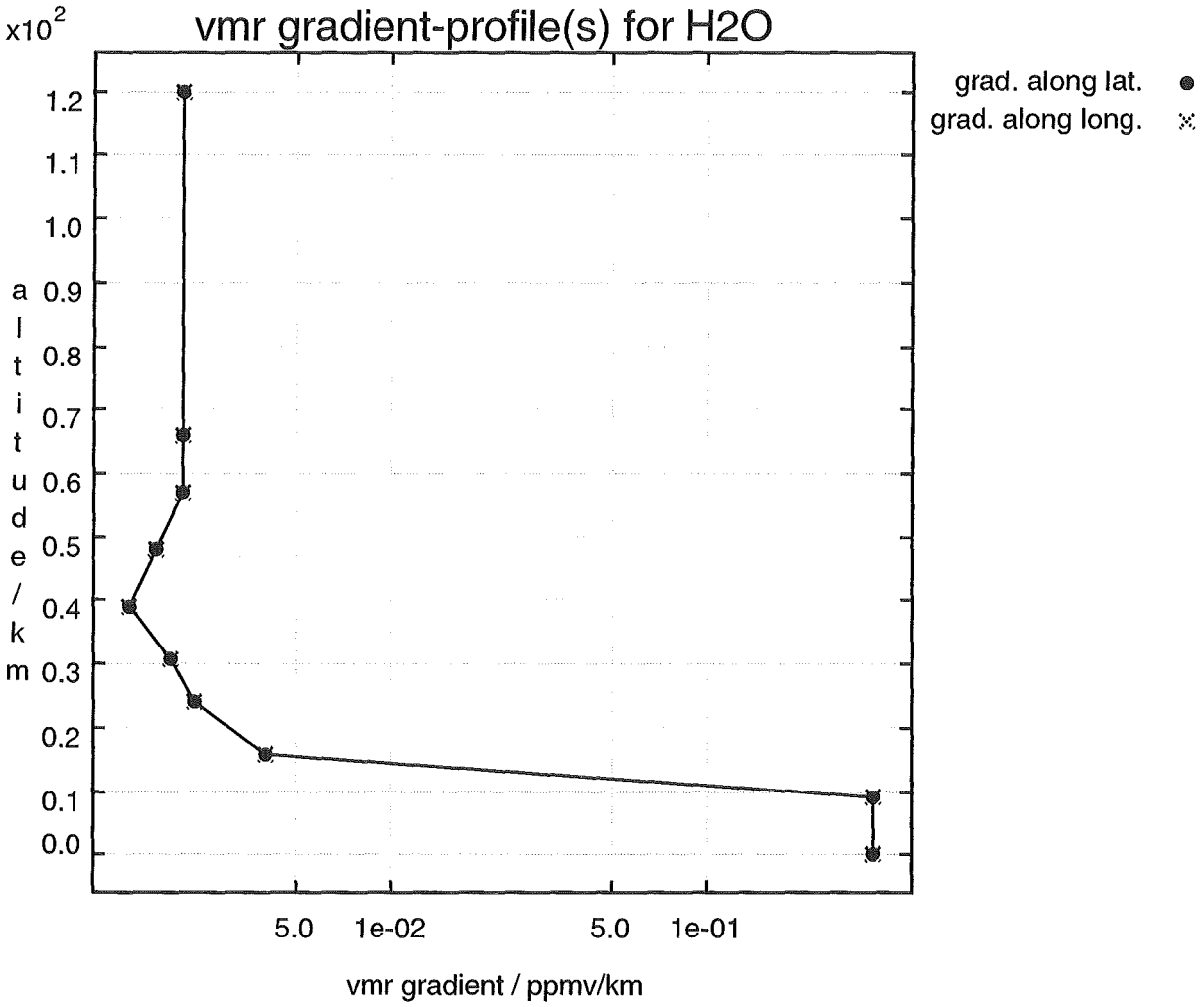


Figure 2: Horizontal inhomogeneities for H_2O

Table 3: Results of test cases: relative retrieval errors

Source of error	790.0-800.0 cm^{-1}	1861.3-1861.9 cm^{-1}	870.875-871.975 cm^{-1}
Target species, tangent altitude	CCl_4, 8km	H_2O, 50 km	HNO_3, 22 km
Noise	2.6 %	35.7 %	6.7 %
Grad T	11.0 %	-3.7 %	1.8 %
Grad vmr	8.2 %	/	-3.9 %
Grad p	-0.06 %	-2.9 %	-2.5 %
Isotopic abundances	/	0.2 %	/
NLTE	-5.8 %	3.4 %	/
Line mixing	-42.3 %	0	/
ILS	0.2 %	3.0 %	2.6 %
FOV	-42.6 %	0.3 %	3.0 %
Total	91.6 %	36.6 %	8.9 %

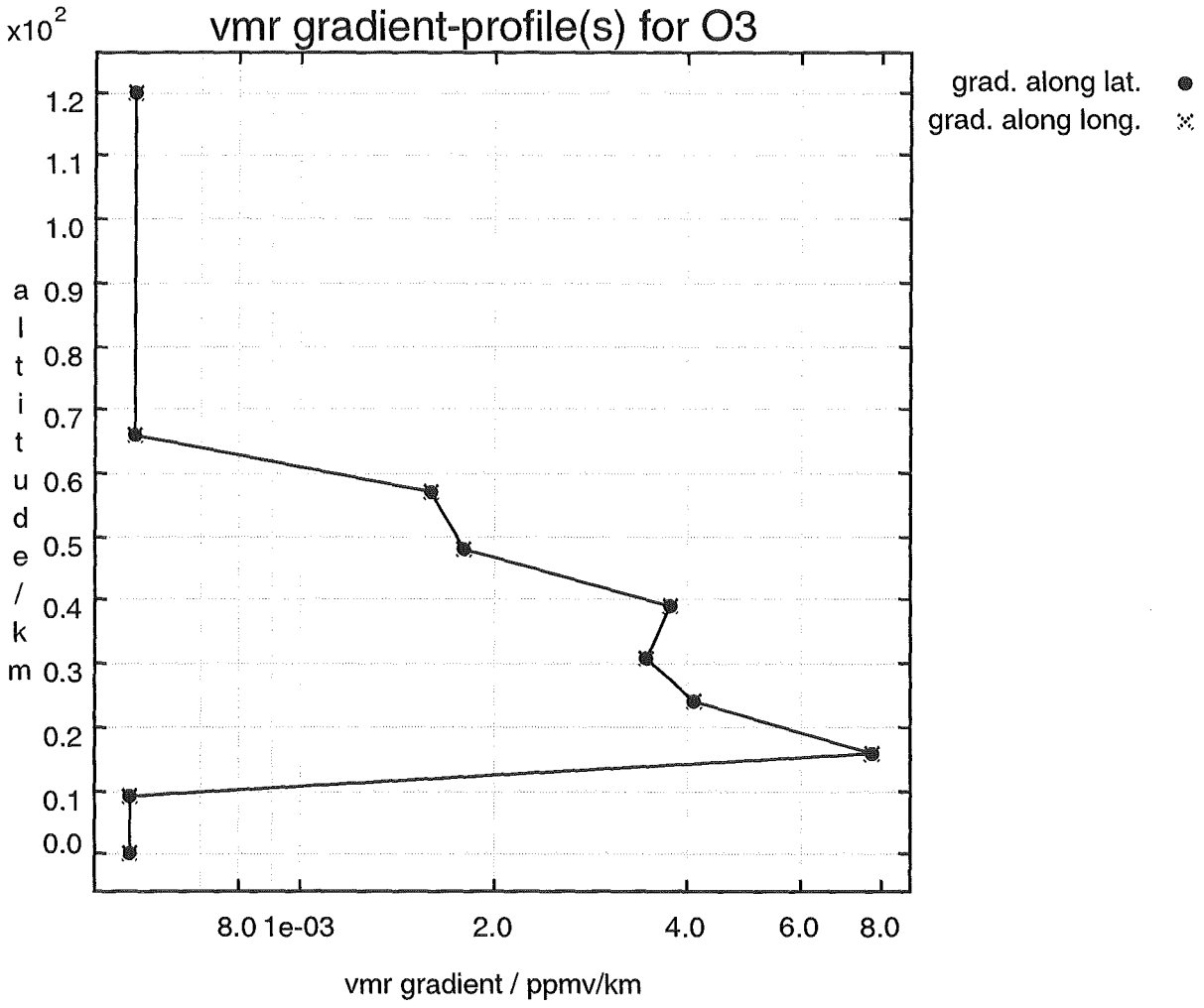


Figure 3: Horizontal inhomogeneities for O_3

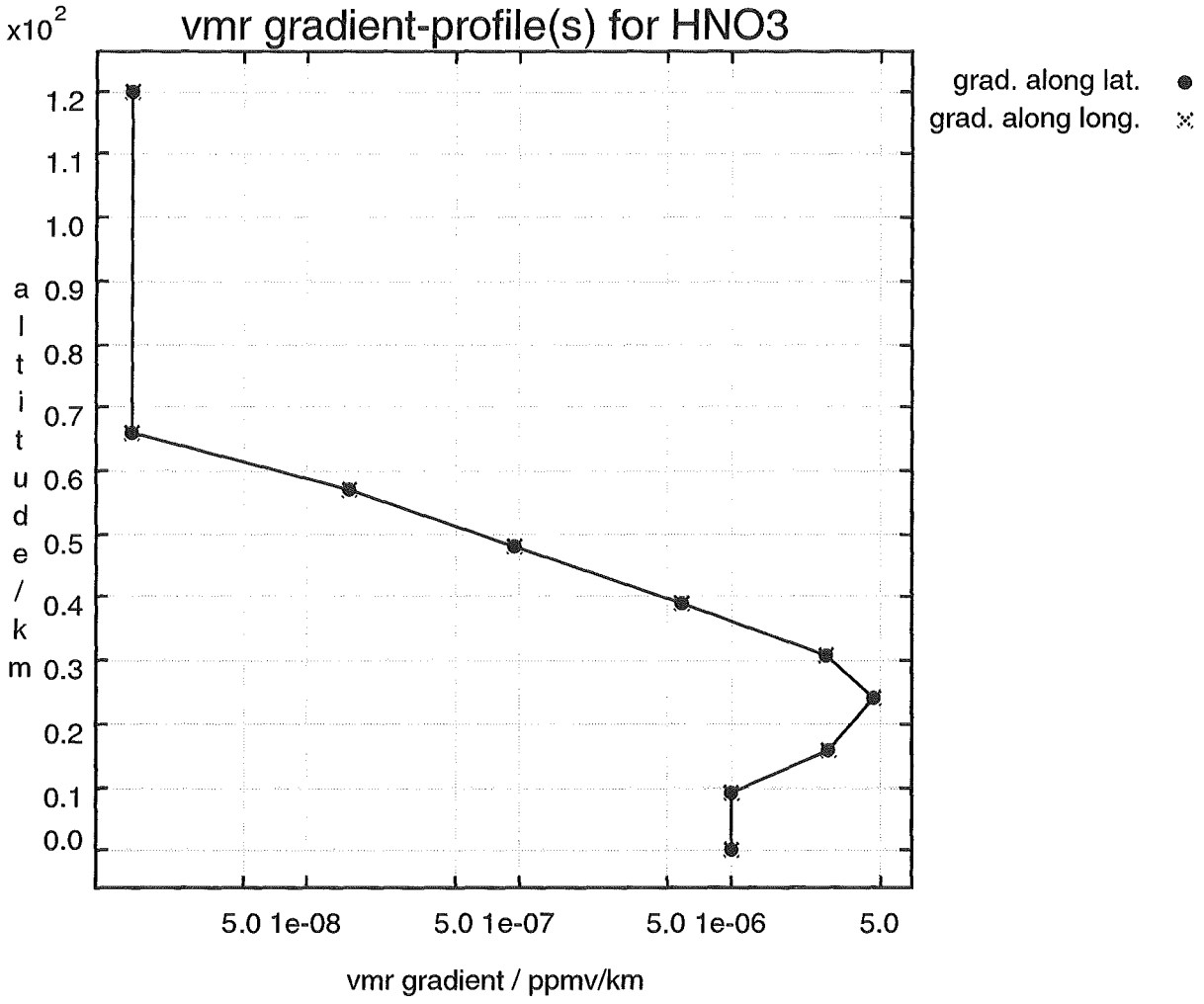


Figure 4: Horizontal inhomogeneities for HNO_3

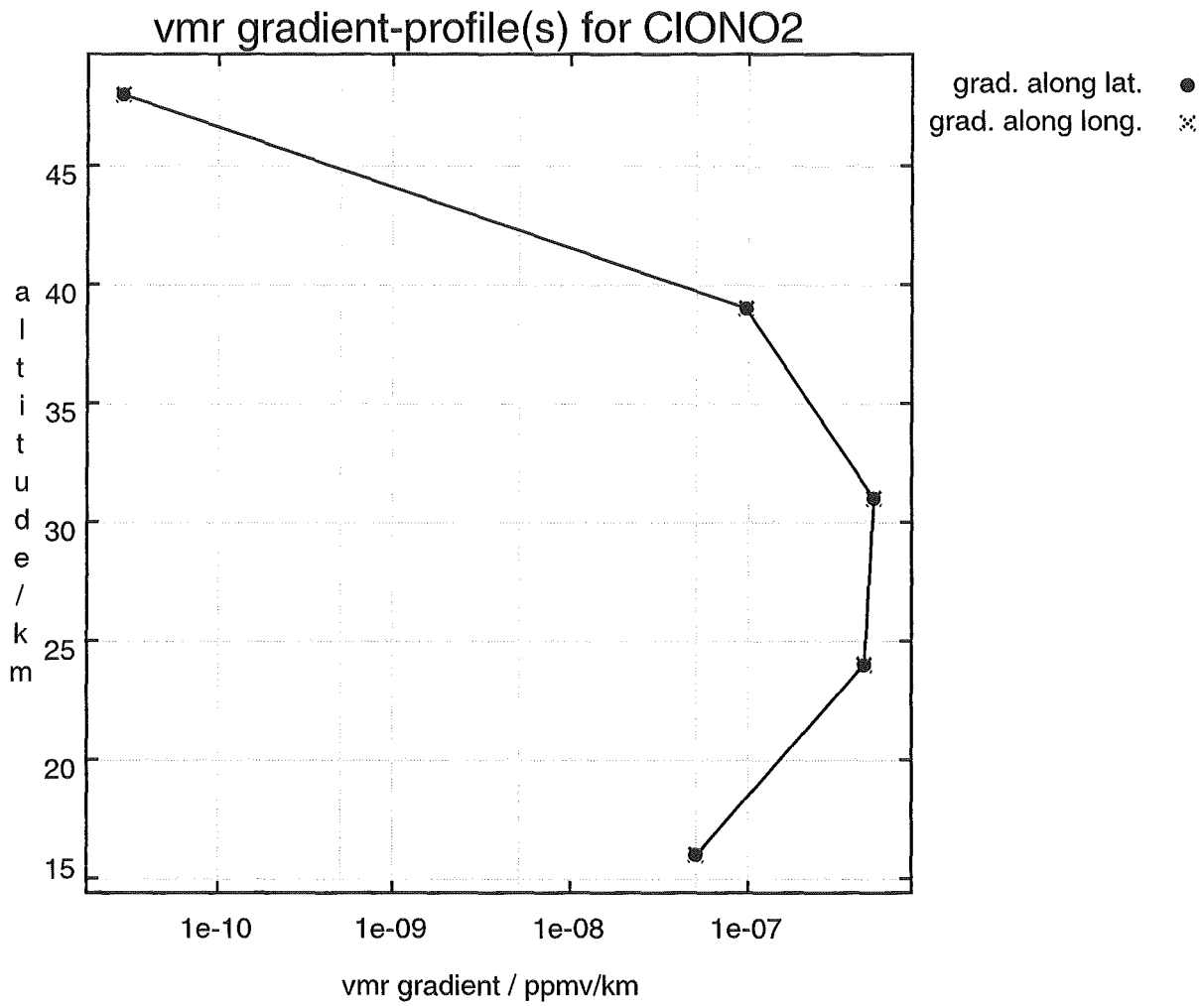


Figure 5: Horizontal inhomogeneities for $ClONO_2$

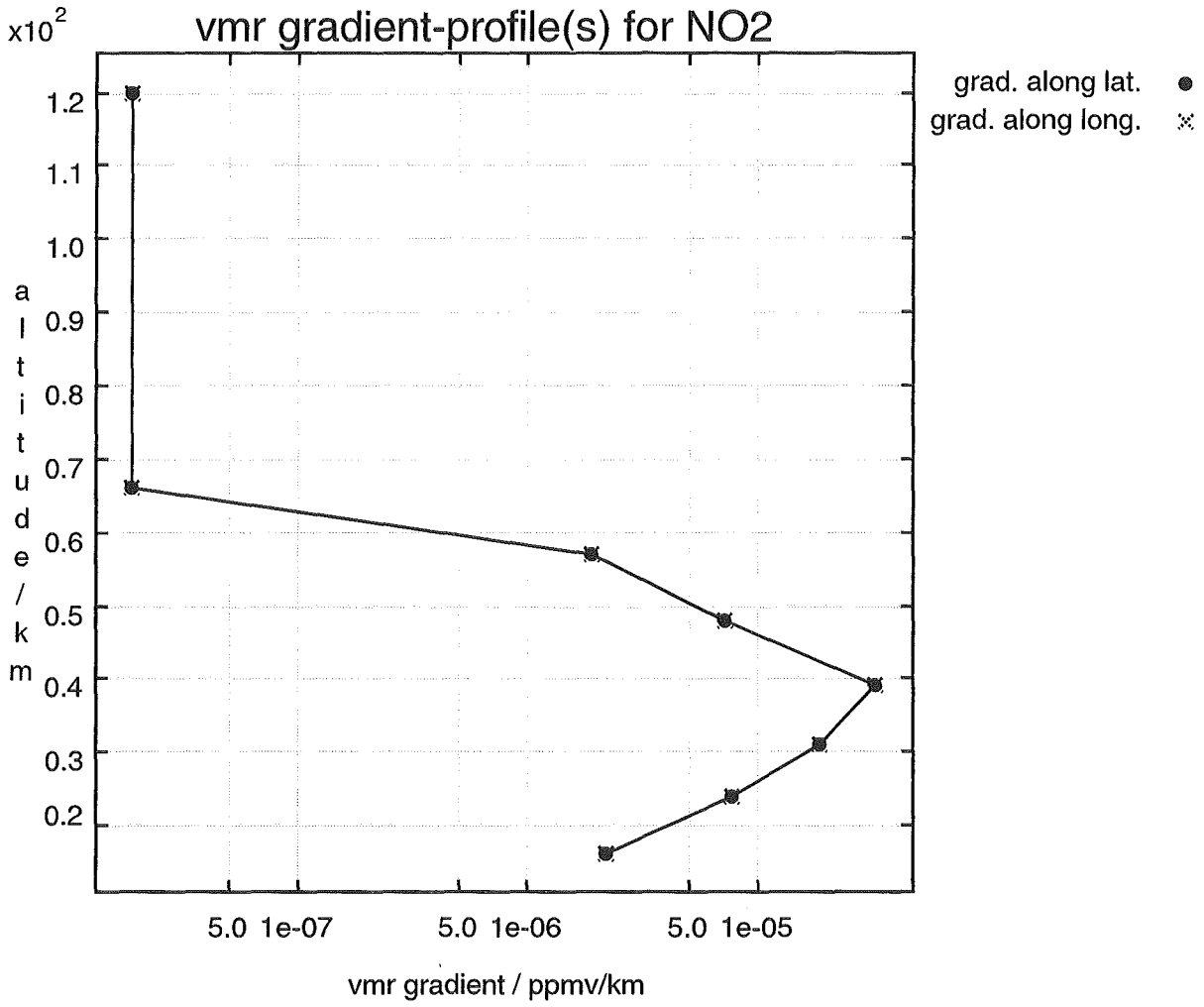


Figure 6: Horizontal inhomogeneities for NO_2

3.2.1 CCl_4 microwindow, 790.0 - 800.0 cm^{-1}

For this microwindow, transitions of the species H_2O, CO_2, O_3, NO_2, NH_3, HNO_3, ClO, HCN, C_2H_2, C_2H_6, COF_2, $ClONO_2$, and CFC-22 were taken into account, besides the target species CCl_4. The microwindow was selected to demonstrate the effect of neglecting CO_2 Q-branch line-mixing on the retrieval error, as the 791 cm^{-1} Q-branch of the 11101 $\leftarrow$ 10002 CO_2 transition is located on top of the CCl_4 signature. Retrieval is possible for tangent heights in the upper troposphere and lower-most stratosphere only, due to the strong decrease of CCl_4 vmr with altitude. Besides the dominating effect of line mixing, disregarding of horizontal temperature and vmr gradients, FOV and NLTE turned out to be relevant error sources. The random error of 2.6 % in 8 km tangent height is increased to a total error of 91.6 % by the errors through imperfect modeling. Spectra demonstrating the effects of disregarding the respective parameters are shown in Fig. 8 to Fig. 12, while Fig. 7 demonstrates for comparison the effect of an 5 % increase of CCl_4 vmr in the tangent layer.

3.2.2 H_2O microwindow, 1861.3 - 1861.9 cm^{-1}

This microwindow was selected as being suitable for retrieval of upper stratospheric water vapor. It contains transitions of H_2O, CO_2, O_3, N_2O, CH_4, NO and NH_3. NLTE effects were considered for H_2O, CO_2, O_3, N_2O and NO. We investigated the effects of neglecting line mixing, NLTE, temperature and pressure gradients, isotopomeric abundances, and the FOV and ILS modeling. The retrieval error for H_2O in 50 km tangent height from this microwindow is dominated by noise (35.7 %). Systematic errors contribute only to a minor extent by adding another 1 % to the over-all error budget. However, in a climatological analysis the random error will be reduced, and systematic errors can become predominant.

3.2.3 HNO_3 microwindow, 870.875 - 871.975 cm^{-1}

The microwindow selected here is suitable for retrieving HNO_3 in the lower stratosphere. It contains transitions of H_2O, CO_2, O_3, NO_2, OCS, C_2H_6, ClO, NH_3 and CFC-12. The retrieval error due to noise is assessed to be 6.7 % for 22 km tangent altitude. It is increased to 8.9 % by systematic error contributions of the horizontal inhomogeneities (vmr, temperature and pressure) and the disregarded FOV.

3.3 Conclusions

KOPRA is a powerful tool for modeling radiance and transmittance spectra as well as spatial derivatives of spectral radiances with respect to atmospheric state parameters and instrument parameters. For some typical examples, we demonstrated that the physical and instrumental modeling implemented in KOPRA is of relevance for realistic simulation of the atmospheric radiance and thus for correct retrieval results. Moreover, it must be kept in mind that modeling errors through neglection of such effects are systematic and thus may become a dominant source of retrieval error in case of temporal or spatial averaging of results where measurement noise is largely reduced. Therefore we consider the full treatment of atmospheric physics as implemented in KOPRA relevant and justified, at least for reference purposes in the context of dedicated scientific studies.

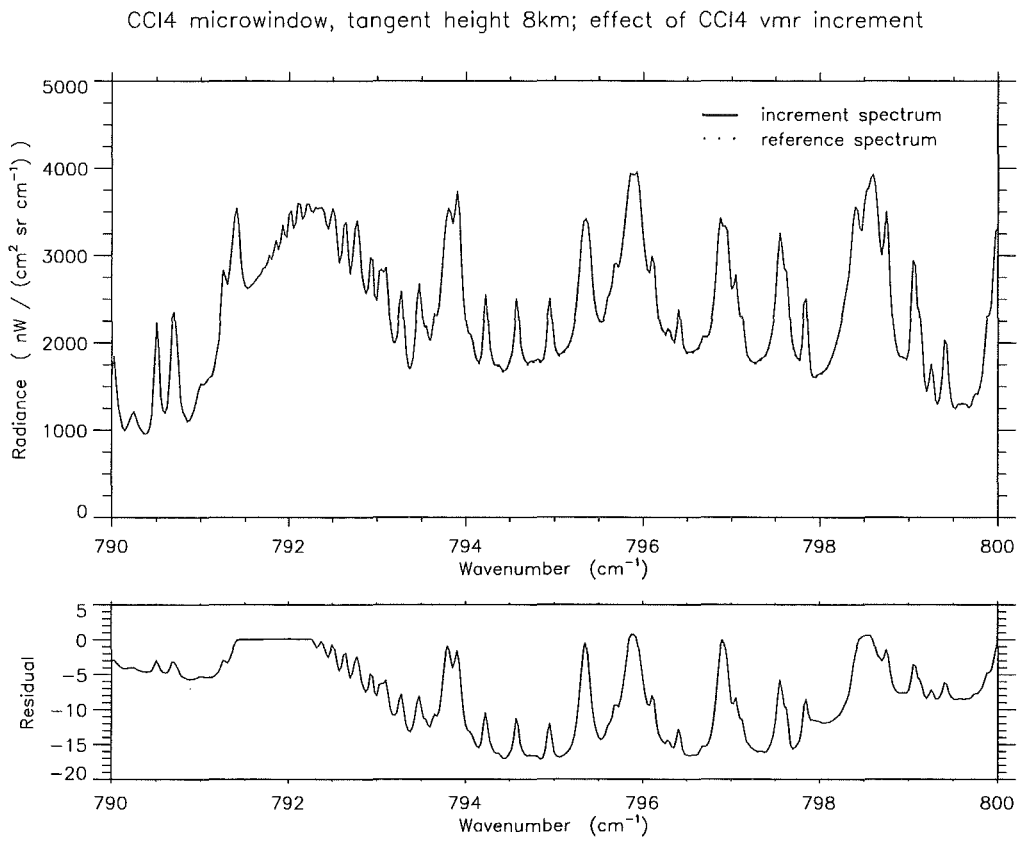


Figure 7: Spectrum of CCl_4 microwindow, tangent height 8 km: Effect of 5 % increase of CCl_4 vmr over 3 km in the tangent layer.

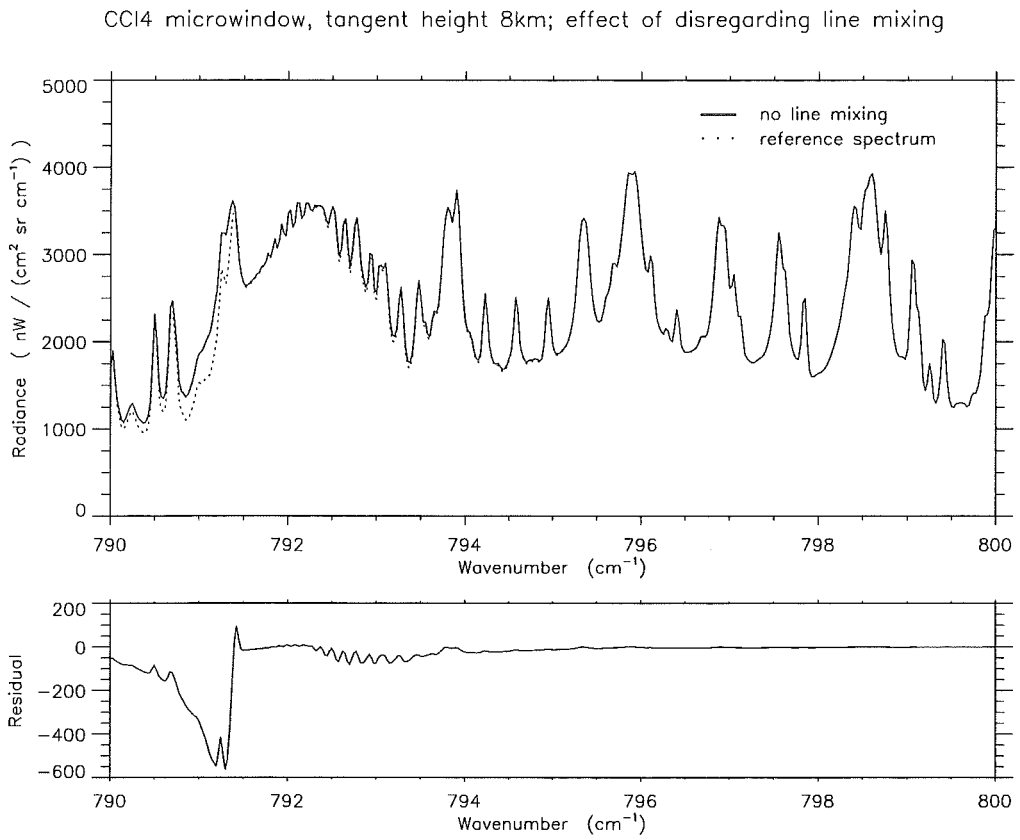


Figure 8: Spectrum of CCl_4 microwindow, tangent height 8 km: Effect of disregarding CO_2 Q-branch line mixing.

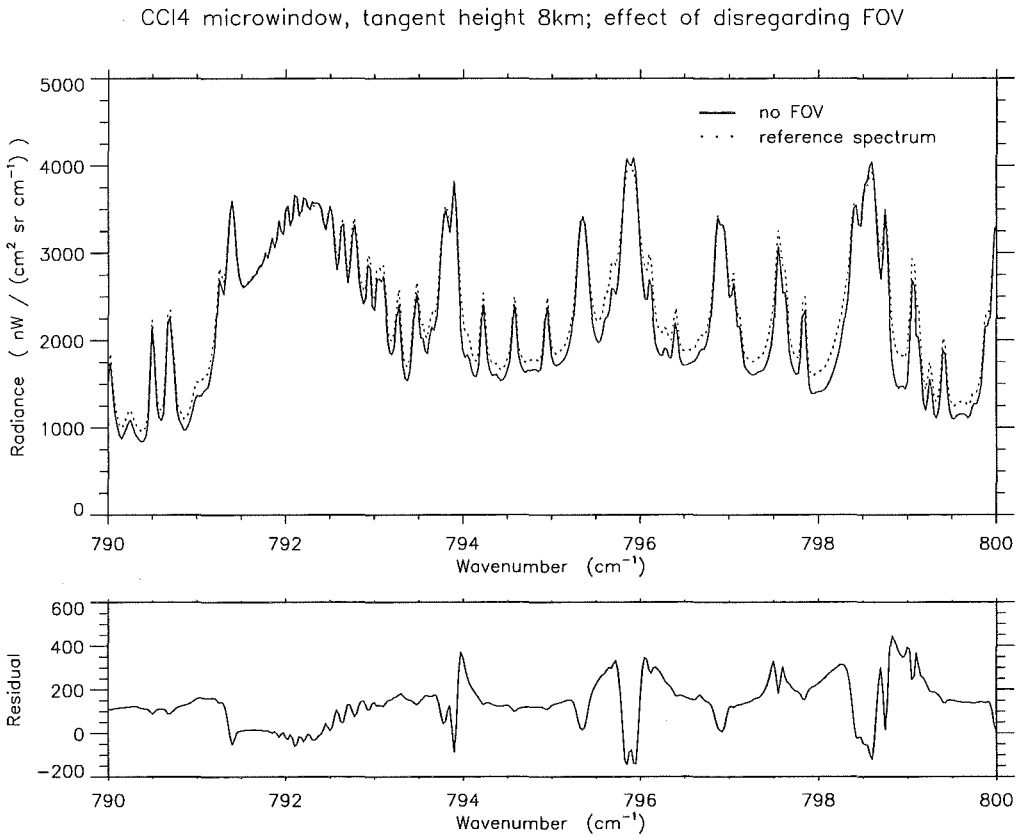


Figure 9: Spectrum of CCl_4 microwindow, tangent height 8 km: Effect of disregarding FOV modeling.

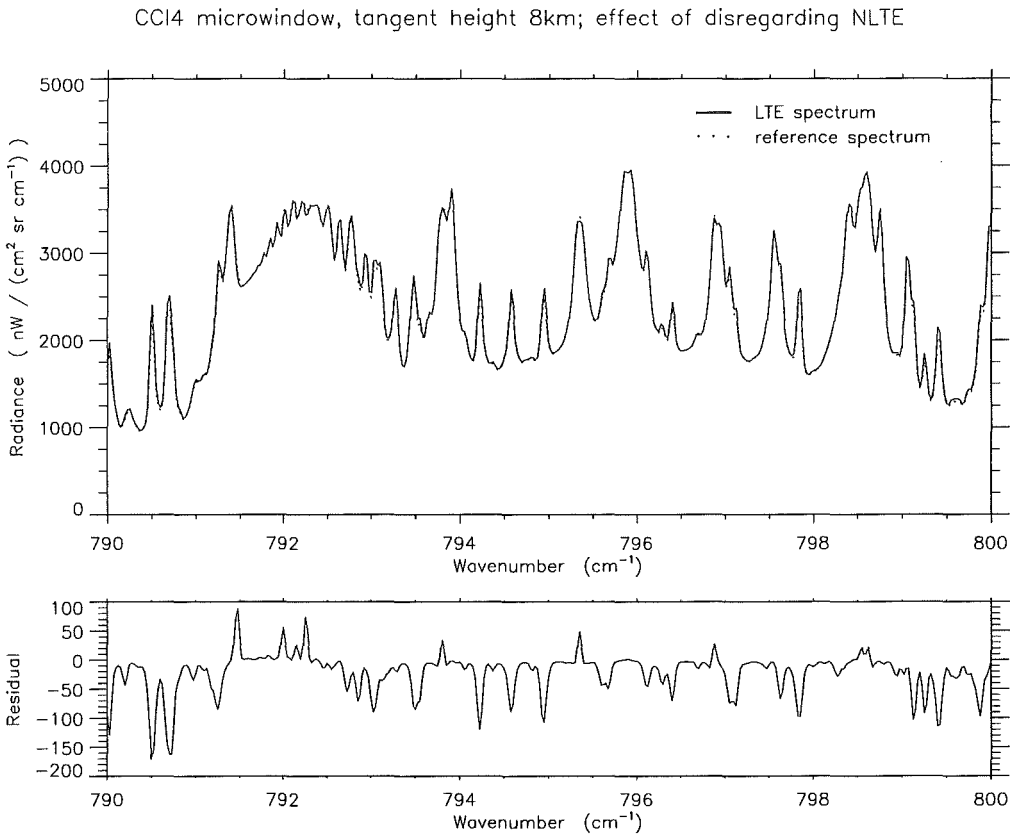


Figure 10: Spectrum of CCl_4 microwindow, tangent height 8 km: Effect of disregarding NLTE modeling.

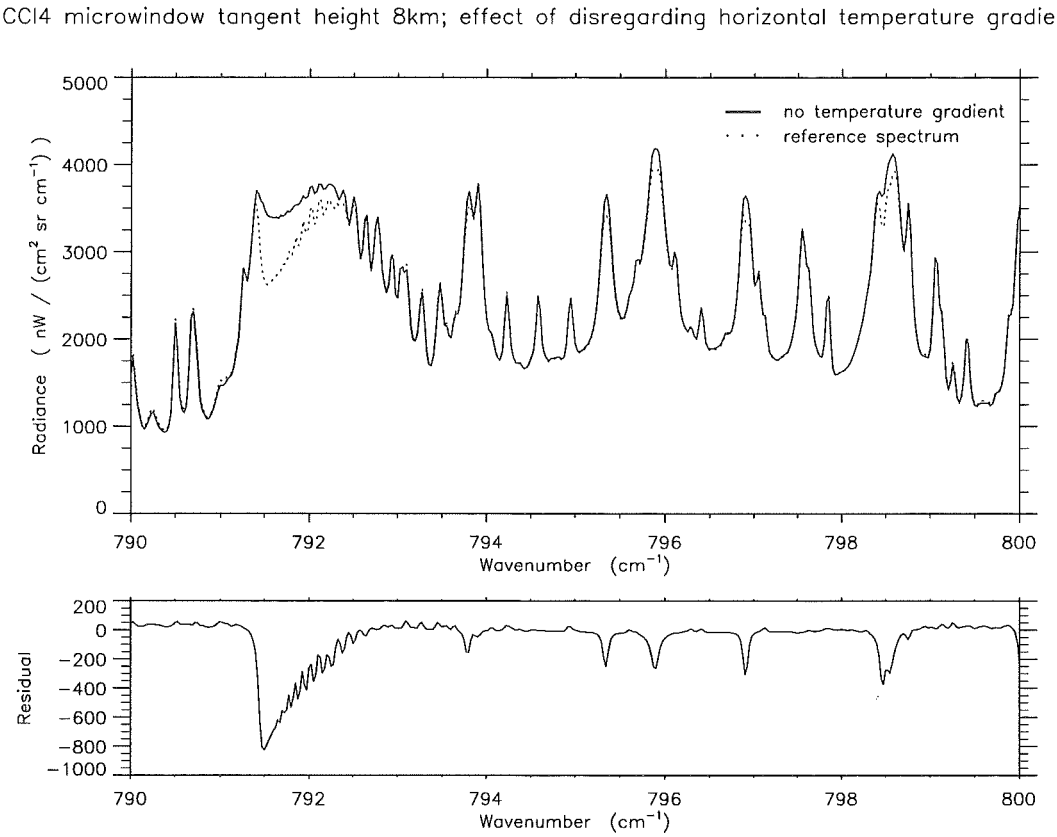


Figure 11: Spectrum of CCl_4 microwindow, tangent height 8 km: Effect of disregarding horizontal temperature gradients.

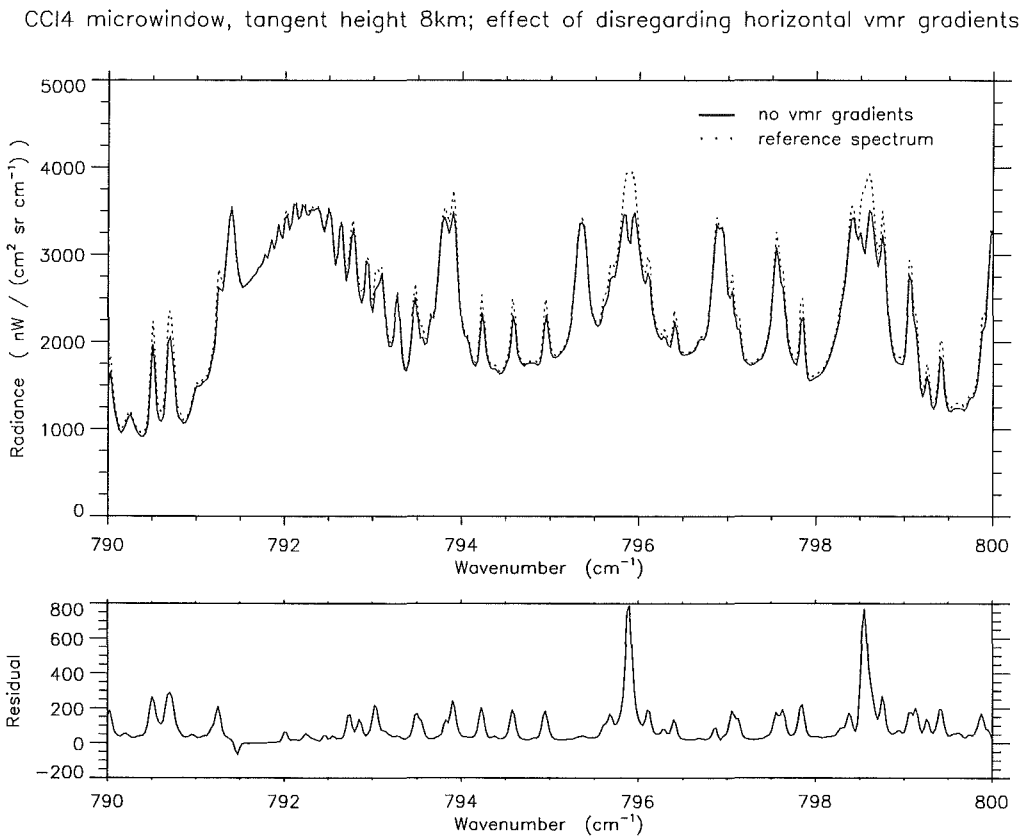


Figure 12: Spectrum of CCl_4 microwindow, tangent height 8 km: Effect of disregarding horizontal vmr gradients.

Bibliography

[1] F. Hase and M. Höpfner, "Atmospheric ray path modeling for radiative transfer algorithms," *Appl. Opt.* **38**(15), pp. 3129–3133, 1999.

[2] T. von Clarmann and G. Echle, "Selection of optimized microwindows for atmospheric spectroscopy," *Appl. Opt.* **37**(33), pp. 7661–7669, 1998.

[3] C. D. Rodgers, "Information content and optimisation of high spectral resolution measurements," in *Optical and Spectroscopic Techniques and Instrumentation for Atmospheric and Space Research II*, P. B. Hays and J. Wang, eds., *SPIE* **2830**, pp. 136–147, 1996.

[4] M. Höpfner, G. P. Stiller, M. Kuntz, T. von Clarmann, G. Echle, B. Funke, N. Glatthor, F. Hase, H. Kemnitzer, and S. Zorn, "The Karlsruhe optimized and precise radiative transfer algorithm. Part II: Interface to retrieval applications," in *Optical Remote Sensing of the Atmosphere and Clouds, Beijing, China, 15–17 September 1998*, J. Wang, B. Wu, T. Ogawa, and Z. Guan, eds., *Proc. SPIE* **3501**, pp. 186–195, 1998.

[5] T. von Clarmann, A. Dudhia, G. Echle, J.-M. Flaud, C. Harrold, B. Kerridge, A. Koutoulaki, A. Linden, M. López-Puertas, M. Á. López-Valverde, F. J. Martín-Torres, J. Reburn, J. Remedios, C. D. Rodgers, R. Siddans, R. J. Wells, and G. Zaragoza, "Study on the simulation of atmospheric infrared spectra," tech. rep., European Space Agency, 1998. Final Report of ESA Contract 12054/96/NL/CN.

[6] H. Nett and M. Endemann, 1996. note of June 24.

[7] *U.S. Standard Atmosphere*, 1976. NOAA–S/T 76–1562, 10.

[8] G. Echle, H. Oelhaf, and A. Wegner, "Measurement of atmospheric parameters with MIPAS," tech. rep., European Space Agency, December 1992. Final Report of ESA Contract 9597/91/NL/SF.

[9] B. Funke and M. López-Puertas, "Nonlocal thermodynamic equilibrium vibrational, rotational, and spin state distribution of NO(ν= 0,1,2) under quiescent atmospheric conditions," *J. Geophys. Res.* **105**(D4), pp. 4409–4426, 2000.

[10] A. Dudhia, "Assessment of horizontal inhomogeneities," tech. rep., Oxford University, October, 6 1997. Task Report on Task 2.3 of ESA Contract 12055/96/NL/CN.

[11] B. M. Dinelli, B. Carli, and M. Carlotti, "Measurement of stratospheric distributions of $H_2{}^{16}O$, $H_2{}^{18}O$, $H_2{}^{17}O$, and $HD^{16}O$ from far infrared spectra," *J. Geophys. Res.* **96**(D4), pp. 7509–7514, 1991.

[12] F. W. Irion, M. R. Gunson, C. P. Rinsland, Y. L. Yung, M. C. Abrams, A. Y. Chang, and A. Goldman, "Heavy ozone enrichments from ATMOS infrared solar spectra," *Geophys. Res. Lett.* **23**(17), pp. 2377–2380, 1996.

KOPRA architecture

M. Höpfner

Abstract: An overview of KOPRA's architecture is given starting from the coarse structure and ending with pseudo-code including all subroutines with a description of their tasks. The main program variables which are responsible for the data flow between the modules are described. An overview of the module tree is presented.

1 Program structure

1.1 Coarse structure

- Definition of input variables
- Forward model run
- Output of spectra and derivatives of spectra wrt retrieval parameters
- Deallocation of variables

1.2 More detailed structure with main modules

- Definition of input variables
  - Read main input and initialize variables.
    (input_m)
  - Read fixed parameters and initialize variables.
    (input_m)
  - Read atmospheric data and initialize variables.
    (input_m)
  - Read spectroscopic data and initialize variables.
    (inspec_m)
  - Read cross-section data on heavy molecules and initialize variables.
    (xinput_m)
  - Initialization of retrieval parameter vectors.
    (inipar_m)
- Forward model run
  - Initialization of derivative variables
    (inider_m)
  - Initialization of internal sub-microwindows
    (inismw_m)
  - Adding of additional geometries to the observed ones for simulation of field-of-view effects.
    (modgeo_m)
  - Adding of additional atmospheric levels to the input levels for more precise simulation of radiative transfer.
    (modlev_m)
  - Ray-tracing, calculation of path integrated values (Curtis-Godson values and column amounts) and the derivatives of path integrated values wrt retrieval parameters.
    (rayctl_m)
  - Calculation of absorption cross sections for each atmospheric path for hitran gases and heavy molecules. In case of nlte the cross sections are calculated for each nlte band. For T derivatives the derivatives of the cross sections wrt CG-T of the paths are determined.
    (abco_m)
  - Radiative transfer: the monochromatic spectra and their derivatives wrt the retrieval parameters are determined.
    (radtra_m)
  - Convolution of the monochromatic spectra and their derivatives wrt retrieval parameters with the AILS function and FOV calculation.
    (radtra_m,ilsfov_m)
  - Initialization of output.
    (iniout_m)
- Output of spectra and derivatives of spectra wrt retrieval parameters
  - Write convolved spectra and their derivatives wrt retrieval parameters into file.
    (wriout_m)

- Deallocation of variables.
  (deallo_m)
  - Deallocate all variables.

1.3 Detailed structure with main subroutines

- Definition of input variables (the following subroutines are called by (kopra))
  - (input@input_m)
    Control the input from files and define input variables
    * (input_main@input_m)
      Read from main input file:
      input file and directory names for fixed data
      input file and directory names for atmospheric/instrumental data
      · (input_hitmol@input_m)
        Read hitran info data on molecules
      · (input_isoprof@input_m)
        Read isotope abundance profiles
      Read from main input file:
      header for output files
      mode of observation and definition of geometry
      wavenumber discretization
      parameters for the adjustment of computational accuracy
      · (input_mwdef@input_m)
        Read microwindow definition section of main
        input file
      · (speciorder@input_m)
        Definition of the species and internal species numbering
      Read from main input file:
      instrumental parameters
      derivative definition
      profile definition
      · (input_pTprof@input_m)
        Read p,T profiles
      · (input_vmrprof@input_m)
        Read vmr profiles
      · (make_hydroequi@input_m)
        Bring input z or p profiles into internal hydrostatic equilibrium
      · (input_contprof@input_m)
        Read the aerosol absorption coefficient profiles for each mw
      · (input_pTgradprof@input_m)
        Read p,T - gradient profiles
      · (input_vmrgradprof@input_m)
        Read vmr - gradient profiles
      Read from main input file:
      line-mixing-parameters
      non-lte-parameters
      · (input_Tvibprof@input_m)
        Read Tvib-profiles

- (input_Tvibgradprof@input_m)
  Read Tvib-gradient-profiles
- (ils_radius@input_m)
  Determination of the radius (in multiples of the fine grid distance wgrid%fine) where the chosen apodization function 'accu%iapo' is decreased to some percentage of it's center value.
- (extend_mw@input_m)
  Extend the microwindow boundaries by the ilsradius.

* (input_spectroscopy@inspec_m)
  Controls the input of the spectroscopic data, i.e. the determination of the spectroscopic data type construction speci%iso%band%branch%line...
  This is performed for different options:
  no line mixing, no nlte
  no line mixing, nlte
  line mixing, no nlte
  line mixing, nlte
  - (numdata@inspec_m)
    Determine how many and which spectroscopy files have to be used
  - (readlines@inspec_m)
    Read spectroscopic data
  - (isoabun@inspec_m)
    Multiply line strength with isotope abundancy if species is single isotope

  If no line-mixing and no nlte:
  - (allocno_nlte@inspec_m)
    Allocate speci%iso and speci%iso%band in the case no nlte is considered
  - (allocno_lm@inspec_m)
    Allocate speci%iso%band%branch in the case no line mixing is considered
  - (allocopy1@inspec_m)
    Allocate speci()% vector and copy line data in case no nlte and no line mixing is considered

  If no line mixing, but nlte:
  - (alloc_nlte@inspec_m)
    Determine number of different isotopes for each species where nlte has to be considered and the number of nlte bands. Allocate speci%iso and speci%iso%band.
  - (allocno_lm@inspec_m)
    Allocate speci%iso%band%branch in the case no line mixing is considered
  - (allocopy2@inspec_m)
    Allocate the vectors speci()%iso()%band()%branch(0)%line and copy the line data from vector spe()%line into this vectors.

  If line mixing but no nlte:
  - (allocno_nlte@inspec_m)
    Allocate speci%iso and speci%iso%band in the case no nlte is considered
  - (read_lmdata@inspec_m)
    Read the line mixing data from file fil%linemix into vector 'lm'.

Only the branches which are 'near' a microwindow (distance w_linemix) are read.

· (alloc_lm@inspec_m)
Allocate the line mixing branches of vector speci and copy the line data of the line mixing lines from the vector lm into speci()%iso()%band()%branch()%line and speci()%iso()%band()%branch()%lmline

· (allocopy2@inspec_m)
Allocate the vectors speci()%iso()%band()%branch(0)%line and copy the line data from vector spe()%line into this vectors.

· (delete_lmlines@inspec_m)
Deletes lines from the %branch(0)%line lines if they are also included in the line mixing branches, so that these lines are not calculated twice.

If line-mixing and nlte:

· (alloc_nlte@inspec_m)
Determine number of different isotopes for each species where nlte has to be considered and the number of nlte bands. Allocate speci%iso and speci%iso%band.

· (read_lmdata@inspec_m)
Read the line mixing data from file fil%linemix into vector 'lm'. Only the branches which are 'near' a microwindow (distance w_linemix) are read.

· (alloc_lm@inspec_m)
Allocate the line mixing branches of vector speci and copy the line data of the line mixing lines from the vector lm into speci()%iso()%band()%branch()%line and speci()%iso()%band()%branch()%lmline

· (allocopy2@inspec_m)
Allocate the vectors speci()%iso()%band()%branch(0)%line and copy the line data from vector spe()%line into this vectors.

· (delete_lmlines@inspec_m)
Deletes lines from the %branch(0)%line lines if they are also included in the line mixing branches, so that these lines are not calculated twice.

End if

· (pointmw@inspec_m)
Determine for each microwindow i the range of lines which will be used: speci()%...%mw_l1(i) speci()%...%mw_l2(i).

* (input_xsection@xinput_m)
Controls the input of the cross-section data for heavy molecules. Determination of the construct: speci%cross%...

· (meas_range@xinput_m)
Determine laboratory measuring range for every microwindow/xsection-gas and tangent altitude

· (readx@xinput_m)
Read heavy molecule cross section data

– (ini_para@inipar_m)
Initialization of parameter vector para&...

    * (ini_para_vmr@inipar_m)
      initialization of parameter vector for vmr parameters
    * (ini_para_T@inipar_m)
      initialization of parameter vector for temperature parameters
    * (ini_para_Tvib@inipar_m)
      initialization of parameter vector for nlte-vibrational temperature parameters
    * (ini_para_aerabs@inipar_m)
      initialization of parameter vector for aerosol absorption coefficient parameters
    * (ini_para_Tgrad@inipar_m)
      initialization of parameter vector for temperature gradient parameters
    * (ini_para_vmrgrad@inipar_m)
      initialization of parameter vector for vmr gradient parameters
    * (ini_para_p@inipar_m)
      initialization of parameter vector for pressure parameters
  - (kopra_forwrd@kopfwd_m)
    Performs the forward model run (see below '* Forward model run')
    Calculates numerical pressure derivatives

- Forward model run
  (the following subroutines are called by (kopra_forwrd@kopfwd_m))
  - (ini_deri@inider_m)
    Define deri% variable
  - (ini_sub_mw@inismw_m)
    Determines internal forward model sub-microwindows
  - (make_modelgeo@modgeo_m)
    Control of adding of additional geometries to the observed ones for simulation of field-of-view effects
    If no field-of-view is calculated:
    define the sim% variable exactly the same as the obs% variable from the input
    * (make_occusim@modgeo_m)
      Determine the microwindow occupation matrix for the simulated geometries

    If field-of-view is calculated:
    In the case the tangent altitudes are given:
    determine a rough estimate for the observation angle (without refraction and earth ellipsoid)
    Add geometries
    * (addsim_a@modgeo_m)
      Add simulated geometries to the observed ones
      for criterion accu%ifov <= 0 (criterion 1 in input-file)
    * (addsim_a@modgeo_m)
      Add simulated geometries to the observed ones
      for criterion accu%ifov > 0 (criterion 2 in input-file)
    * (make_occusim@modgeo_m)
      Determine the microwindow occupation matrix for the simulated geometries

In the case the tangent altitudes of the observations are given:
calculate tangent altitudes of the additional simulation geometries

- (make_modelgrid@modlev_m)
  Define the model altitude levels for the forward calculation. Starting from a base grid, levels are added in order to fulfill criteria on the T difference and half-width change between levels. Then a criterion for smaller levels distances directly over tangent altitudes is applied. At the end all levels which are less distant than a threshold are deleted.
  * (grid_t_hw@modlev_m)
    Fine-level gridding using T-differences and half-width changes

  Add model fine levels above simulated tangent altitudes
  If the tangent altitudes are given:
  * (grid_tang@modlev_m)
    Fine-level gridding due to levels above tangent points

  If the observation angles are given:
  * (calc_ztang@modlev_m)
    Determination of the tangent altitudes if the nadir angles of a limb scan are given
    · (tangalt@ray_m)
      calculate estimate for tangent altitude with refraction
  * (grid_tang@modlev_m)
    Fine-level gridding due to levels above tangent points

  End if
  * (min_distance@ray_m)
    Levels which are less distant than accu%dmin are selected out (only the ones that do not belong to the base-grid)

- (raytrace_ctrl@rayctl_m)
  Controls ray-tracing, calculation of path integrated values (Curtis-Godson values and column amounts) and the derivatives of path integrated values wrt retrieval parameters.
  For homogeneous path (cuvette) calculation:
  * (homog_path@rayctl_m)
    Determine the path parameters for homogeneous path calculation

  For atmospheric calculations:
  For each geometry :
  * (raytra@ray_m)
    Calculation of ray-tracing in inhomogeneous atmosphere and path integration
    · (nmax_calc@ray_m)
      Determination of the max. number of integration variables per layer

    Case of satellite tangent altitudes:
    determine index of lowest layer
    · (observer@ray_m)
      Calculate position and viewing direction at tangent altitude in cartesian coordinates

    for both parts of the geometry:
    · (latlon@ray_m)
      Calculate geographic latitude and longitude of cartesian point r

· (tnew@ray_m)
Calculation of new tangent vector along LOS

determine new point of LOS using old and new tangent vector
if a level boundary of the atmospheric model levels is crossed:

· (leveltrans@ray_m)
Find exact level positions along LOS and perform integration for the actual layer

+ (integrate@ray_m)
Explicit integration for layer values and derivatives (columns, Curtis-Godson-T, -p, -Tvib, ...) with frequent calls to give_...@give_m and para_..._change@parchk_m

overwrite old tangent vector with new one
begin again with (tnew@ray_m)
End case of satellite tangent altitudes
Case of satellite observer altitude and elevation angle:

· (observer@ray_m)
Calculate position and viewing direction of observer in cartesian coordinates

· (findtop@ray_m)
Find (coming from outside) the crossing point r with top level of the atmosphere

· (latlon@ray_m)
Calculate geographic latitude and longitude of cartesian point r

Now go through the atmosphere from top to top:

· (tnew@ray_m)
Calculation of new tangent vector along LOS

determine new point of LOS using old and new tangent vector

· (latlon@ray_m)
Calculate geographic latitude and longitude

if the altitude is decreasing/increasing monotonously and
if a level boundary of the atmospheric model levels is crossed:

· (leveltrans@ray_m)
Find exact level positions along LOS and perform integration for the actual layer

+ (integrate@ray_m)
Explicit integration for layer values and derivatives (columns, Curtis-Godson-T, -p, -Tvib, ...) with frequent calls to give_...@give_m and para_..._change@parchk_m

else if the altitude is increasing again:

· (leveltang@ray_m)
perform integration for tangent layer and find exactly tangent position

overwrite old tangent vector with new one
begin again with (tnew@ray_m)
End case of satellite tangent altitudes
Cases for other observation modes are handled equivalently.
Write output (integrated values) in geo%... vector

· (ray_out@ray_m)
Prepare output variable geo()%... which contains all path parameters

+ (alloc_geo@ray_m)
  Allocate geo()%...%lay and geo()%...%lay%speci geo%...%lay%speci%iso , geo%... %iso%state
+ (pathcopy@ray_m)
  Copy path parameters into vector geo%...

If nlte should be considered:

* (calc_nlte_ratios@rayctl_m)
  The derivative of the ratio of nlte/lte with respect to Tkin is calculated.

If vibrational temperature derivatives should be calculated:

* (para_dTvib_ne0@rayctl_m)
  Determine the parameters which do not influence the Tvib's i.e. for which the derivative dTvib_cg_dT = 0

If vmr derivatives are calculated:

* (para_dcol_ne0@rayctl_m)
  Determine the parameters which do not influence the partial columns i.e. for which the derivative dcol = 0

If aerosol absorption derivatives are calculated:

* (para_dabsopt_ne0@rayctl_m)
  Determine the parameters which do not influence the aerosop absorption optical depth i.e. for which the derivative daeropt = 0

If p derivatives are calculated:

* (para_dp_ne0@rayctl_m)
  Determine the p parameters which do not influence the cg-p of air i.e. for which the derivative dp = 0

If T derivatives are calculated:

* (para_dT_ne0@rayctl_m)
  Determine the T parameters which do not influence the cg-T of air i.e. for which the derivative dT = 0

If T-gradient derivatives are calculated:

* (para_dTgrad_ne0@rayctl_m)
  Determine the Tgrad parameters which do not influence the cg-T of air i.e. for which the derivative dTgrad = 0

If vmr derivatives are calculated:

* (para_dcolgrad_ne0@rayctl_m)
  Determine the vmr gradient parameters which do not influence the partial columns i.e. for which the derivative dcol = 0

– (absco_calc@abco_m)
Calculation of absorption cross sections ('absorption coefficients') [cm^2/molecule] for each atmospheric path for hitran gases and heavy molecules. In case of nlte the cross sections are calculated for each nlte band. For T derivatives the derivatives of the cross sections wrt CG-T of the paths are determined.

* (allocate_cutoff@addlin_m)
  Allocates memory and reads cutoff-files "cutdop.dat" and "cutlor.dat"
* (allocate_geo_mw@abco_m)
  Allocates geo%...%mw part of geo% variable (part where absorption coefficients will be stored)

Begin loop on microwindows

* (allocate_grid@addlin_m)
  Allocate and initialize the absco grid
* (allocate_x@xintpl_m)
  Allocate and initialize the cross-section grid

Begin loop on simulation geometries
Begin loop on geometry-parts for which absorption coefs. will be calculated explicitly
Begin loop on layers
Begin loop on line-data species for line-by-line calculation
Calculate lte absorption coefficients:

* (absco_branch@abco_m)
  Calculates the absorption coefficient of a branch
  - (line_strength@abco_m)
    Calculates the line intensities and optionally the T-derivatives for a bundle of lines

  In case of line-mixing calculation with direct diagonalization:
  - (y_calc_dd@linmix_m)
    Calculate the y-coefficients for direct diagonalization

  In case of line-mixing calculation with Rosenkranz-approximation
  - (y_calc_rk@linmix_m)
    Calculates the y-coefficients for the Rosenkranz-approximation

  In both cases:
  - (corr_y_coefs@linmix_m)
    Correct the y-coefficients

  In case of lines without chi-factor:
  - (add_lines@addlin_m)
    Add the individual lines on the irregular grid
  - (add_lines_lm@addlin_m)
    Add the individual lines on the irregular grid with line mixing

  In case of lines with chi-factor:
  - (add_lines_chi@addlin_m)
    Add the individual lines on the irregular grid
  - (add_lines_chilm@addlin_m)
    Add the individual lines on the irregular grid with line mixing

Store absorption coefficients in geo%...%absco:

* (init_absco@transf_m)
  Initialize absco-variable
* (interpolate_grid@addlin_m)
  Interpolates the intervals of the lower grid to the
  Calculate non-lte absorption coefficients: (equivalent to calculation of lte absorption coefficients above, only that the band index is now /=0)
  Calculate gas continua for line-data species:
* (n2calc@transf_m)
  - (calc_n2cont@gascon_m.f90)
    Calculates the absorption cross section of the N_2-continuum in units of [cm^2/molec] as a function of wavenumber, N_2 density and temperature. The derivative of the N_2-continuum with respect to temperature is calculated optionally.

  * (o2calc@transf_m)
    - (calc_o2cont@gascon_m.f90)
      Calculates the absorption cross section of the O_2-continuum in units of [cm^2/molec] as a function of wavenumber, O_2 density and temperature. The derivative of the N_2-continuum with respect to temperature is calculated optionally.
  * (h2ocalc@tansf_m)
    - (calc_h2ocont@gascon_m.f90)
      Calculates the absorption cross section of the H_2O-continuum in units of [cm^2/molec].

End loop on line-data species for line-by-line calculation
Begin loop on cross-section species for heavy-molecule cross section calculation

  * (interpl_xpt@xintpl_m)
    Pressure, temperature and wavenumber grid interpolation of heavy molecule cross-section measurements

Store absorption coefficients in geo%...%absco
End loop on cross-section species for heavy-molecule cross section calculation
For geometries higher than the lowest one:
use already calculated cross sections from the lowest geometry depending on accuracy parameter accu%iexpath
End loop on layers
End loop on geometry-parts for which absorption coefs. will be calculated explicitly
For a geometry-part for which the absorption coefs. are not calculated explicitly:
copy geometry part 1 absorption coefficients to geometry part 2
End loop on simulation geometries

  * (deallocate_x@xintpl_m)
    Deallocate the cross-section grid
  * (deallocate_grid@addlin_m)
    Deallocate the absco grid

End loop on microwindows

  * (deallocate_cutoff@addlin_m)
    Deallocates space used for the cutoff-tables

- (radtrans@radtra_m)
  Calculation of the radiative transfer through the atmosphere and determination of the fine-grid spectra and their derivatives. Convolution of the fine-grid spectra with the ails to get the spectra on the coarse-grid (measurement grid) and calculation of the field-of-view weighting.

  * (alloc_Sails@radtra_m)
    Allocate data vector where the coarse-grid spectrum and derivatives are stored

  Begin loop on microwindows and sub-microwindows
  Allocate radiance and derivative - variables for storage on non-equidistant fine grid
  Calculate radiances and derivatives for each microwindow:

  * (radtrans_mw@radtra_m)
    Begin loop on geometries

Begin loop on geometry-parts and on layers
Calculate radiative transfer layer-by-layer:

- (tausrc@radtra_m)
  Calculation of layer transmission, layer source function (nlte considered) and some derivatives
  Initialize the numerator of the source function
  Initialize the variable where the optical depths are added with the aerosol extinction
  Begin loop on species
  Add the absorption coefficients for all bands of one species and multiply by the partial column of the species in the path to determine the optical depth of the path. Determine also the source function numerator.
  For vmr derivatives calculate d(optical thickness)/d(partial column) of the vmr-derivative-species.
  For vmr-gradient derivatives calculate d(optical thickness)/d(partial column) of the vmr-gradient-derivative-species.
  End loop on species
  In case of continuum derivatives calculate derivative of source function wrt aerosol optical depth
  Calculate source function
  Calculate layer transmission

Calculate radiance at end of layer
Calculate layer-derivatives
End loop on geometry-parts and on layers
Begin loop on geometry-parts and on layers
Multiply layer-derivatives by total transmission between layer and observer
End loop on geometry-parts and on layers
Calculate derivatives with respect to parameters:

- (derivmr_calc@radtra_m)
  Calculation of derivatives with respect to vmr parameters
- (deriaer_calc@radtra_m)
  Calculation of derivatives with respect to aerosol absorption parameters
- (deriT_calc@radtra_m)
  Calculation of derivatives with respect to T parameters
- (deriTvib_calc@radtra_m)
  Calculation of derivatives with respect to Tvib parameters
- (deriTgrad_calc@radtra_m)
  Calculation of derivatives with respect to T-gradient parameters
- (derivmrgrad_calc@radtra_m)
  Calculation of derivatives with respect to vmr-gradient parameters

End loop on geometries

* (ilsapofov_calc@radtra_m)
  Convolution of the fine-grid spectra and their derivatives wrt retrieval parameters with the AILS function and FOV calculation
  If field-of-view is considered:
  Interpolate spectrum of each simulated geometry to the fine grid and store result:

  - (fin@transf_m)

Interpolate non-equidistant fine grid spectrum to equidistant fine grid spectrum

· (sub_mw_minmax@radtra_m)
Determine the wavenumber indices (in the result of fin@transf_m) which are equal to the sub-microwindow boundaries

Begin loop on observation geometries
Calculate ails convolution and field-of-view and determine derivatives with respect to shift, phase and/or linear apodization, (or ESA ils-parameters), elevation angle:
For spherical aperture:

· (fovils1@ilsfov_m)
Calculate ails only if it is necessary:

+ (makeifg@ilsfov_m)

⋮

For ESA-ils:

· (envfovils@ilsfov_m)

⋮

End loop on observation geometries
For vmr, T, Tvib, vmr-gradient, T-gradient, aerosol - derivatives:
Begin loop over parameters

· (lsimobs@radtra_m)
Determine lsim and lobs, the 'activated' simulated and observed geometries i.e. the ones which are influenced by the actual retrieval parameter lpara

· (finewrk@radtra_m)
Interpolate spectrum of each 'activated' simulated geometry to the fine grid and store result:

+ (fin@transf_m)
Interpolate non-equidistant fine grid spectrum to equidistant fine grid spectrum

+ (sub_mw_minmax@radtra_m)
Determine the wavenumber indices (in the result of fin@transf_m) which are equal to the sub-microwindow boundaries

Begin loop on observation geometries
Calculate ails convolution and field-of-view of derivatives:
For spherical aperture:

· (fovils1@ilsfov_m)
Calculate ails only if it is necessary:

+ (makeifg@ilsfov_m)

⋮

For ESA-ils:

· (envfovils@ilsfov_m)

⋮

End loop on observation geometries
End loop over parameters

If no field-of-view is considered:
Begin loop on observation geometries
Interpolate spectrum to the fine grid and store result:
· (fin@transf_m)
Interpolate non-equidistant fine grid spectrum to equidistant fine grid spectrum
· (sub_mw_minmax@radtra_m)
Determine the wavenumber indices (in the result of fin@transf_m) which are equal to the sub-microwindow boundaries
Calculate ails convolution and determine derivatives with respect to shift, phase and/or linear apodization, (or ESA ils-parameters), elevation angle:
For spherical aperture:
· (ilsapo@ilsfov_m)
Calculate ails only if it is necessary:
+ (makeifg@ilsfov_m)
⋮

For ESA-ils:
· (envils@ilsfov_m)
⋮

For vmr, T, Tvib, vmr-gradient, T-gradient, aerosol - derivatives:
Begin loop over parameters
Interpolate spectrum to the fine grid and store result:
· (fin@transf_m)
Interpolate non-equidistant fine grid spectrum to equidistant fine grid spectrum
· (sub_mw_minmax@radtra_m)
Determine the wavenumber indices (in the result of fin@transf_m) which are equal to the sub-microwindow boundaries
Calculate ails convolution of derivatives:
For spherical aperture:
· (ilsapo@ilsfov_m)
Calculate ails only if it is necessary:
+ (makeifg@ilsfov_m)
⋮

For ESA-ils:
· (envils@ilsfov_m)
⋮

End loop over parameters
End loop on observation geometries
Deallocate radiance and derivative - variables for storage on non-equidistant fine grid
End loop on microwindows and sub-microwindows
– (offset_scale@offsca_m)
Add offset and scale spectra. Calculate offset and scale derivatives.

    * (off_deri@offsca_m)
      Calculate offset derivative
    * (sca_deri@offsca_m)
      Calculate scale derivative
    * (sca@offsca_m)
      Multiply spectrum by scale factor
    * (off@offsca_m)
      Add offset to spectrum

  - (ini_output@iniout_m)
    Initialization and determination of the output variable outdat%... (Pointer to Sails%...)
    Allocate data vector outdat%.. where the output spectrum is stored. The sub-microwindow index is now on the external microwindows. The spectra in outdat% are pointers to the spectra stored in Sails% where the sub-microwindow index is on the internal microwindows.

- Output of spectra and derivatives of spectra wrt retrieval parameters (the following subroutines are called by (kopra))

  - (writeout@wriout_m)
    Write convolved spectra and their derivatives wrt retrieval parameters (variable: outdat%) into file

- Deallocation of variables (the following subroutines are called by (kopra) or (kopra_forwrd@kopwd_m))

  - (deallocate_main@deallo_m)
    Deallocate variables.

2 Dataflow and interfaces

The data exchange for the main variables and parameters inside kopra is based on the use of variable/parameter-modules. Parameters and fixed data are made accessible by the modules: param_m, precis_m, xdata_m, and xparam_m. The main variables of the program are contained in the three modules inpdat_m, modat_m, and outdat_m. Whereas in modat_m only the forward-model internal variables are made visible, inpdat_m and outdat_m are used for the interface to the exterior. (Though parts of variables of inpdat_m are also determined inside the forward model.)

In order not to be forced to handle too many different single variables in the forward model user-defined-type variables are used for all the main data of kopra. These structures allow to sort the data in a systematic way. The totally extended variables are described explicitly in the variable description sector of this document.

The variables in the three data-exchange modules are: (in brackets the modules are given where the variables are determined or modified)

In inpdat_m:

accu%	definition of the computational accuracy of the forward model (input_m)
deri%	definition of the derivatives which should be calculated during the forward model run (input_m,inider_m)
fil%	file and directory names (input_m)
inprof%	atmospheric input profiles (input_m)
inst%	instrumental specifications (for ails and fov) (input_m)
mol%	HITRAN molecule/isotope information (input_m)
mw%	definition of microwindows and sub-microwindows (input_m, inismw_m, modgeo_m)
n%	numbers (input_m, modlev_m)
nlte%	non-lte definition (input_m)
obs%	definition of the observation geometry (input_m, modgeo_m)
outheader	header for output data (input_m)
para%	definition of the atmospheric data which are handled as parameters (inipar_m)
speci%	definition of the species containing the spectroscopic data and the cross-section data on heavy molecules (input_m, xinput_m, inspec_m)
sw%	switches (input_m)
wgrid%	definition of the wavenumber grid (input_m)

In modat_m:

geo%	information on the paths along all simulated geometries, i.e. path length, partial column amounts, Curtis-Godson-values, derivatives of Curtis-Godson-values wrt retrieval parameters, and absorption coefficients per microwindow (rayctl_m, ray_m, abco_m)
modprof%	forward model altitude grid (modlev_m)
sim%	definition of the simulated geometries (=observation geometries + geometries necessary for fov-calculation) (modgeo_m, modlev_m)
Sails%	result-spectra and derivatives with internal sub-microwindow indices (radtra_m, offsca_m)

In outdat_m:

outdat%	output spectra and derivatives with external sub-microwindow indices (iniout_m)

3 Module tree

In the following module architecture the dependence on module precis_m is omitted since this describes the precision and is needed by quasi every other modules.
The following dependencies are not written explicitly in the general overview:

```
types_m
        param_m
inpdat_m
        types_m
modat_m
        types_m
outdat_m
        types_m
varsub_m
        param_m
give_m
        inpdat_m
        param_m


kopra
    inpdat_m
    input_m
            param_m
            types_m
            inpdat_m
            recipe_m
            varsub_m
            inspec_m
                    param_m
                    inpdat_m
                    varsub_m
                    recipe_m
            xinput_m
                    types_m
                    inpdat_m
                    xparam_m
                    xdata_m
                            xparam_m
                    varsub_m
    inipar_m
            inpdat_m
    wriout_m
            inpdat_m
            outdat_m
    deallo_m
            inpdat_m
            modat_m
            outdat_m
            transf_m
    varsub_m
    kopfwd_m
            param_m
            inpdat_m
            outdat_m
            varsub_m
            inider_m
                    inpdat_m
            inismw_m
```

```
                inpdat_m
modgeo_m
                param_m
                types_m
                inpdat_m
                modat_m
                varsub_m
modlev_m
                types_m
                inpdat_m
                modat_m
                varsub_m
                give_m
rayctl_m
                inpdat_m
                modat_m
                ray_m
                                param_m
                                inpdat_m
                                modat_m
                                give_m
                                recipe_m
                                parchk_m
                                                param_m
                                                inpdat_m
                                                give_m
abco_m
                param_m
                inpdat_m
                modat_m
                addlin_m
                xintpl_m
                                types_m
                                inpdat_m
                                xparam_m
                                xdata_m
                                recipe_m
                transf_m
                                param_m
                                types_m
                                gascon_m
                                                ckdcoe_m
                linmix_m
                                linpac_m
radtra_m
                param_m
                inpdat_m
                modat_m
                transf_m
                                param_m
                                types_m
                                gascon_m
                                                ckdcoe_m
                ilsfov_m
```

```
                                    param_m
                                    types_m
                                    inpdat_m
iniout_m
                inpdat_m
                modat_m
                outdat_m
offsca_m
                inpdat_m
                modat_m
deallo_m
                inpdat_m
                modat_m
                outdat_m
                transf_m
                                    param_m
                                    types_m
                                    gascon_m
                                                        ckdcoe_m
ifnlte_m
                varsub_m
                inpdat_m
                modat_m
                param_m
                give_m
nltder_m
                inpdat_m
                modat_m
```

KOPRA installation

M. Höpfner

1 Modules

For the basic version of KOPRA the following modules are necessary (in required order of compilation):

precis_m.f90
param_m.f90
co2coe_m.f90
ckdcoe_m.f90
types_m.f90
inpdat_m.f90
outdat_m.f90
modat_m.f90
varsub_m.f90
recipe_m.f90
give_m.f90
parchk_m.f90
inismw_m.f90
inider_m.f90
inipar_m.f90
inspec_m.f90
xinput_m.f90
input_m.f90
ray_m.f90
rayctl_m.f90
modgeo_m.f90
modlev_m.f90
linpac_m.f90
linmix_m.f90
addlin_m.f90
gascon_m.f90
transf_m.f90
xintpl_m.f90
ilsfov_m.f90
abco_m.f90
radtra_m.f90
offsca_m.f90
iniout_m.f90
wriout_m.f90
deallo_m.f90
kopfwd_m.f90
kopra.f90 (main program)

2 Run KOPRA

KOPRA reads its main input from standard input. The command line to start the program is:

```
kopra < filename_main_input_file
```

The file names of further input and the output are given in the main input file which is described in the following.

3 Required files

Basic principle:
For most of the input files: input will be read beginning directly in the line following a $ in the first column and ending when all the relevant input for this topic is read. No fixed format is required for this input. Until the next $ there can be any number of comment lines.

3.1 Main input file

This is the basic input file for kopra. All other file names and directories are read from this file. All the topics $ are indicated also by a number which is, however, not used by kopra and only meant to facilitate referencing. In the following a commented main input file is given:

```
============================================================
============================================================
KOPRA main input file
version 9-3-99


============================================================
0. version of input file
------------------------

This is the version number of the kopra input-file and identical to the second
number in the kopra version number. Kopra is downward compatible and supports
all lower input file versions.

$0.1
  0.8

============================================================
1. input files and directories for fixed data
---------------------------------------------

1.1 and 1.2 are fixed data used for the calculation of absorption coefficients
in module addlin_m.

File with Doppler cut-offs for calculation of the absorption coefficient
$1.1
input/data/cutdop.dat

File with Lorentz cut-offs for calculation of the absorption coefficient
$1.2
input/data/cutlor.dat

Data file with information on each hitran molecule and isotope
  (number of isotopes, molecular weights, hitran isotope abundancies,
  Gamache coefficient for partition sum calculation, state energies)
$1.3
input/data/hitmol.dat

File with line-mixing data. This file is only necessary if $12.1 is set
```

```
to 'T', i.e. line mixing is switched on.
  (y-coefficients, ...)
$1.4
input/data/linemix.dat

Directory with spectroscopic data
  (The files have to be in the form: fXXXXX, where XXXXX is the wavenumber;
  they are produced with the program 'hitsort.f90')
$1.5
/home/amelib/data/kopra/hitran96/

Directory with cross-section data for the heavy molecules.
This file is only necessary if cross-section species are calculated.
$1.6
/home/amelib/data/cross/


============================================================
2. input files and directories for atmospheric/instrumental data
------------------------------------------------------------

File with p,T profiles.
$2.1
input/profiles/pt.prf

File with p,T - gradient profiles.
  (will only be used if p or T gradients are switched on in $5.9)
$2.2
input/profiles/ptgra.prf

File with vmr profiles
$2.3
input/profiles/vmr.prf

File with vmr - gradient profiles
  (will only be used if vmr gradients are switched on in $5.9)
$2.4
input/profiles/vmrgra.prf

File with continuum profiles
  (for each microwindow one profile)
$2.5
input/profiles/cont.prf

File with vibrational temperature profiles
  (will only be used if nlte is switched on in $13.1)
$2.6
input/profiles/Tvib.gabi

File with Tvib gradient profiles
  (will only be used if Tvib gradients are switched on in $5.9)
$2.7
input/profiles/Tvibgra.prf
```

```
File with isotope abundance profiles
$2.8
input/profiles/isoabu.prf

File with (apodized) instrumental line shape per microwindow
  (will only be used if $9.7==3)
$2.9
input/instrument/ails.dat

File with broadband-aerosol profiles:
  number densities, mode-radii, mode-widths
  (will only be read if $14.1==T)
$2.10
input/profiles/brdaero.prf

File with broadband-aerosol altitude and wavenumber dependent refraction indices
  (will only be read if $14.1==T and $14.2==T)
$2.11
input/profiles/brdaero.refra

File with broadband-aerosol altitude and wavenumber dependent cross-sections
  (will only be read if $14.1==T and $14.2==F)
$2.12
input/profiles/brdaero.cross11


============================================================
3. output files and directories
------------------------------

Spectra on fine grid
  (directory and first letters of output file;
  numbers for microwindows and geometries will be added;
  type  F  for no output; if end letter is  e  the spectral will
  be on the equidistant fine grid, else on the non-equidistant)
$3.1
output/x

Spectra on coarse grid
  (directory and first letters of output file;
  numbers for microwindows and geometries will be added;
  type  F  for no output)
$3.2
output/y

Test-output grade for logfile
  0 = no log-output
  1 = total log-output
$3.3
  1

Filename for logfile
  screen = output to screen
$3.4
```

```
screen

Filename for error-file
  screen = output to screen
$3.5
output/errors


============================================================
4. header for output files [a80]
------------------------------

 (the version number will be added to this header)
$4.1
   test run 25-04-2000


============================================================
5. mode of observation and definition of geometry
------------------------------------------------

1= satellite / limb / tangent altitude
2= satellite / limb / nadir angle and observer altitude
3= balloon / limb / tangent altitude and observer altitude
4= balloon / limb / nadir angle and observer altitude
5= upward / nadir angle and observer altitude
6= upward+limb / nadir angle and observer altitude
7= homogeneous path (cuvette)
$5.1
   1

Background temperature [K]
  (for negative values only the transmission is calculated)
  {default: 2.7 K for emission, 6000 K for solar absorption}
$5.2
   2.7

Number of observational geometries ('sweep')
  *** implies changes in $5.3-$5.7, $8.3, $8.4, $9.11, $9.12 ***
  (for observation mode 1-6)
$5.3
   2

Tangent altitudes [km]         (observation mode 1,3)
Viewing nadir angles [rad]     (observation mode 2,4,5,6)
Homogeneous path length [km]   (observation mode 7)
  (# observation geometries)
  -->   (ordering: low -> high observation geometry)
$5.4
  10.0 13.0

Observer altitude [km]
  (# observation geometries)
  -->   (ordering: low -> high observation geometry)
```

```
$5.5
  800.0 800.0

Latitude of the tangent points [rad]   (observation mode 1,3)
Latitude of the observer [rad]         (observation mode 2,4,5,6)
  (# observation geometries)
  -->   (ordering: low -> high observation geometry)
$5.6
   0.78539 0.78539

Longitude of the tangent points [rad]  (observation mode 1,3)
Longitude of the observer [rad]        (observation mode 2,4,5,6)
  (# observation geometries)
  -->   (ordering: low -> high observation geometry)
$5.7
     0.0 0.0

Azimuth of the observation [rad]
  (i.e. the direction in which the instrument is looking)
  (south=0,direction S->O->N->W)
  (# observation geometries)
  -->   (ordering: low -> high observation geometry)
$5.8
  0.0 0.0

Which horizontal inhomogeneity should be considered?
  (be careful that the related input file is defined in $2.2,$2.4,$2.7)
 p       vmr
 |   T   |  Tvib
 |   |   |   |
$5.9
 F   F   F   F


============================================================
6. wavenumber discretization
----------------------------

Smallest wavenumber interval for calculation of absorption coefficients [cm-1]
  (will internally be adjusted so that $6.2 is an integer multiple of $6.1)
$6.1
   0.0005

Output grid interval [cm-1]
$6.2
   0.025


============================================================
7. parameters for the adjustment of computational accuracy
----------------------------------------------------------

Wavenumber range outside the microwindow where lines are taken into account [cm-1]
  {default 25.0}
```

```
$7.1
    25.00

Accuracy for absorption coefficient calculation
  absolute accuracy for optical              =0 lines my be rejected and truncated
  depth calculation of each line             =1 all lines retained, no truncation
  (negative for slow reference calculation)
              |                                |  {default 0}
$7.2          |                                |
           1.0e-6                              0


Constant base-levels for the forward-model layering:
  the input-profile levels are used exclusively                : 0
  the input-profile levels are used and additional
    levels with respect to criteria $7.3-7.6 are added         : 1
  the levels under $7.32 are used exclusively                  : 2
  the levels under $7.32 are used and additional
    levels with respect to criteria 7.3-7.6 are added          : 3
  the levels are set up automatically with respect             : 4
    to criteria $7.3-7.6
$7.31
   0

Number of base-levels and
Base level altitudes [km]
  (only used if $7.31 = 2,3)
$7.32
  3
  0.0 30.0  120.0


Maximum allowed temperature variations between two forward model levels [K] for
      lower and higher altitude regions and altitude [km] which divides this regions
        |         |                                 |
  (only used if $7.31 = 1,3,4)
$7.3
      10.0      3.0                           40.0

Maximum variation of the Lorentz line width between two model levels
  (values > 1.0)
  (only used if $7.31 = 1,3,4)
$7.4
  1.4

Maximum forward model layer thickness above the tangent altitude [km] and altitude
  above the tangent altitude up to which this maximum thickness is valid [km]
  (only used if $7.31 = 1,3,4)
  {default 100 , 2 to use only input levels}
$7.5
   100.0             2.0

Minimal layer thickness [km]
  (only used if $7.31 = 1,3,4)
```

```
$7.6
   0.4

Definition for the recalculation of the absorption coefficients (extra paths)
  -1 = all absorption coefficients recalculated
   0 = all absorption coefficients above the lowest geometry
       are taken from lowest geometry
   1 = new calculation only of tangent-layer, all others
       are taken from lowest geometry
   2 = new calculation for tangent-layer and the one above, all others
       are taken from lowest geometry
   {default 2}
$7.7
    2

Accuracy of apodization (defines the microwindow extension)
    1 = apodization function decreases to  1%    of center value
    2 = apodization function decreases to  0.1%  of center value
    3 = apodization function decreases to  0.01% of center value
    {default 1}
$7.8
    1

Number of the geometry ('sweep') up to which the $7.10 criterion no. 1 is used
  for addition of simulated geometries for field of view calculation;
  for the geometries above $7.10 criterion no. 2 is used
$7.9
    3

2 Criteria for the addition of simulated geometries for field of view calculation:
    0 = no addition of extra geometries, only one below the lowest observation
        and/or one above the highest
   negative values:
   -1 = between all observed geometries one extra will be added
   -2 = between all observed geometries two extra will be added
   -3 = ...
   positive values (only odd values >= 3 allowed !)
    3 = for fov calculation of each observed geometry 3 geometries at least will be
        used
    5 = for fov calculation of each observed geometry 5 geometries at least will be
        used
    7 = ...

   crit. 1 for low geometries      crit. 2 for high geometries
   {default  0  0   for no extra geometries}
$7.10
         3                               3

Integration step length for ray-tracing and Curtis-Godson/mass calculation [km]
  {default 1.0}
$7.11
        1.000

Gas/isotope number of the main gas
```

```
  (it must appear in all microwindows under $8.6)
 >0 = gas/isotope number of the main gas: it's Curtis-Godson T will be used for
      Planck function
  0 = for no main gas (mix Curtis-Godson T's of all species for Planck function)
      (this is necessary for nlte calculations and is the slowest choice)
 -1 = use Curtis-Godson T of air for Planck function
$7.12
        -1


================================================================
8. microwindow definition
-------------------------

Number of microwindows
  *** implies changes in $8.2-$8.6,$9.10-9.12 ***
$8.1
   1

Wavenumber ranges [cm-1] (ordering low -> high first wavenumbers)
  *** implies changes in $8.3 ***
  --> mw
$8.2
  (940.,941.)

Number of sub-microwindows for each microwindow and
  each observational geometry
  (if = -1: the mircowindows from 8.2 will be used for all altitudes and
   input $8.4 is not valid)
  --> mw
 g
 e
 o
$8.3
  1
  2

Wavenumber range [cm-1] for each sub-microwindow and
  each observational geometry  (must be inside ranges of $8.2)
  --> sub-mw
 g
 e
 o
$8.4
  (940.,941.)
  (940.,940.5) (940.6,940.8)

Number of gases/isotopes to be considered in each microwindow
  *** implies changes in $8.6 ***
  --> mw
$8.5
   2

Gases/isotopes (i4: xxxy, xxx=molecule number, y=hitran isotope number,
```

```
                y=0: use all isotopes)
  (e.g. for separate calculation of HDO and all other isotopes give: 011 010 )

 (line-data gases:)
 1=H2O    2=CO2    3=O3     4=N2O    5=CO     6=CH4    7=O2     8=NO     9=SO2
10=NO2   11=NH3   12=HNO3  13=OH    14=HF    15=HCL   16=HBr   17=HI    18=CLO
19=OCS   20=H2CO  21=HOCL  22=N2    23=HCN   24=CH3CL 25=H2O2  26=C2H2  27=C2H6
28=PH3   29=COF2  30=SF6   31=H2S   32=HCOOH 33=HO2   34=O     35=ClNO3 36=NO+
37=HOBr  38=C2H4

 (additional non-hitran line-data gases:)
101=HNO4

 (pseudo-line data gases:)


 (x-section gases:)
351=CFC-11(CCL3F)            352=CFC-12(CCl2F2)            353=CFC-22(CHClF2)
354=SF6                      355=CFC-14(CF4)               356=CCl4
357=CFC-113(CCl2FCClF2)      358=CFC-114(CClF2CClF2)       359=N2O5
360=ClONO2

  --> gas
 m
 w
$8.6
   10 20

Spectral shift of lines for each gas/isotope
  --> gas
 m
 w
$8.7
 0.0 0.0

continuum is considered for (but only if the related gas number is chosen in $8.6)
artificial continuum (as defined for each mw in file $2.5)
(default=T)

                 N2        O2        H2O       CO2    artificial
$8.8             |         |         |         |         |
                 F         F         T         T         T

============================================================
9. instrumental parameters
--------------------------

Kind of apodization
  1= boxcar in ifg == sinc in spectrum
  2= triangle in ifg == sinc^2 in spectrum
  3= Hamming (Happ-Genzel)
  4= 3 term Blackmann-Harris
  5= 4 term Blackmann-Harris
  6= Norton-Beer weak
```

```
  7= Norton-Beer medium
  8= Norton-Beer strong
    (for MIPAS ENVISAT (i.e. $9.7 =2) only Norton-Beer strong allowed)
$9.1
   8

Maximum optical path difference of the interferometer [cm]
$9.2
   20.0

Should field-of-view be taken into account?
  *** implies changes in $7.9, $7.10 ***
$9.3
   T

Should field-of-view effect on the instrumental profile be taken into account (F/T)?
$9.4
   F

Number of channels (the microwindows must lie in these channels)
$9.43
  1

Wavenumber range [cm-1] for each channel of $9.43
$9.46
  (100.,6000.)

Half vertical field-of-view extension [rad]
  (one number for each channel of $9.43)
$9.5
   6.1e-4

Relative weighting function for vertical field-of-view
  (20 values along vertical fov extension, ordered by elevation,
  the vertical fov is subdivided in 20 bands of equal thickness
  and the values give the relative weighting (integral!) of each band)
  (one function for each channel of $9.43)
$9.6
  1.,3.,5.,6.,6.,6.,6.,6.,6.,6.,6.,6.,6.,6.,6.,5.,3.,1.

Ils is calculated for circular aperture with phase and lin. apo. error:  1
Ils is calculated using ESA parametrization:                            2
Ils is read in for each microwindow from file($2.9):                    3
  (in case 3 the area under the ils must be normalized to 1!)
$9.7
  1

In case of $9.7 = 1:
====================
->maximum inclination of the ray in the interferometer [rad]
   [max. value = 1/sqrt({$9.2} * max_wavenumber_of_interferometer) ]
->part of linear Apodization in interferogram
   (1.0 = perfect instrument, 0.0 = triangle) {default 1.0}
->phase error [rad] {default 0.0}
```

```
In case of $9.7 = 2:
====================
 ESA parameters:
->infrared misalignment y-direction [rad]
->infrared misalignment z-direction [rad]
->retroreflector linear shear along y [cm]
->retroreflector linear shear along z [cm]
->linear shear variation along y [-]
->linear shear variation along z [-]
->full interferometer divergence along y [rad]
->full interferometer divergence along z [rad]
->blur angular width along y [rad]
->blur angular width along z [rad]
->laser misalignment along y [rad]
->laser misalignment along z [rad]
->optical speed of interferometer [cm/s]
->initial sampling perturbation [dxnull]
->time constant of exponential attenuation of initial sampling perturbation [s]
->time constant of exponential attenuation of initial speed fluctuation [s]
->initial relative speed fluctuation at beginning of scan [cm/s]
->gain slope of IR electrical response [-]
->mismatched delay between electronic response and ADC trigger [s]
->laser wvnr [cm-1]
->relative drift rate of laser wvnr [-]
->bandwidth laser white noise [Hz]
->bandwidth laser 1/f noise [Hz]

$9.8
   4.6e-5  1.0  0.0

   0.0, 0.2e-3, 0.004, 0.004, 0.0, 0.0, 5.4e-3, 9.0e-3
   5.2e-4, 3.3e-4, 0.15e-3, 0.15e-3, 10.0, 4.0e-8, 0.16, 0.016, 0.03
   -0.22, 1.4e-6, 7692.0, 1.0e-8, 20.0e6, 0.0

Wavenumber shift [cm-1]
   --> m i c r o w i n d o w
$9.9
   0.0 0.0

Offset per mw and geometry [nW/cm2Srcm-1]  (Spectrum*scale + offset)
  *** implies changes in $8.1, $5.2 ***
   --> m i c r o w i n d o w
 g
 e
 o
$9.10
   0.0 0.0

Scale per mw and geometry [nW/cm2Srcm-1]   (Spectrum*scale + offset)
  *** implies changes in $8.1, $5.2 ***
   --> m i c r o w i n d o w
 g
 e
```

```
 o
$9.11
   1.0 1.0

Should noise be added to the spectra?
$9.12
  F

Rms value of the noise in the sinc-spectrum
 (independent grid-values of distance 1/2*opdmax) [nW/cm2Srcm-1]
 --> mw
$9.13
 30.0 30.0

Seed-value for noise
 ( >= 0: same noise in two runs, < 0: noise always changes)
$9.14
   -1

============================================================
10. derivative definition
-------------------------

Which derivatives shall be calculated?
  in case of esa-ils-parameters ($9.7==2):
    phase is parameter 1 and apolin is parameter 2

Temp  artific. shift  offset   phase   Tgrad   press    aerosol         weighting
 |    continuum  |       |       |       |       | nlte numden          fuction
 |  vmr  |  LOS  |  Tvib | scale | apolin|vmrgrad| model | Mie-para     |
 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |    |         |
$10.1
 T   T   F   F   F   F   F   F   F   F   F   F   F   F   F    F         F

How many gases/isotopes for VMR derivative?
  *** implies changes in $10.3 ***
$10.2
    1

Which gas/isotope for VMR derivative?
  --->
$10.3
   010

How many gases/isotopes for VMR-gradient derivative?
  *** implies changes in $10.5 ***
$10.4
    1

Which gas/isotope for VMR-gradient derivative?
  --->
$10.5
   020
```

```
Which gas/isotope (nlte-species) for Tvib derivative?
$10.6
   021

How many vibrational temperature profiles (states) for this gas/isotope?
  *** implies changes in $10.8 ***
$10.7
   2

Which Tvib-state(s) (HITRAN state number)?
  --->
$10.8
   1 2


============================================================
11. profile definition
----------------------

Should the profiles be recalculated for hydrostatic equilibrium
  (no:0) and should the level-pressures (1) or the level-altitudes (2) be adjusted?
  {default  0}
$11.1
   0

Number of base levels for p,T,vmr,Tvib,continuum and gradient profiles
$11.2
   50

Latitude of the profiles read from files $2.1-2.7 [rad]
$11.3
   0.78539

Longitude of the profiles read from files $2.1-2.7 [rad]
$11.4
   0.0

Level altitudes [km]
$11.5
       0.0,       1.0,       2.0,       3.0,       4.0,
       5.0,       6.0,       7.0,       8.0,       9.0,
      10.0,      11.0,      12.0,      13.0,      14.0,
      15.0,      16.0,      17.0,      18.0,      19.0,
      20.0,      21.0,      22.0,      23.0,      24.0,
      25.0,      27.5,      30.0,      32.5,      35.0,
      37.5,      40.0,      42.5,      45.0,      47.5,
      50.0,      55.0,      60.0,      65.0,      70.0,
      75.0,      80.0,      85.0,      90.0,      95.0,
     100.0,     105.0,     110.0,     115.0,     120.0


============================================================
12. line mixing
```

```
--------------

Should line mixing be considered (F/T)?
$12.1
  F

Direct diagonalization (1), Rosenkranz (2)
$12.2
  2

Only Q- (1), Q- and PR- coupling (2)
$12.3
  2

============================================================
13. NLTE
--------

Should NLTE be considered (F/T)?
$13.1
  F

Should fq be set to 1.0?
  {default  T}
$13.2
  T

Should rotational-spin nlte for NO be calculated (F/T)?
  {default  F}
$13.25
  F

NLTE for how many different species?
   *** implies changes in 13.6, 13.8 ***
$13.3
  1

Species identifier(s) [(hitran molecular code) *10 + (hitran isotope number)]
  -->
$13.4
   021

How many Tvib profiles (states) for each species?
   *** implies changes in 13.6 ***
  -->
$13.5
  2

Which Tvib-states for each species?
  states -->
 s
 p
 e
 c
```

```
 i
$13.6
  5 9

How many transitions should be calculated for each species?
   *** implies changes in 13.8 ***
  -->
$13.7
   1

Which transitions (ground state =1)?
 (higher,lower state)

  transitions -->
 s
 p
 e
 c
 i

$13.8
   (9,5)

Should the nlte-model be used (F/T)?
$13.9
   F

Nlte-model for which nlte-species?
   *** must be one of 13.4 ***
$13.10
   021


================================================================================
14. broadband-aerosol
---------------------

Should broadband-aerosol be considered?
   *** input files $2.10 and $2.11 or $2.12 must exist ***
$14.1
  F

Should the KOPRA internal Mie-model be used?
   (=T: wavenumber and altitude dependent refraction indices must be provided: $2.11)
   (=F: wavenumber and altitude dependent cross-sections must be provided: $2.12)
$14.2
  T

============================================================
end KOPRA main input file



============================================================
```

```
============================================================
```

3.2 Files with fixed data

The fixed data files are delivered together with the source-files of KOPRA. The directory tree is not fixed but given under $1.* in the main input file.

3.2.1 Files with Doppler and Lorentz cut-offs

```
============================================================

   Files with Doppler ($1.1) and Lorentz ($1.2) cut-offs for calculation of the
   absorption coefficient. There are two versions of each file:
   For normal accuracy calculations:
     cutdop.dat  and cutlor.dat
   For very high accuracy:
     cutdop3.dat and cutlor3.dat
============================================================
```

3.2.2 Molecule information data file

```
============================================================

   Data file with information on each hitran molecule and isotope
   (number of isotopes, molecular weights, hitran isotope abundancies,
   Gamache coefficient for partition sum calculation, state energies)
   ($3.3)
============================================================


$ number of hitran molecules, additional molecules, pseudo-line molecules
  38, 3, 10

$  number of isotopes per HITRAN molecule
  4,8,5,5,6,3,3,3,2,1,2,1,3,1,2,2,1,2,5,3,2,1,3,2,1,2,1,1,1,1,3,1,1,1,2,1,2,2

$  number of isotopes per additional line gas molecule
   1,1,1

$  number of isotopes per pseudo line gas molecule
   1,1,1,1,1,1,1,1,1,2

$  molecular weights per HITRAN molecule
   18., 20., 19., 19.
   44., 45., 46., 45., 47., 46.,48.,47.
   48., 50., 50., 49., 49.
   44., 45., 45., 46., 45.
   28., 29., 30., 29., 31., 30.
   16., 17., 17.
   32., 34., 33.
   30., 31., 32.
   64., 66.
```

```
   46.
   17., 18.
   63.
   17., 19., 18.
   20.
...
...

$  molecular weights per additional gas molecule, to be VERIFIED
   79.
   44.
   30.

$  molecular weights per pseudo-line gas molecule
   1.
   1.
   1.
   1.
   8.
   1.
   0.0004
   0.
   0.001
   97. 99.

$  isotope abundancies of HITRAN
  0.997317,0.00199983,0.000372,0.00031069
  0.98420,0.01106,0.0039471,0.000734,0.00004434,0.00000825,0.0000039573,0.00000147
  0.992901,0.00398194,0.00199097,0.000740,0.000370
  0.990333,0.0036409,0.0036409,0.00198582,0.000369
  0.98654,0.01108,0.0019782,0.000368,0.00002222,0.00000413
  0.98827,0.01110,0.00061575
  0.995262,0.00399141,0.000742
  0.993974,0.0036543,0.00199312
  0.94568,0.04195
  0.991616
  0.9958715,0.0036613
  0.989110
  0.997473,0.00200014,0.00015537
  0.99984425
  0.757587,0.242257
  0.50678,0.49306
  0.99984425
  0.75591,0.24172
  0.93739,0.04158,0.01053,0.00739908,0.0018797
  0.98624,0.01108,0.0019776
  0.75579,0.24168
  0.9926874
  0.98511,0.01107,0.0036217
  0.74894,0.23949
...
...

$  isotope abundancies of additional line gases
```

```
  1.0
  1.0
  1.0

$  isotope abundancies of pseudo-line gases
  1.0
  1.0
  1.0
...
...

  Total internal partition sums for 70 - 500 K and
  500 - 1500 K coefficients for temperature interpolation
  for hitran molecules
$
 !...   H2O  --   161
  -.44405E+01, .27678E+00, .12536E-02,-.48938E-06
  -.94327E+02, .81903E+00, .74005E-04, .42437E-06
 !...   H2O  --   181
  -.43624E+01, .27647E+00, .12802E-02,-.52046E-06
  -.95686E+02, .82839E+00, .68311E-04, .42985E-06
 !...   H2O  --   171
  -.25767E+02, .16458E+01, .76905E-02,-.31668E-05
  -.57133E+03, .49480E+01, .41517E-03, .25599E-05
...
...

  Total internal partition sums for 70 - 500 K and
  500 - 1500 K coefficients for temperature interpolation
  for additional molecules
$
 !...HNO4  --  146                              ! derived from Toon
    -93323.15,1501.75154,-7.63597,0.01955       !160-350 K
    -1.36049E6, 12156.008, -37.79588,0.0483     !350-500 K
 !...C3H8  --  221
  .10000E+01, .0000E+00,.0000E+00, .0000E+00
  .10000E+01, .0000E+00,.0000E+00, .0000E+00
 !...  H2CO  --   126, taken from hitran-gas 20 (H2CO)
  -.11760E+03, .46885E+01,.15088E-01, .35367E-05
  -.11760E+03, .46885E+01,.15088E-01, .35367E-05


  Total internal partition sums for 70 - 500 K and
  500 - 1500 K coefficients for temperature interpolation
  for pseudo-line gas molecules
$
 !...ccl3f  f11                                 ! from Toon
    -0.3529, 1.00617, -3.15636E-5, 4.90591E-8
    32.7266, 0.89325, 5.73225E-5, 6.49571E-8
 !...ccl2f2  f12                                ! from Toon
    2.70287 0.99566, -5.51294E-4, 3.26664E-6
   -1330.38301, 6.58174, -0.00826, 6.78855E-6
 !...chf2cl  f22                                ! from Toon
    -1.2063, 1.02228, -1.23158E-4, 2.11691E-7
```

```
    62.86447, 0.74045, 2.71348E-4, 4.48698E-8
...
...

   total internal partition sums at reference temperature 296 K
   for hitran molecules
$
  .174626E+03, .176141E+03, .105306E+04, .865122E+03
  .286219E+03, .576928E+03, .607978E+03, .354389E+04, .123528E+04, .714432E+04,
  .323407E+03, .376700E+04
  .348186E+04, .746207E+04, .364563E+04, .430647E+05, .212791E+05
  .499183E+04, .334938E+04, .344940E+04, .526595E+04, .307008E+05
  .107428E+03, .224704E+03, .112781E+03, .661209E+03, .236447E+03, .138071E+04
  .589908E+03, .117974E+04, .477061E+04
  .215726E+03, .452188E+03, .263998E+04
  .339730E+04, .157040E+04, .358252E+04
...
...

   total internal partition sums at reference temperature 296 K
   for additional molecules
$
    189178.416             ! HNO4 derived from Toon
   .100000E+01             ! C3H8
   .268388E+04             ! taken from hitran-gas 20 (H2CO)

   total internal partition sums at reference temperature 296 K
   for pseudo-line gas molecules
$
   2.959803E+02
   3.338342E+02
   2.960880E+02
   2.959858E+02
...
...

   State Energies  for hitran gases
     number of states
     state energies
     number of ground states
     hitran ground state number
$
 !...   H2O  --   161
   4
   0.0,1594.750,3151.630,3657.053
   1
   1
 !...   H2O  --   181
   2
   0.0,1588.279
   1
   1
 !...   H2O  --   171
...
```

```
...

   State Energies  for additional line-gases
     number of states
     state energies
     number of ground states
      ground state number
$
!...HNO4  --  146
   1
   0.
   1
   1
!...C3H8   --  221
...
...

   State Energies  for pseudo line-gases
     number of states
     state energies
     number of ground states
      ground state number
$
   !...ccl3f
   1
   0.
   1
   1
   !...ccl2f2
...
...
================================================================
```

3.2.3 Line-mixing data file

For a detailed description of this file see Part VI: 'Line mixing'.

```
================================================================
================================================================

   File with line-mixing data. This file is only necessary if $12.1 is set
   to 'T', i.e. line mixing is switched on.
   (y-coefficients, ...) ($3.4)

================================================================
```

3.2.4 Spectroscopic data files

Files with spectroscopic data of HITRAN molecules, additional molecules, and pseudo-lines. (The file-names have to be in the form: fXXXXX, where XXXXX is the wavenumber; the files are produced with the auxiliary program 'hitsort.f90') ($3.5)

3.2.5 Cross-section data files for heavy molecules

Files with cross-section data for the heavy molecules. (Only necessary if cross-section species are calculated.) For each molecule one binary direct-access data file is necessary and is produced with the auxiliary program 'hitX2bin.f90' from standard hitran cross-section files. The cross-section file-names are given in the file xfiles.dat which is in the same directory as the cross-section files (name given under $3.6).

3.3 Input files for atmospheric profiles

The altitudes of the profiles and the number of levels can be different from file to file. All profiles will be interpolated to the grid under $11.5 in the main input file. Be careful: if the altitude range of $11.5 extends the altitude ranges in the profile-files the values will also be extrapolated to the levels of $11.5! This can sometimes lead to implausible values. So care should be taken that the altitude range of $11.5 includes all ranges given in the profiles.
All input-profiles are assumed to be at the same geolocation which is given under $11.3-11.4 in the main input file.

3.3.1 p,T profiles

```
============================================================

   File with p,T profiles. The filename is given under $2.1
   in the main input file.
============================================================

Number of levels
$
  50


Altitude [km]
$
    0.0,       1.0,       2.0,       3.0,       4.0,
    ...

Pressure [hPa]
$
  1.013E+03, 8.988E+02, 7.950E+02, 7.012E+02, 6.166E+02,
  ...

Temperature [K]
$
    288.20,    281.70,    275.20,    268.70,    262.20,
    ...
============================================================
```

3.3.2 p,T gradient profiles

```
============================================================

   File with p,T gradient profiles. The filename is given
   under $2.2 in the main input file.
```

```
================================================================
Number of levels
$
 50

Level altitudes [km]
$
       0.0,       1.0,       2.0,       3.0,       4.0,
     ...

Pressure lat. gradient [hPa/km], i.e. the gradient is parallel to
  (along) the latitude circles.
  Positive to the east and negative to the west.
$
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  ...

Pressure lon. gradient [hPa/km], i.e. the gradient is parallel to
  (along) the longitude circles (meridians).
  Positive to the south and negative to the north.
$
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  ...


Temperature lat. gradient [K/km], i.e. the gradient is parallel to
  (along) the latitude circles.
  Positive to the east and negative to the west.
$
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  ...

Temperature lon. gradient [K/km], i.e. the gradient is parallel to
  (along) the longitude circles (meridians).
  Positive to the south and negative to the north.
$
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  ...
================================================================
```

3.3.3 vmr profiles

```
================================================================

   File with vmr profiles. The filename is given under $2.3
   in the main input file.
   There must exist a profile for each molecule which
   should be calculated under $8.6 of the main input file.
================================================================
Number of levels
$
  50

Level altitudes [km]
```

```
$
       0.0,       1.0,       2.0,       3.0,       4.0,
       ...

Number of gas profiles given below (at least for the gases which have to be
considered in all microwindows)
$
  2

Hitran gas number
Profile [ppmv]

H2O
$
   1
 7.745E+03 6.071E+03 4.631E+03 3.182E+03 2.158E+03 1.397E+03 9.254E+02 5.720E+02
 ...
CO2
$
   2
 3.300E+02 3.300E+02 3.300E+02 3.300E+02 3.300E+02 3.300E+02 3.300E+02 3.300E+02
 ...


============================================================
```

3.3.4 vmr gradient profiles

```
============================================================

   File with vmr - gradient profiles. The filename is given under $2.4 in
   main input file.
   There must exist a profile for each molecule which
   should be calculated under $8.6 of the main input file
   (even if its gradients are zero).
============================================================
Number of levels
$
10

Level altitudes [km]
$
0.0,   9.0,      16.0,      24.0,      31.0,      39.0,
      48.0,      57.0,      66.0,     120.0


Number of gas gradient profile pairs given below
 (at least for the gases which have to be
  considered in all microwindows)
$
 2

Hitran gas number
```

```
Profile [ppmv/km]

H2O   lat. gradient [ppmv/km], i.e. the gradient is parallel to
  (along) the latitude circles.
  Positive to the east and negative to the west.
$
   1
 3.375E-01 3.375E-01 4.050E-03 2.400E-03 2.000E-03 1.500E-03
 1.800E-03 2.200E-03 2.200E-03 2.200E-03

H2O   lon. gradient [ppmv/km], i.e. the gradient is parallel to
  (along) the longitude circles (meridians).
  Positive to the south and negative to the north.
$
   1
 3.375E-01 3.375E-01 4.050E-03 2.400E-03 2.000E-03 1.500E-03
 1.800E-03 2.200E-03 2.200E-03 2.200E-03

CO2   lat. gradient [ppmv/km], i.e. the gradient is parallel to
  (along) the latitude circles.
  Positive to the east and negative to the west.
$
   2
 0.000E-02 0.000E-02 0.000E-02 0.000E-02 0.000E-02
 0.000E-02 0.000E-02 0.000E-02 0.000E-02 0.000E-02

CO2   lon. gradient [ppmv/km], i.e. the gradient is parallel to
  (along) the longitude circles (meridians).
  Positive to the south and negative to the north.
$
   2
 0.000E-02 0.000E-02 0.000E-02 0.000E-02 0.000E-02
 0.000E-02 0.000E-02 0.000E-02 0.000E-02 0.000E-02
============================================================
```

3.3.5 continuum profiles

```
============================================================

   File with continuum profiles. The filename is given under $2.5 in
   main input file. (For each microwindow one profile has to be given.)
============================================================
aerosol absorption and scattering
coefficient profiles (for each microwindow) [km-1]


Number of levels
$
  50

Level altitudes [km]
$
```

```
      0.0,       1.0,       2.0,       3.0,       4.0,
     ...

Number of microwindows for which profiles are given below
$
 2

Continuum absorption coefficient profiles (for each microwindow) [km-1]

Microwindow number
Profile

$
 1
 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 ...

$
 2
 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 ...


Continuum scattering coefficient profiles (for each microwindow) [km-1]

Microwindow number
Profile

$
 1
 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 ...

$
 2
 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 ...

================================================================
```

3.3.6 Vibrational temperature or non-LTE/LTE population ratio profiles

```
================================================================

   File with vibrational temperature or population ratio profiles.
   The filename is given under $2.6 in
   main input file.
================================================================

Number of levels
$
```

```
  50

Level altitudes [km]
$
       0.0,       1.0,       2.0,       3.0,       4.0,
     ...

Vibrational temperatures (F),  population ratios (T)
$
  F

Total number of following profiles
$
  1

Species
Vibrational state
Vibrational temperatures or population ratios

co2  2 (vib state 01101)
$
 021
 2
 2.8820E+02 2.8170E+02 2.7520E+02 2.6870E+02 2.6220E+02 2.5570E+02 2.4920E+02
 ...

============================================================
```

3.3.7 Vibrational temperature or non-LTE/LTE population ratio gradient profiles

```
============================================================

   File with vibrational temperature or population ratio gradient profiles.
   The filename is given under $2.7 in
   main input file.
============================================================
Number of levels
$
  50

Level altitudes [km]
$
       0.0,       1.0,       2.0,       3.0,       4.0,
     ...

Vibrational temperatures (F),  population ratios (T)
$
  F

Total number of following latitude/longitude profile-pairs
$
```

```
  1

Species
Vibrational state
Vibrational temperature gradients

co2  2 (vib state 01101) lat. gradient [K(or 1)/km], i.e. the gradient is parallel to
  (along) the latitude circles.
  Positive to the east and negative to the west.
$
 021
 2
 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2
  ...
co2  2 (vib state 01101)  lon. gradient [K(or 1)/km], i.e. the gradient is parallel to
  (along) the longitude circles (meridians).
  Positive to the south and negative to the north.
$
 021
 2
 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2 1.000E-2
 ...
============================================================
```

3.3.8 Isotope abundance profiles

```
============================================================

   File with isotope abundance profiles.
   The filename is given under $2.8 in
   main input file.
============================================================


The isotope (relative) abundance profiles for all isotopes of all hitran gases:

Number of levels for gas: 1
$
 1

Altitude levels for this gas [km]
$
 1.0

Isotope abundances on the levels
$ iso 1
   0.997317
$ iso 2
   0.00199983
$ iso 3
   0.000372
$ iso 4
```

```
   0.00031069

Number of levels for gas: 2
$
 1

 ...
 ...
============================================================
```

3.3.9 Broadband aerosol number density, mode-radius and -width profiles

```
============================================================

  File with broadband-aerosol profiles:
  number densities, mode-radii, mode-widths.
  The filename is given under $2.10
  in the main input file.
============================================================

Number of altitude levels
$
  4

Altitude [km]
$
  0.0  22.0  25.0  120.0

Broadband-aerosol number density [particles/cm-3]
$
  0.0  10.0  0.0 0.0


The following input data is only necessary if the KOPRA internal Mie-model is used:

Number of modes for distribution function (1 or 2)
$
  1

Mode-radius for mode 1 [micrometer]
$
  0.2   0.2  0.2  0.2

Mode-width for mode 1
$
  0.4   0.4  0.4  0.4



The following input data is only necessary if 2 modes are considered:

Mode-radius for mode 2 [micrometer]
$
```

```
   0.9   0.9  0.9  0.9

Mode-width for mode 2
$
  0.4 0.4 0.4 0.4

Weight for mode 1
 (range: 0.0 - 1.0)
$
  1.0
================================================================
```

3.3.10 Broadband aerosol altitude and wavenumber dependent refraction indices

```
================================================================

  File with broadband-aerosol altitude and wavenumber
  dependent refraction indices.
  The filename is given under $2.11
  in the main input file.
================================================================

Number of altitude levels
$
  2

Altitude [km]
$
  0.0  120.0



The following section is repeated (number of levels) times:

Number of wavenumber grid points for refraction indices:
$
  1691

Wavenumber, real-part, imaginary part of refraction index
$
   482.12 2.172E+00 3.333E-01
   485.97 2.121E+00 4.082E-01
   489.83 2.077E+00 4.928E-01
   493.69 2.010E+00 5.676E-01
   497.54 1.917E+00 6.039E-01
    ...

Number of wavenumber grid points for refraction indices:
$
  1691

Wavenumber, real-part, imaginary part of refraction index
$
```

```
   482.12 2.172E+00 3.333E-01
   485.97 2.121E+00 4.082E-01
   489.83 2.077E+00 4.928E-01
   493.69 2.010E+00 5.676E-01
   497.54 1.917E+00 6.039E-01
    ...

================================================================
```

3.3.11 Broadband aerosol altitude and wavenumber dependent cross-sections

```
================================================================

  File with broadband-aerosol altitude and wavenumber
  dependent cross-sections.
  The filename is given under $2.12
  in the main input file.
================================================================
Number of altitude levels
$
  2

Altitude [km]
$
  0.0  120.0


The following section is repeated (number of levels) times:


Number of wavenumber grid points for broadband-aerosol
  cross-sections:
$
  5

Wavenumber [cm-1], absorption cross-section [cm2/particle],
  extinction cross-section [cm2/particle]
$
 500.0  1.0e-17 1.2e-17
 955.0  1.0e-16 1.1e-16
 960.0  1.0e-18 1.2e-18
 1500.0 1.0e-18 1.3e-18
 2000.0 1.0e-18 1.4e-18

Number of wavenumber grid points for broadband-aerosol
  cross-sections:
$
  5

Wavenumber [cm-1], absorption cross-section [cm2/particle],
  extinction cross-section [cm2/particle]
$
```

```
 500.0  1.0e-17 1.2e-17
 955.0  1.0e-16 1.1e-16
 960.0  1.0e-18 1.2e-18
 1500.0 1.0e-18 1.3e-18
 2000.0 1.0e-18 1.4e-18
```

```
============================================================
```

3.4 Input file for AILS-function

```
============================================================

   File with (apodized) instrumental line shape (AILS) function per microwindow
   The filename is given under $2.9 in the main input file.
============================================================

 The apodized instrumental line shape (ails) function for
 each microwindow on the wavenumber fine grid.

Wavenumber distance between the ails points  [cm-1]
$
  0.0005

Number of grid points (must be odd!)
$
  303

Ails functions per microwindow
$  mw 1
    -0.01928
    -0.01878
    -0.01849
   ...

============================================================
```

3.5 Output files for spectra and derivative spectra

The file name convention for spectra and derivative spectra is:
For spectra:
directory/abcd...xxyyzz
For derivative-spectra of offset, scale, wavenumber-shift, ils-phase, ils-apolin, and line-of-sight:
directory/abcd...offxxyyzz
directory/abcd...scaxxyyzz
directory/abcd...shixxyyzz
directory/abcd...phaxxyyzz
directory/abcd...apoxxyyzz
directory/abcd...losxxyyzz
For derivative-spectra of temperature, temperature-gradient, pressure, and continuum:

directory/abcd...Txxyyzzvvv
directory/abcd...Tgraxxyyzzvvv
directory/abcd...pxxyyzzvvv
directory/abcd...aerxxyyzzvvv
For derivative-spectra of vmr, vmr-gradient, and vibrational temperature:
directory/abcd...vmrxxyyzzuuvvv
directory/abcd...vmrgraxxyyzzuuvvv
directory/abcd...Tvibxxyyzzuuvvv
Where:
directory/abcd... = directory and file-initials as given under 3.1*and*3.2 xx = microwindow number yy = observation geometry number zz = sub-microwindow number uu = species number (in case of vmr and vmr-gradient) uu = vibrational state number (in case of Tvib) vvv = (altitude) retrieval-parameter number (e.g. altitude level)

```
================================================================

   Spectra and derivative spectra on the fine wavenumber (monochromatic) grid.
   The file name directory and initials are given under $3.1 and under $3.2
================================================================
#
#     number of wavenumber grid-points
  wavenumber          spectral value [nW/cm2srcm-1]
  wavenumber          spectral value [nW/cm2srcm-1]
  wavenumber          spectral value [nW/cm2srcm-1]
  wavenumber          spectral value [nW/cm2srcm-1]
     ...                    ...
================================================================
```

3.6 log- and error-files

If $3.3==1 the log-file is written to screen or file as selected under $3.4. The warnings and errors are written into the log-file and into an extra error-file with the filename of $3.5.

Module, subroutine and variable listing and description

M. Höpfner

1 Module Listing and Description

- Modules for fixed data and parameters:
  - ckdcoe_m
    Parameters for the calculation of the H_2O-continuum.
    Subroutines:
    -/-
  - param_m
    Definition of mathematical-physical and dimensioning parameters.
    Subroutines:
    -/-
  - precis_m Definition of precision parameters for integer and real variables.
    Subroutines:
    -/-
- Module for type-definition:
  - types_m
    Definition of types for all derived-type variables in kopra.
    Subroutines:
    -/-
- Modules for variable-interfaces:
  - inpdat_m
    Definition of input variables.
    Subroutines:
    -/-
  - modat_m
    Definition of internal model variables.
    Subroutines:
    -/-
  - outdat_m
    Definition of output variables.
    Subroutines:
    -/-
- Modules with mathematical libraries:
  - linpac_m
    Subroutine collection for linear algebra.
    (Translated from the slatec library on http://www.netlib.org/slatec/)
    Subroutines:
    public :: balanc, cbal, balbak, cgedi, cgefa, cginv, corth, elmhes
    eltran, compqr2, hqr, cg, rg, qsort
  - recipe_m
    Subroutine collection from numerical recipes.
    Subroutines:
    public :: polcoe, hunt, locate, spline, splint, splint1, linint1, isort

- Modules for often used subroutines:
  - give_m
    Collection of subroutines which return the value of atmospheric parameters for a given location in the atmosphere (latitude, longitude, altitude). The interpolation is performed either in the input profiles or in the retrieval parameter profiles.
    Subroutines:
    public :: give_p,give_T,give_vmr,give_Tvib, give_aerabs,give_aersca
    private :: give_pgrad, give_Tgrad, give_Tvibgrad, geodistance
  - varsub_m
    Collection of subroutines that are often use by different modules.
    Subroutines:
    public :: delete_element_i, multiple_cut, go_next, g, press, alti, seconds, yinterpol1, yinterpol2
- Modules for specialized tasks:
  - abco_m
    Calculates absorption coefficients [cm^2] for all layers and observation geometries.
    Subroutines:
    public :: absco_calc
    private :: absco_branch, allocate_geo_mw, line_strength, cho2chi
  - addlin_m
    Provides subroutines for the efficient calculation of absorption coefficient on a line-by-line basis.
    The basic idea the use of look-up-tables for determining the optimal set of sampling points for each spectral line.
    Subroutines:
    public :: allocate_cutoff, deallocate_cutoff, allocate_grid, deallocate_grid, add_lines_chilm, add_lines_chi, add_lines_lm, add_lines, interpolate_grid
    private :: voigt
  - gascon_m
    Calculation of N_2,O_2, H_2O, CO_2 continua.
    Subroutines:
    public :: calc_h2ocon, calc_n2con, calc_o2con
  - ifnlte_m
    Interface to the nlte-model
    Subroutines:
    public :: interface_nlte
  - ilsfov_m
    Calculate ails and convolve fine-grid spectrum. Determine field-of-view weighted average spectrum.
    Subroutines:
    public :: envils, envfovils, ilsapo, fovils1, ilsapo_fixed, ilsfov_fixed
    private :: value_test, vertinterpol, fovmeanspec, scndderfov, fovilsspec, derive, convolution, makeifg, makeils, shiftwvnr, makecifg, fovtrm, makecils, addnoise

- inider_m
  Initialization of variable deri% for derivative calculation.
  Subroutines:
  public :: ini_deri

- inipar_m
  Initialization of the vector para% which contains the parameterization of atmospheric profiles.
  Subroutines:
  public :: ini_para
  private :: ini_para_vmr, ini_para_T, ini_para_Tvib, ini_para_aerabs, ini_para_Tgrad, ini_para_vmrgrad, ini_para_p

- inismw_m
  Determination of internal forward model sub-microwindows.
  Subroutines:
  public :: ini_sub_mw

- iniout_m
  Initialization and determination of the output variable outdat%.
  Subroutines:
  public :: ini_output

- input_m
  Initialization and definition of all input variables.
  Subroutines:
  public :: input
  private :: input_hitmol, input_main, input_mwdef, input_isoprof, input_pTprof, input_vmrprof, input_pTgradprof, input_vmrgradprof, input_contprof, input_Tvibprof, input_Tvibgradprof,input_ails, make_hydroequi, ils_radius, extend_mw, speciorder, isomult

- inspec_m
  Read spectroscopic data and built up speci% variable.
  Subroutines:
  public :: input_spectroscopy
  private :: numdata, readlines, isoabun, linein, allocopy1, allocno_nlte, allocno_lm, alloc_nlte, allocopy2, pointmw, check_mwspeci, read_lmdata, alloc_lm, delete_lmlines,

- linmix_m
  Provides subroutines for line-mixing.
  Subroutines:
  public :: y_calc_rk, y_calc_dd, corr_y_coeffs,

- miemod_m
  Performs the Mie-calculations.
  Subroutines:
  public :: mie
  private :: miecoe,mieampli,set_miepar,set_gauher

- modgeo_m
  Introduces additional geometries for the simulation of FOV and defines the new occupation matrix for all the simulated geometries ('simulated geometries' are called all geometries to be calculated in the forward model i.e. the

observed plus the additional geometries for FOV.) Subroutines:
public :: make_modelgeo
private :: addsim_a, addsim_b, addgeo, make_occusim,

- modlev_m
  Defines the model altitude levels for the forward calculation.
  Subroutines:
  public :: make_modelgrid
  private :: min_distance, calc_ztang, base, grid_t_hw, grid_tang, delete_level, add_level,
- nltder_m
  Post derivatives in case of use of nlte-model
  Subroutines:
  public :: nlte_postderi
  private :: nlte_vmr_postderi, nlte_x_postderi
- offsca_m
  Add offset and scale spectra. Calculate offset and scale derivatives.
  Subroutines:
  public :: offset_scale
  private :: sca, off, sca_deri, off_deri
- parchk_m
  Checks if variation of a distinct retrieval parameter influences the values of the atmospheric quantity
  inside an altitude region of the atmosphere.
  Subroutines:
  public :: para_aerabs_change, para_vmr_change, para_T_change, para_Tvib_change, para_Tgrad_change, para_vmrgrad_change
- radtra_m
  Calculation of the radiative transfer through the atmosphere and determination of the fine-grid spectra and their derivatives.
  Convolution of the fine-grid spectra with the ails to get the spectra on the coarse-grid (measurement grid) and
  calculation of the field-of-view weighting.
  Subroutines:
  public :: radtrans
  private :: radtrans_mw, tausrc, optsrcdTvib, derivmr_calc, deriaer_calc, deriT_calc, deriTvib_calc,
  deriTgrad_calc, derivmrgrad_calc, writeabco, writespec, lsimobs, finewrk, ilsapofov_calc, sub_mw_minmax,
  alloc_Sails
- rayctl_m
  Controls ray-tracing, calculation of path integrated values (Curtis-Godson values and column amounts) and the
  derivatives of path integrated values wrt retrieval parameters.
  Subroutines:
  public :: raytrace_ctrl
  private :: allocgeo1, homog_path, para_dcol_ne0, para_dabsopt_ne0, para_dT_ne0, para_dTgrad_ne0,
  para_dTvib_ne0, para_dcolgrad_ne0, calc_nlte_ratios, calc_nlte_dratios
- ray_m Calculate ray-tracing, path integrated values (Curtis-Godson values and column amounts) and the

derivatives of path integrated values wrt retrieval parameters.
Subroutines:
public :: raytra, tangalt
private :: leveltang, leveltrans, integrate, nmax_calc, index_nlte, tangpara, height, findtop, facds, cross,
unitvector, refin, epsi, gradient, observer, tnew, latlon, vektorsin, alloc_geo, pathcopy, ray_out

- transf_m Collection of subroutines specialized for radiative transfer calculation. (Interpolation between different grids.)
  Calculation of planck/source functions and N_2 and O_2 continua.
  Subroutines:
  n2calc, o2calc, h2ocalc, planck, alpha, dalphadTvib, source, dsourcedTvib, absco1, absco2, scal_add_fabsco,
  fabsco_add_scal, scal_add_absco, absco_add_scal, scal_sub_fabsco, fabsco_sub_scal, scal_sub_absco,
  absco_sub_scal, scal_addsub_absco, scal_mul_fabsco, fabsco_mul_scal, absco_mul_scal, scal_mul_absco,
  absco_mul_complex, complex_mul_absco, cut_fabsco, cut_absco, ipos, fine_absco, fine_fabsco, exp_fabsco,
  exp_absco, absco_to_absco, fabsco_to_absco, fabsco_sub_fabsco, absco_sub_fabsco, fabsco_sub_absco,
  absco_sub_absco, fabsco_add_fabsco, absco_add_fabsco, fabsco_add_absco, absco_add_absco, absco_addsub_absco,
  fabsco_mul_fabsco, absco_mul_fabsco, fabsco_mul_absco, absco_mul_absco, Re_absco, Re_fabsco, Im_absco, Im_fabsco,
  deallocate0, deallocate1, deallocate2, deallocate3, init0_absco, init1_absco, init2_absco, init3_absco
- wriout_m
  Write output spectrum outdat%... to file.
  Subroutines:
  public :: writeout
  private :: writespec
- xinput_m
  Reads measured heavy molecule cross-section data and determines measuring range for every microwindow/xsct.-gas.
  Subroutines:
  public :: input_xsection
  private :: deallocate_range, make_x, readxfilenames, readxbody, readxheader, check_mwXspeci
- xintpl_m
  Pressure, temperature and wavenumber grid interpolation of heavy molecule cross-section measurements.
  Subroutines:
  public :: allocate_x, deallocate_x, interpl_xpt
  private :: interpl_pt1, interpl_pt2, interpl_pt3, interpl_pt4, interpl_x, quadrant, quad_occ, distance1, distance2

• Modules for control tasks:

- kopfwd_m
  Controls forward model run (without in/output) and determines numerical p-derivatives.

Subroutines:
public :: kopra_forwrd
private :: kopra_derip

– kopra Main program.
Subroutines:
-/-

2 Subroutine Listing and Description

```
pri...private
pub...public

Subroutines from:_____________________________________________________________
______________________________________________________________________________
abco_m.f90   (module abco_m)
addlin_m.f90 (module addlin_m)
give_m.f90   (module give_m)
ifnlte_m.f90 (module ifnlte_m)
ilsfov_m.f90 (module ilsfov_m)
inider_m.f90 (module inider_m)
iniout_m.f90 (module iniout_m)
inipar_m.f90 (module inipar_m)
inismw_m.f90 (module inismw_m)
input_m.f90  (module input_m)
inspec_m.f90 (module inspec_m)
kopfwd_m.f90 (module kopfwd_m)
linmix_m.f90 (module linmix_m)
miemod_m.f90 (module miemod_m)
modgeo_m.f90 (module modgeo_m)
modlev_m.f90 (module modlev_m)
nltder_m.f90 (module nltder_m)
offsca_m.f90 (module offsca_m)
parchk_m.f90 (module parchk_m)
radtra_m.f90 (module radtra_m)
ray_m.f90    (module ray_m)
rayctl_m.f90 (module rayctl_m)
varsub_m.f90 (module varsub_m)
wriout_m.f90 (module wriout_m)
______________________________________________________________________________
______________________________________________________________________________
absco_calc@abco_m            pub Calculates absorption coefficients [cm**2]
                                 for all layers and observation geometries
______________________________________________________________________________
absco_branch@abco_m          pri Calculates the absorption coefficient of
                                 branch
______________________________________________________________________________
accuracy_calc@abco_m         pri Calculates the accuracy for absortion cross
                                 section calculation
______________________________________________________________________________
addlev@modlev_m              pri  Add additional levels between two base-levels
                                  if temperature or half-width conditions are
                                  valid
```

add_level@modlev_m	pri	Add one altitude level to a given profile.
add_lines@addlin_m	pub	add the individual lines on the grid made up of different tiers
add_lines_chi@addlin_m	pub	add the individual lines on the grid made up of different tiers
add_lines_chilm@addlin_m	pub	add the individual lines on the grid made up of different tiers
add_lines_lm@addlin_m	pub	add the individual lines on the grid made up of different tiers
addgeo@modgeo_m	pri	adds one geometry to the list of simulated geometries
addnoise@ilsfov_m	pri	Adds Gaussian noise to the coarse spectra
addsim_a@modgeo_m	pri	adds simulated geometries to the observed ones for criterion accu%ifov <= 0 (criterion 1 in input-file)
addsim_b@modgeo_m	pri	adds simulated geometries to the observed ones for criterion accu%ifov > 0 (criterion 2 in input-file)
allocate_cutoff@addlin_m	pub	allocates memory and reads cutoff-files "cutdop.dat" and "cutlor.dat"
allocate_geo_mw@abco_m	pri	allocates geo%...%mw part of geo% variable
allocate_grid@addlin_m	pub	allocates and initializes the grid tiers
alloc_geo@ray_m	pri	allocate geo()%...%lay and geo()%...%lay%speci geo%...%lay%speci%iso , geo%... %iso%state
allocgeo1@rayctl_m	pri	allocation of geo%() and geo%()par()%
alloc_lm@inspec_m	pri	allocate the line mixing branches of vector speci and copy the line data of the line mxing lines from the vector lm into speci()%iso()%band()%branch()%line and speci()%iso()%band()%branch()%lmline
alloc_nlte@inspec_m	pri	determine number of different isotopes for each species where nlte has to be considered and the number of nlte bands. Allocate speci%iso and speci%iso%band
alloc_Sails@radtra_m	pri	allocate data vector where the spectrum and derivatives are stored be careful: the sub-microwindow index is on the internal!

		sub-microwidows
allocno_lm@inspec_m	pri	allocate speci%iso%band%branch in the case no line mixing is considered
allocno_nlte@inspec_m	pri	allocate speci%iso and speci%iso%band in the case no nlte is considered
allocopy1@inspec_m	pri	allocate speci()% vector and copy line data in case no nlte and no line mixing is considered
allocopy2@inspec_m	pri	allocate the vectors speci()%iso()%band()%branch(0)%line and copy the line data from vector spe()%line into this vectors.
alti@varsub_m	pri	Calculation of the altitude using mean temperature of the layer and g at the lower boundary level
altitest@input_m	pri	Test if altitudes in profile file are equal to altitudes in kopra.inp
base@modlev_m	pri	determine index of highest atmospheric base_level: nbaselev so that the highest altitudelevel is lower or equal accu%upatm and make the base-levels
brdaero_gridinv_calc@radtra_m	pri	determine for each gridpoint of the irregular fine grid the corresponding grid point in the broadband-aerosol grid
brdaero_inter@radtra_m	pri	interpolate the broadband-aerosol wavenumber dependence on the irregular fine grid on which the radiative transfer is performed
calc_fq@input_m	pri	calculate the fq-nlte-factors if sw%fq_equ1= .false.
calc_new_abcos@abco_m	pri	determines if the absorption cross sections for a layer of a geometry will be calculated new or if they will be taken from the lowest geometry
calc_nlte_ratios@rayctl_m	pri	the derivative of the ratios of population geo()%...%state()%r with respect to Tkin is calculated
calc_ztang@modlev_m	pri	determination of the tangent altitudes if the nadir angles of a limb scan are given (sw%modeobs = 2 or 4)
check_mwspeci@inspec_m	pri	deletes species from the microwindow species list if there are no lines of the species in

		the microwindow
check_T_hw@modlev_m	pri	Check if conditions on temperature variation and half-width variation are fulfilled so that an additional level must be added
convolution@ilsfov_m	pri	convolution of spectrum with ILS and gridpoint density reduction
corr_y_coefs@linmix_m	pri	corrects the y-coefficients
co2chi@abco_m	pri	Calculates the chi-factor for the correction of the co2-lineshape. The chi-factor is calculated for the n2- and the o2-broadening of co2-lines using the parametrizations from: C. Cousin, R. Le Doucen, C. Boulet, and A. Henry, 'Temperature dependence of the absorption in the region beyond the 4.3-um band head of CO2. 2: N2 and O2 broadening', Appl. Opt., 24, 3899-3907, 1985. V. Menoux, R. Le Doucen, J. Boissoles, and C. Boulet, 'Line shape in the low frequency wing of self- and N2 broadened v3 CO2 lines: temperature dependence of the asymmetrie', Appl. Opt., 30, 281-286, 1991. The chi-factors are linearily interpolated in the ranges 193-238K and 238-296K and linearily extrapolated to lower (higher) temperatures from these ranges.. The chi-factor is then calculated by weighting of the N2 and O2 factors according to their atmospheric abundance.
cross@ray_m	pri	cross - product
dangle_dztang@rayctl_m	pri	calculates sim()%dangle_dztang: the derivative of the nadir angle wrt the tangent altitude
deallocate_cutoff@addlin_m	pub	deallocates space used for the cutoff-tables
deallocate_grid@addlin_m	pub	deallocates the grid tiers
delete_element_i@varsub_m	pri	delete integer element i from an integer vector
delete_level@modlev_m	pri	Delete one level from a given profile.
delete_lmlines@inspec_m	pri	Deletes lines from the %branch(0)%line lines if they are also included in the line mixing

		branches, so that these lines are not calculated twice.
deriaer_calc@radtra_m	pri	Calculation of derivatives with respect to aerosol absorption parameters
derinum_calc@radtra_m	pri	Calculation of derivatives with respect to broadband aerosol number density parameters
deriMie_calc@radtra_m	pri	Calculation of derivatives with respect to Mie parameters
derip_refr@ray_m	pri	Calculate derivatives of the cg-pressure of air and the air-column with respect to p-parameters (including refraction dependence)
deriT_calc@radtra_m	pri	Calculation of derivatives with respect to T parameters
deriTgrad_calc@radtra_m	pri	Calculation of derivatives with respect to Tgrad parameters
deriTvib_calc@radtra_m	pri	Calculation of derivatives with respect to vmr parameters
derive@ilsfov_m	pri	derivative of spectrum with respect to abszissa
derivmr_calc@radtra_m	pri	Calculation of derivatives with respect to vmr parameters
derivmrgrad_calc@radtra_m	pri	Calculation of derivatives with respect to vmr-gradient parameters
envfovils@ilsfov_m	pub	AILS and FOV convolution for ENVISAT
envils@ilsfov_m	pub	AILS convolution for Envisat
epsi@ray_m	pri	calculate refraction index-1 at cartesian point r in atmosphere
error@varsub_m	pub	stop program
extend_mw@input_m	pri	Extend the microwindows on each side by the ilsradius.
facds@ray_m	pri	find, along a direction vector dr, with endpoint heights h1 and h2 the point with height h
findtop@ray_m	pri	find (coming from outside) the crossing point with top level of atmosphere

finewrk@radtra_m	pri	determine the vector wrkrad which contains the spectra for all geometries to the used for field-of-view
fovils1@ilsfov_m	pub	AILS and FOV convolution for linear aperture
fovils_fixed@ilsfov_m	pub	AILS and FOV convolution for read-in AILS-function
fovilsspec@ilsfov_m	pri	determines finite FOV taking 2nd derivative of vertically varying spectrum into account (thus modifying resulting lineshape)
fovmeanspec@ilsfov_m	pri	determines spectrum representative for FOV FOV is assumed to extend vertically around elevcenter from -vertfovradius...+vertfovradius FOV is described by weights of nvertfov horizontal stripes of equal width covering the FOV. Therefore the used elevations extend over a slightly narrower range than -vertfovradius...+vertfovradius
g@varsub_m	pri	Calculation of earth gravitational acceleration depending on altitude and latitude: slightly modified version of: - Redemann (1984) - Clarmann (1986)
gauher@recipe_m	pri	numerical recipe to calculate abscissas and weights for Gauss-Hermite integration
geodistance@give_m	pri	determine the distance (in terms of dx[km] and dy[km] of the point rlat,rlon from the profile location
give_aerabs@give_m	pub	give the aerosol absorption for a distinct altitude
give_aersca@give_m	pub	give the aerosol scattering for a distinct altitude
give_altbase@give_m	pub	give the altitude value to a given index
give_brdaero_prf	pub	give the broadband aerosol parameters for a certain altitude
give_brdaero_wave	pub	give the broadband-aerosol absorption and extinction cross-sections (no Mie-model) or the real and imaginary refraction indices for a distinct altitude (Mie-model)

give_p@give_m	pub give the pressure for a distinct altitude and temperature at this altitude
give_pgrad@give_m	pri give the p-gradients for a given altitude
give_t@give_m	pub give the temperature for a distinct altitude
give_Tgrad@give_m	pri give the T-gradients for a given altitude
give_tvib@give_m	pub give the vibrational temperature for a distinct altitude
give_Tvibgrad@give_m	pri give the vmr-gradients for a given altitude
give_vmr@give_m	pub give the vmr for a distinct altitude
give_vmrgrad@give_m	pri give the vmr-gradients for a given altitude
go_next@varsub_m	pri goto next '§' in input file ifile
gradient@ray_m	pri calculate gradient of (refraction index-1) around point r
grid_t_hw@modlev_m	pri fine-level gridding using T-differences and half-width changes
grid_tang@modlev_m	pri fine-level gridding due to levels above tangent points
height@ray_m	pri calculate distance to surface of ellipse
homog_path@rayctl_m	pri determine the path parameters for homogeneous path calculation
ilsapo@ilsfov_m	pub AILS convolution for circular aperture
ilsapo_fixed@ilsfov_m	pub AILS convolution for read-in AILS-function
ilsapofov_calc@radtra_m	pri convolution of the fine-grid spectra add their derivatives with the apodized instrumental fine shape and field of view calculation
ils_radius@input_m	pri Determination of the radius (in multiples of the fine grid distance wgrid%fine) where the chosen apodisation function 'accu%iapo' is decreased to 1, 0.1, or 0.01% of it's centre value.
index_nlte@ray_m	pri gives for species ispeci and state istate the index in 'tot' vector where Tvibs are stored
ini_deri@inider_m	pub initialize deri% variable
ini_output@iniout_m	pub initialize data vector where the output

		spectrum is stored be careful: the sub-microwindow index is on the external! sub-microwidows
ini_para@inipar_m	pub	Initialization of parameter vector para%.
ini_para_aerabs@inipar_m	pri	initialization of parameter vector for aerosol absorption coefficient parameters
ini_para_Mie@inipar_m	pri	initialization of parameter vector for Mie model
ini_para_num@inipar_m	pri	initialization of parameter vector for broadband aerosol number density
ini_para_p@inipar_m	pri	initialization of parameter vector for pressure parameters
ini_para_T@inipar_m	pri	initialization of parameter vector for temperature parameters
ini_para_Tgrad@inipar_m	pri	initialisation of parameter vector for temperature derivative parameters
ini_para_Tvib@inipar_m	pri	initialization of parameter vector for Tvib parameters
ini_para_vmr@inipar_m	pri	initialization of parameter vector for vmr parameters
ini_para_vmrgrad@inipar_m	pri	initialisation of parameter vector for vmr parameters
ini_sub_mw@inismw_m	pub	determines internal forward model sub-microwindows
input@input_m	pub	controls the input from files and defines input variables
input_brdaero_prf@input_m	pri	read broadband-aerosol altitude profiles of number density, mode-radii and mode-widths
input_brdaero_wave@input_m	pri	read broadband-aerosol wavenumber and altitude dependent absorption and extinction cross-sections or real and imaginary refraction indices
input_contprof@input_m	pri	reads the continuum profiles from file ifile.
input_hitmol@input_m	pri	reads the hitran molecule information from file
input_isoprof@input_m	pri	reads the isotope abundance profiles from file ifile. These profiles are used to

		scale the vmr profiles if single isotopes have to be calculated
input_main@input_m	pri	-reads the data from the main input file -defines the species -extends the microwindows due to ils
input_mwdef@input_m	pri	reads microwindow definition section of main input file
input_pTgradprof@input_m	pri	reads the p,T - gradient profiles from file ifile.
input_pTprof@input_m	pri	reads the p,T profiles from file ifile.
input_spectroscopy@inspec_m	pub	Controls the input of the spectroscopic data, i.e. the determination of the spectroscopic data type construction speci%iso%band%branch%line. This is performed for different options: - no line mixing, no nlte - no line mixing, nlte - line mixing, no nlte - line mixing, nlte
input_Tvibgradprof@input_m	pri	reads the Tvibr profiles from file ifile.
input_Tvibprof@input_m	pri	reads the Tvib profiles from file ifile.
input_vmrgradprof@input_m	pri	reads the vmr - gradient profiles from file ifile.
input_vmrprof@input_m	pri	reads the vmr profiles from file ifile.
integrate@ray_m	pri	explicit integration for layer values and derivatives (columns, Curtis-Godson-T, -p, -Tvib, ...)
interface_nlte@ifnlte_m	pub	Interface to nlte-model
interpolate_grid@addlin_m	pub	interpolates the intervals of the lower tiers to the finest tier used in the summation of lines
isoabun@inspec_m	pri	multiply line strength with isotope abundancy if species is single isotope
isomult@input_m	pri	multiplication of the vmr input profiles of one molecule with the isotope abundance profile in the case a species is a single isotope
kopra_derip@kopfwd_m	pri	Perform numeric p-derivatives

kopra_forwrd@kopfwd_m	pub	Performs Kopra forward calculation and controls numeric p-derivatives
latlon@ray_m	pri	calculate geographic latitude and longitude (first point) of cartesian point r
leveltang@ray_m	pri	perform integration for tangent layer and find exactly tangent position
leveltrans@ray_m	pri	perform integration for all layers except tangent layer and find exact level positions
linein@inspec_m	pri	determines if the line should be read or not
line_strength@abco_m	pri	Calculates the line intensities and optionally the T-derivatives for a bundle of lines
lsimobs@radtra_m	pri	determine lsim and lobs, the activated simulated and observed geometries i.e. the ones which are influenced by the actual retrieval parameter lpara
next_para@ray_m	pri	calculate the index for the next p-parameter which influences the lowest part of the actual geometry (which is influenced most by refraction)
nmax_calc@ray_m	pri	determination of the max. number of integration variables
numdata@inspec_m	pri	Determination of the spectroscopic data files to be opened and allocation of dummy vector spe()%... where the line data are stored temporarily
makecils@ilsfov_m	pri	Generation of ILS from complex interferogram
makeenvifg@ilsfov_m	pri	generates complex IFG (MIPAS-ENVISAT)
make_hydroequi@input_m	pri	Transfer of the input profiles into hydrostatic equilibrium: (the pressures or the altitudes are adjusted)
makeifg@ilsfov_m	pri	generates a real, single sided modulation efficiency interferogram kind of apodisation 1= boxcar in ifg == sinc in spectrum 2= triangle in ifg == sinc^2 in spectrum 3= Hamming (Happ-Genzel) 4= 3 term Blackmann-Harris 5= 4 term Blackmann-Harris 6= Norton-Beer weak 7= Norton-Beer medium

		8= Norton-Beer strong selfapo=T/F with/without selfapodisation by evenly illuminated FOV ifg(1):= ZPD
makeils@ilsfov_m	pri	generates ILS from given interferogram
make_mainspeci@input_m	pri	determine main species number sw%mainspeci
make_modelgeo@modgeo_m	pub	introduces additional geometries for the simulation of FOV and defines the new occupation matrix for all the simulated geometries ('simulated geometries' are called all geometries to be calculated in the forward model i.e. the observed plus the additional geometries for FOV)
make_modelgrid@modlev_m	pri	Defines the model altitude levels for the forward calculation. Starting from a base grid, levels are added in order to fulfill criteria on the T difference and half-width change between levels. Then a criterium for smaller levels distances directly over tangent altitudes is applied. At the end all levels which are less distant than a threshold are deleted.
make_occusim@modgeo_m	pri	determines the microwindow occupation matrix for the simulated geometries
mie@miemod_m	pub	calculate extinciton and absorption cross sections for broadband-aerosol and their derivatives with respect to the distribution parameters
mieampli@miemod_m	pri	calculation of the Mie-amplitude function
Mie_call@abco_m	pri	call the Mie model in order to calculate extinciton and absorption cross sections for broadband-aerosol and their derivatives with respect to the distribution parameters
miecoe@miemod_m	pri	calculate the Mie coefficients for given refraction indices
min_distance@modlev_m	pri	levels which are less distant than accu%dmin are selected out (only the ones that do not belong to the base-grid)
multiple_cut@varsub_m	pri	multiple (direct following) elements of an integer vector are deleted

nlte_postderi@nltder_m	pub	calculate post-derivatives in case of use of nlte-model
nlte_vmr_postderi@nltder_m	pri	calculate post-derivatives for vmr derivatives (dependence on nlte)
nlte_x_postderi@nltder_m	pri	calculate post-derivatives for nlte-model parameters
observer@ray_m	pri	calculate position and viewing direction of observer in cartesian coordinates
off@offsca_m	pri	add offset to spectrum
off_deri@offsca_m	pri	calculate offset derivatives
offset_scale@offsca_m	pub	multiply spectrum and derivatives by factor add offset to spectrum calculate offset and scale derivatives
openerror@varsub_m	pub	stop program if error occurs while opening a file
optsrcdTvib@radtra_m	pri	Calculation of derivative of optical thickness and source function with respect to Tvib
para_aerabs_change@parchk_m	pub	checks if variation of a distinct aerosol absorption parameter influences the absorption values inside an altitude region of the atmosphere (for all microwindows)
para_dabsopt_ne0@rayctl_m	pri	determine the parameters which do not influence the aerosop absorption optical depth i.e. for which the derivative daeropt = 0
para_dbrdaerocol_ne0@rayctl_m	pri	determine the broadband aerosol number density parameters which do not influence the aerosol column
para_dcol_ne0@rayctl_m	pri	determine the parameters which do not influence the partial columns i.e. for which the derivative dcol = 0
para_dcolgrad_ne0@rayctl_m	pri	determine the vmr gradient parameters which do not influence the partial columns i.e. for which the derivative dcol = 0
para_dMie_ne0@rayctl_m	pri	determine the Mie model-parameters which do not influence the CG-Mie layer value
para_dp_ne0@rayctl_m	pri	determine the p parameters which do not influence the cg-p of air i.e. for which the derivative dp = 0

para_dT_ne0@rayctl_m	pri	determine the T parameters which do not influence the cg-T of air i.e. for which the derivative dT = 0
para_dTgrad_ne0@rayctl_m	pri	determine the Tgrad parameters which do not influence the cg-T of air i.e. for which the derivative dTgrad = 0
para_dTvib_ne0@rayctl_m	pri	determine the parameters which do not influence the Tvib's i.e. for which the derivative dTvib_cg_dT = 0
para_Mie_change@parchk_m	pub	checks if variation of a distinct Mie model parameter influences the values inside an altitude region of the atmosphere
para_num_change@parchk_m	pub	checks if variation of a distinct broadband aerosol number density parameter influences the number density values inside an altitude region of the atmosphere
para_p_change@parchk_m	pub	checks if variation of a distinct p parameter influences the p values inside an altitude region of the atmosphere
para_T_change@parchk_m	pub	checks if variation of a distinct T parameter influences the T values inside an altitude region of the atmosphere
para_Tgrad_change@parchk_m	pub	checks if variation of a distinct Tgrad parameter influences the T values inside an altitude region of the atmosphere
para_Tvib_change@parchk_m	pub	checks if variation of a distinct Tvib parameter influences the Tvib values inside an altitude region of the atmosphere
para_vmr_change@parchk_m	pub	checks if variation of a distinct vmr parameter influences the vmr values inside an altitude region of the atmosphere
para_vmrgrad_change@parchk_m	pub	checks if variation of a distinct vmr gradient parameter influences the vmr values inside an altitude region of the atmosphere
pathcopy@ray_m	pri	copy path parameters into vector geo%... for geometry part ipart
pointmw@inspec_m	pri	determine for each microwindow i the range of lines which will be used: speci()%...%mw_l1(i) speci()%...%mw_l2(i).

press@varsub_m	pri	Calculation of the pressure using mean temperature of the layer and g at the lower boundary level
radtrans@radtra_m	pub	Calculation of the radiative transfer through the atmosphere and determination of the fine-grid spectra and their derivatives. Convolution of the fine-grid spectra with the ails to get the spectra on the coarse-grid (measurement grid) and calculation of the field-of-view weighting.
radtrans_mw@radtra_m	pri	perform radiative transfer for spectra and calculate derivatives on the non-equidistant fine-grid
ray_out@ray_m	pri	prepare output variable geo()%... which contains all path parameters
raytra@ray_m	pub	calculation of ray-tracing in inhomogeneous atmosphere and path integration
raytrace_ctrl@rayctl_m	pub	Controls ray-tracing, calculation of path integrated values (Curtis-Godson values and column amounts) and the derivatives of path integrated values wrt retrieval parameters.
readlines@inspec_m	pri	read in spectroscopic data
read_lmdata@inspec_m	pri	read the line mixing data from file fil%linemix into vector 'lm'. Only the branches which are 'near' a microwindow (distance w_linemix) are read.
refin@ray_m	pri	calculate refraction index for pressure px and temperatute tx
sca@offsca_m	pri	multiply spectrum and derivatives by scale
sca_deri@offsca_m	pri	calculate scale derivatives
scndderfov@ilsfov_m	pri	second derivative with respect to elevation representative for FOV (notice: unit of angle is vertfovradius)
set_gauher@miemod_m	pri	determine the Gauss-Hermite abscissas and weights for the particle distribution and its derivative with respect to the distribution parameters
set_miepar@miemod_m	pri	set the discretization for the scattering angles

shiftwvnr@ilsfov_m	pri	performs shift of spectrum along abszissa
speciorder@input_m	pri	determination of species describing variables
sub_mw_minmax@radtra_m	pri	determine first and last index in equidistant fine grid spectrum for one sub_microwindow
tangpara@ray_m	pri	determine tangent height and distances by parabolic interpolation
tangalt@ray_m	pub	calculate tangent altitude (not exact)
tausrc@radtra_m	pri	Calculation of layer transmission tau and layer source function src (nlte considered)
tnew@ray_m	pri	calculation of new tangent vector along LOS
unitvector@ray_m	pri	calculate unit vector of vector vr
value_test@ilsfov_m	pri	Test for change of variable
vektorsin@ray_m	pri	sine of angle between two vectors
vertinterpol@ilsfov_m	pri	interpolates linear in elevation between given spectra
voigt@addlin_m	pri	Calculates the Voigt-Function times a user-definied value fac with a relative accuracy better than 2*10-4. The algorithm switches automatically from the calculation of the Voigt-Function to the calculation of the Lorentz-Function. If this subroutine is called several times with similar values y, the numerically expensive coefficents a1..t8 are only calculated once. The coefficients are only recalculated if the relative change in y is grater than the internal parameter rel, which is set to 1e-4. ref: A new implementation of the Humlicek algorithm for the calculation of the Voigt profile function, J. Quant. Spectrosc. Radiat. Transfer, 57, 819-824, 1997
warning@varsub_m	pub	write warning
writeabco@abco_m	pri	printout of absorption coefficient
writeabco@radtra_m	pri	write in file absco_type spectra
writeout@wriout_m	pub	write the spectra and derivatives on the coarse wavenumber grid on file
writespec@radtra_m	pri	write in file spectrum stored in vector

```
------------------------------------------------------------------------------
writespec@wriout_m          pri write one spectrum on file
------------------------------------------------------------------------------
y_calc_dd@linmix_m          pri calculates the y-coefficients for direct
                                diagonalisation
------------------------------------------------------------------------------
y_calc_rk@linmix_m          pri calculates the y-coefficients for the
                                Rosenkranz-approximation
------------------------------------------------------------------------------
yinterpol1@varsub_m         pri returns the linearly interpolated and
                                extrapolated y value when the x value and a
                                grid y(n) and x(n) is given
------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

3 Variable Listing and Description

```
==============================================================================

accu%
       accuracy_type : computational accuracy
accu%absolute_absco
                real : absolute accuracy for optical depth
              Origin : input_main@input_m
accu%basealt()
                real : altitude of the base-levels [km]
           Dimension : n%baselev
              Origin : input_main@input_m
accu%dif
                real : maximum layer thickness above the tangent altitude [km]
              Origin : input_main@input_m
accu%dmin
                real : minimal layer thickness [km]
              Origin : input_main@input_m
accu%iapo
             integer : accuracy of apodisation (defines mw extension)
                       1= apodisation function decreases to  1%    of center value
                       2= apodisation function decreases to  0.1%  of center value
                       3= apodisation function decreases to  0.01% of center value
              Origin : input_main@input_m
accu%iexpath
             integer : number of extra paths where the absorption
                       coefficients are recalculated/interpolated
              Origin : input_main@input_m
accu%ifov()
             integer : criteria for addition of simulated
                       geometries for field of view calculation
           Dimension : 2
              Origin : input_main@input_m
```

```
accu%ifov_sep
             integer : number of the geometry up to which ifov(1)
                       is used as criterium for addition of
                       simulated geometries
              Origin : input_main@input_m
accu%raytrace
                real : integration step length [km]
              Origin : input_main@input_m
accu%retain_absco
             integer : =0 lines my be rejected and truncated
                       =1 all lines retained, no truncation
              Origin : input_main@input_m
accu%Tvar1
                real : max. temperature variation between
                       levels in the lower altitude region
              Origin : input_main@input_m
accu%Tvar2
                real : max. temperature variation between
                       levels in the higher altitude region
              Origin : input_main@input_m
accu%w_useline
                real : wavenumber range outside mw where lines
                       are taken into account for
                       calculation of each line
              Origin : input_main@input_m
accu%wvar
                real : maximum variation of the line width
                       between two model levels
              Origin : input_main@input_m
accu%upto
                real : altitude above the tangent up to which
                       this maximum thickness (accu%dif) is valid [km]
              Origin : input_main@input_m
accu%zTvar
                real : altitude [km] which divides the regions
                       for temperature variation between levels
              Origin : input_main@input_m
================================================================================
brdaero%
        brdaero_type : broadband-aerosol definition

brdaero%alt_prf()
                real : altitudes of broadband-aerosol profile levels [km]
           Dimension : brdaero%nlev_prf
              Origin : input_brdaero_prf@input_m
brdaero%alt_wave()
                real : altitudes of levels of broadband-aerosol
                       refraction indices or cross sections [km]
           Dimension : brdaero%nlev_wave
              Origin : input_brdaero_wave@input_m
brdaero%iwave1()
             integer : begin (1,:)  wavenumber index for broadband-aerosol
                       refraction indices or cross sections for each mw
           Dimension : n%mw
```

```
              Origin : input_brdaero_wave@input_m
brdaero%nlev_prf
             integer : number of levels for the broadband-aerosol
                       profiles
              Origin : input_brdaero_prf@input_m
brdaero%nlev_wave
             integer : number of altitude levels for the broadband-aerosol
                       refraction indices or cross sections
              Origin : input_brdaero_wave@input_m
brdaero%nMie
             integer : number of Mie-parameter sets
              Origin : input_main@input_m
brdaero%nmode
             integer : number of modes for broadband-aerosol
                       particle distribution function
              Origin : input_brdaero_prf@input_m
brdaero%nwave()
             integer : number of wavenumber indices for broadband-aerosol
           Dimension : n%mw
              Origin : input_brdaero_wave@input_m
================================================================================

brdaero_prf(:,:)
                real : broadband-aerosol profiles of particle number density
                       [particles/cm3] and mode-radii [micrometer] and
                       mode-widths and ratio of mode1 to mode1+mode2
           Dimension :(brdaero%nlev_prf,6) if sw%Miemod (Mie-model)
                      (brdaero%nlev_prf,1) if not sw%Miemod (no Mie-model)
              Origin : input_brdaero_prf@input_m
================================================================================

brdaero_wave(:,:,:,:)
                real : broadband-aerosol wavenumber
                       dependent refraction index (if sw%Miemod)
                       or cross section [cm2/particle] (if not sw%Miemod)
           Dimension : (i,2,brdaero%nlev_wave,n%mw) with
                       i is max number of aerosol grid points for all mw's
================================================================================

deri%
     derivative_type : derivative definition
deri%igasiso()
             integer : gas/isotope identifier for deriv. species
           Dimension : deri%nspeci
              Origin : input_main@input_m
deri%igasiso_grad()
             integer : gas/isotope identifier for vmr-gradientderiv. species
           Dimension : deri%n_vmrgrad_speci
              Origin : input_main@input_m
deri%igasiso_nlte
             integer : Tvib species for derivative
              Origin : input_main@input_m
deri%i_nlte_iso
             integer : global nlte_isotope number
```

```
                    for which Tvib derivatives are calculated
             Origin : ini_deri@inider_m
deri%i_nlte_speci
             integer : global species number for which Tvib
                       derivatives are calculated
             Origin : ini_deri@inider_m
deri%i_nlte_state()
             integer : pointer to the global nlte state number
           Dimension : deri%n_nlte_state
             Origin : ini_deri@inider_m
deri%istate()
             integer : states for which T-vib derivatives
                       should be calcualted
           Dimension : deri%n_nlte_state
             Origin : input_main@input_m
deri%i_vmrgrad_speci()
             integer : pointer to global species numbering
           Dimension : deri%n_vmrgrad_speci
             Origin : ini_deri@inider_m
deri%i_vmr_speci()
             integer : pointer to global species numbering
           Dimension : deri%n_vmr_speci
             Origin : ini_deri@inider_m
deri%nlte_speci
             integer : pointer to the nlte-species of nlte()%..input variable
             Origin : ini_deri@inider_m
deri%n_nlte_state
             integer : number of T-vib states for which derivatives
                       should be calculated
             Origin : input_main@input_m
deri%n_vmrgrad_speci
             integer : number of species for vmr-gradient derivative
             Origin : input_main@input_m
deri%n_vmr_speci
             integer : number of species for vmr derivative
             Origin : input_main@input_m
================================================================================

fil%
      filenames_type : input file names and directories
fil%ails
      character(200) : apodized instrumental line shape
             Origin : input_main@input_m
fil%brdaero_cross
      character(200) : particle cross sections for broadband-aerosol
                       absorption and extinction
             Origin : input_main@input_m
fil%brdaero_prf
      character(200) : profiles for broadband-aerosol number
                       density and size distribution parameters
             Origin : input_main@input_m
fil%brdaero_refind
      character(200) : wavenumber dependent refraction indice
                       profiles for broadband-aerosol
```

```
              Origin : input_main@input_m
fil%contprof
      character(200) : continuum profiles
              Origin : input_main@input_m
fil%cutdop
      character(200) : doppler cut-off for abco calculation
              Origin : input_main@input_m
fil%cutlor
      character(200) : lorentz cut-off for abco calculation
              Origin : input_main@input_m
fil%err
      character(200) : output error file
              Origin : input_main@input_m
fil%errunit
             integer : output error file unit number
              Origin : input_main@input_m
fil%hitmol
      character(200) : data on hitran molecules, isotopes, states
              Origin : input_main@input_m
fil%isoprof
      character(200) : isotope abundance profiles
              Origin : input_main@input_m
fil%linedatadir
      character(200) : spectrospcopic data directory
              Origin : input_main@input_m
fil%linemix
      character(200) : line-mixing data
              Origin : input_main@input_m
fil%log
      character(200) : output log file
              Origin : input_main@input_m
fil%logunit
             integer : output log file unit number
              Origin : input_main@input_m
fil%outcoarse
      character(200) : spectra on coarse wavenumber grid
              Origin : input_main@input_m
fil%outfine
      character(200) : spectra on fine wavenumber grid
              Origin : input_main@input_m
fil%ptgradprof
      character(200) : p,T - gradient - profiles
              Origin : input_main@input_m
fil%ptprof
      character(200) : p,T - profiles
              Origin : input_main@input_m
fil%Tvibgradprof
      character(200) : Tvib-gradient profiles
              Origin : input_main@input_m
fil%Tvibprof
      character(200) : Tvib profiles
              Origin : input_main@input_m
fil%vmrgradprof
      character(200) : vmr - gradient - profiles
```

```
              Origin : input_main@input_m
fil%vmrprof
      character(200) : vmr - profiles
              Origin : input_main@input_m
fil%Xdatadir
      character(200) : cross-section data directory
              Origin : input_main@input_m
================================================================================

geo()%
            geo_type : path definition
           Dimension : n%simgeo
geo()%nparts
             integer : number of parts of the geometry
              Origin : allocgeo1@rayctl_m, homog_path@rayctl_m

geo()%par()%
            par_type : parts of geometry (1:far and 2:near observer)
           Dimension : geo()%nparts
geo()%par()%l_high
             integer : highest layer number
              Origin : ray_out@ray_m, homog_path@rayctl_m
geo()%par()%l_low
             integer : lowest layer number
              Origin : ray_out@ray_m, homog_path@rayctl_m

geo()%par()%lay()%
            lay_type : layers description
           Dimension : (geo()%par()%l_low,geo()%par()%l_high)
geo()%par()%lay()%absopt()
                real : aerosol absorption optical depth
           Dimension : n%mw
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%brdaerocol
                real : broadband-aerosol column [particles/cm2]
              Origin : pathcopy@ray_m
geo()%par()%lay()%col
                real : total air column in layer [molecules/cm2]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%cross(:,:,:)
                real : broadband-aerosol absorption and extinction
                       cross-sections [cm2/particle]
           Dimension : size(brdaero_wave,1),2,n%mw
              Origin : pathcopy@ray_m
geo()%par()%lay()%dabsopt()
                real : derivative of aerosol absorption optical
                       depth with respect to each aerosol
                       absorption coefficient parameter for each mw
           Dimension : (n%mw,para%n_aerabs_para)
              Origin : pathcopy@ray_m
geo()%par()%lay()%dbrdaerocol()
                real : derivative of broadband aerosol column
                       with respect to each broadband aerosol
                       number density parameter
```

```
           Dimension : para%n_brdaeronum_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%dcol()
                real : derivative of air column
                       with respect to each p-parameter
           Dimension : para%n_p_para, deri_p_calc@ray_m
              Origin : pathcopy@ray_m
geo()%par()%lay()%dcross(:,:,:,:)
                real : derivatives of broadband-aerosol absortion
                       and extinction cross-sections with respect to
                       C.G. Mie layer values
           Dimension : size(brdaero_wave,1),2,brdaero%nMie,n%mw
              Origin : pathcopy@ray_m
geo()%par()%lay()%dMie(:,:)
                real : derivative of Mie C.-G. parameters
           Dimension : brdaero%nMie, para%n_Mie_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%dp()
                real : derivative of cg-p of air with
                       respect to each p-parameter
           Dimension : para%n_p_para, deri_p_calc@ray_m
              Origin : pathcopy@ray_m
geo()%par()%lay()%dT()
                real : derivative of cg-T of air with
                       respect to each T-parameter
           Dimension : para%n_T_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%dTgrad()
                real : derivative of cg-T of air with
                       respect to each Tgrad-parameter
           Dimension : para%n_Tgrad_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%extopt()
                real : aerosol extinction optical depth
           Dimension : n%mw
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%i_dabsopt_ne0()
             integer : dabsopt index for dabsopt /= 0
           Dimension : geo()%par()%lay()%n_dabsopt_ne0
              Origin : para_dabsopt_ne0@rayctl_m
geo()%par()%lay()%i_dbrdaerocol_ne0()
             integer : dbrdaerocol index for dbrdaerocol /= 0
           Dimension : geo()%par()%lay()%n_dbrdaerocol_ne0
              Origin : para_dbrdaerocol_ne0@rayctl_m
geo()%par()%lay()%i_dMie_ne0
             integer : dMie index for dMie /= 0
              Origin : para_dMie_ne0@rayctl_m
geo()%par()%lay()%i_dp_ne0()
             integer : dp index for dp /= 0
           Dimension : geo()%par()%lay()%n_dp_ne0
              Origin : para_dp_ne0@rayctl_m
geo()%par()%lay()%i_dTgrad_ne0()
             integer : dTgrad index for dTgrad /= 0
           Dimension : geo()%par()%lay()%n_dTgrad_ne0
```

```
              Origin : para_dTgrad_ne0@rayctl_m
geo()%par()%lay()%i_dT_ne0()
             integer : dT index for dT /= 0
           Dimension : geo()%par()%lay()%n_dT_ne0
              Origin : para_dT_ne0@rayctl_m
geo()%par()%lay()%n_dabsopt_ne0
             integer : number of dabsopt values which are /= 0
              Origin : para_dabsopt_ne0@rayctl_m
geo()%par()%lay()%n_dbrdaerocol_ne0
             integer : number of dbrdaerocol values which are /= 0
              Origin : para_dbrdaerocol_ne0@rayctl_m
geo()%par()%lay()%n_dMie_ne0
             integer : number of dMie values which are /= 0
              Origin : para_dMie_ne0@rayctl_m
geo()%par()%lay()%n_dp_ne0
             integer : number of dp values which are /= 0
              Origin : para_dp_ne0@rayctl_m
geo()%par()%lay()%n_dTgrad_ne0
             integer : number of dTgrad values which are /= 0
              Origin : para_dTgrad_ne0@rayctl_m
geo()%par()%lay()%n_dT_ne0
             integer : number of dT values which are /= 0
              Origin : para_dT_ne0@rayctl_m
geo()%par()%lay()%path
                real : path length in layer [km]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%pcg
                real : curtis godson pressure for air [hPa]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%refind(:,:,:)
                real : broadband-aerosol CG refraction index values
           Dimension : size(brdaero_wave,1),2,n%mw
              Origin : pathcopy@ray_m
geo()%par()%lay()%Tcg
                real : curtis godson temperature for air [K]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%vn
                real : quotient of number density of mode 1 to total
                       nuber density for broadband-aerosol particle
                       distribution function
              Origin : pathcopy@ray_m
geo()%par()%lay()%vr()
                real : mode-radius for broadband-aerosol distribution
                       function [micrometer]
           Dimension : 2
              Origin : pathcopy@ray_m
geo()%par()%lay()%vs()
                real : mode-width for broadband-aerosol distribution
                       function
           Dimension : 2
              Origin : pathcopy@ray_m

geo()%par()%lay()%speci()%
        speci_type2 : layerspecies
```

```
           Dimension : n%speci
geo()%par()%lay()%speci()%col
                real : partial column amount [molecules/cm2]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%dcol()
                real : derivative of partial column amount with
                       respect to each vmr-parameter
           Dimension : para%speci()%n_vmr_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%speci()%dcolgrad()
                real : derivative of partial column amount with
                       respect to each vmr-gradient parameter
           Dimension : para%speci()%n_vmrgrad_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%speci()%i_dcolgrad_ne0()
             integer : dcolgrad index for dcol /= 0
           Dimension : geo()%par()%lay()%speci()%n_dcolgrad_ne0
              Origin : para_dcolgrad_ne0@rayctl_m
geo()%par()%lay()%speci()%i_dcol_ne0()
             integer : dcol index for dcol /= 0
           Dimension : geo()%par()%lay()%speci()%n_dcol_ne0
              Origin : para_dcol_ne0@rayctl_m
geo()%par()%lay()%speci()%n_dcolgrad_ne0
             integer : number of dcolgrad values which are /= 0
              Origin : para_dcolgrad_ne0@rayctl_m
geo()%par()%lay()%speci()%n_dcol_ne0
             integer : number of dcol values which are /= 0
              Origin : para_dcol_ne0@rayctl_m
geo()%par()%lay()%speci()%n_nlte_iso
             integer : number of nlte isotopes
              Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%pcg
                real : curtis godson pressure [hPa]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%Tcg
                real : curtis godson temperature [K]
              Origin : pathcopy@ray_m, homog_path@rayctl_m

geo()%par()%lay()%speci()%iso()%
          iso_type3 : isotope
           Dimension : speci()%n_nlte_iso
geo()%par()%lay()%speci()%iso()%i
             integer : hitran isotope number
              Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%iso()%nlte_speci
             integer : pointer to the nlte-species
              Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%iso()%nstate
             integer : number of nlte vib states
              Origin : alloc_geo@ray_m, homog_path@rayctl_m

geo()%par()%lay()%speci()%iso()%state()%
        state_type2 : band vibrational state
          Dimension : 0:nlte(speci()%iso()%nlte_speci)%nstate
```

```
geo()%par()%lay()%speci()%iso()%state()%i
             integer : HITRAN vibr. state-number
              Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%i_dTvib_ne0()
             integer : index for dTvib_cg_dT /= 0
           Dimension : geo()%par()%lay()%speci()%iso()%state()%n_dTvib_ne0
              Origin : para_dTvib_ne0@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%dr_dTkin
                real : derivative of r with respect to Tkin - cg
              Origin : calc_nlte_ratios@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%dTvib_cg_dT()
                real : derivative of Tvib with respect to
                       each Tvib parameter
           Dimension : para%state()%n_Tvib_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%speci()%iso()%state()%n_dTvib_ne0
             integer : number of dTvib_cg_dT values which are /= 0
              Origin : para_dTvib_ne0@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%r
                real : ratio of populations between nlte and lte
                       case for actual state
              Origin : pathcopy@ray_m, homog_path@rayctl_m

geo()%par()%lay()%speci()%mw()%
       mw_absco_type : microwindow (for absorption coefficients)
           Dimension : n%mw
geo()%par()%lay()%speci()%mw()%l_absco_calc
             logical : =.true. if absorption coeff's in
                       mw_absco_type are calculated
                       =.false. if absorption coeff's are
                       mw_absoc type is only a pointer
                       to already calculated abscos
              Origin : allocate_geo_mw@abco_m, absco_calc@abco_m

geo()%par()%lay()%speci()%mw()%absco%
          absco_type : absorption coefficient lte
geo()%par()%lay()%speci()%mw()%absco%abco()
                real : absorption coefficients [cm2/molecule]
           Dimension : geo()%par()%lay()%speci()%mw()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%dabcodT()
                real : derivative of absorption coefficients
                        with respect to temperature
                        [cm2/molecule/K]
           Dimension : geo()%par()%lay()%speci()%mw()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%fdel
                real : wavenumber fine grid distance [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%fmax
                real : highest wavenumber [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%fmin
                real : lowest wavenumber [cm-1]
```

```
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%iabco()
             integer : index for the point in the wavenumber
                       fine grid where the absorption coefficient
                       is stored
           Dimension : geo()%par()%lay()%speci()%mw()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%iabcomx
             integer : number of absorption coeffients
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%Tflag
             logical : =.true. if absorption coefficient
                       T-derivative is calculated
              Origin : absco_calc@abco_m

geo()%par()%lay()%speci()%mw()%iso()%
      iso_absco_type : nlte isotope for abs. coef.
           Dimension : speci()%n_nlte_iso

geo()%par()%lay()%speci()%mw()%iso()%band()%
     band_absco_type : band for absorption coefficients
           Dimension : speci()%iso()%n_nlte_bands
geo()%par()%lay()%speci()%mw()%iso()%band()%l_absco_calc
             logical : =.true. if absorption coeff's in
                       mw_absco_type are calculated
                       =.false. if absorption coeff's are
                       mw_absoc type is only a pointer
                       to already calculated abscos
              Origin : allocate_geo_mw@abco_m, absco_calc@abco_m

geo()%par()%lay()%speci()%mw()%iso()%band()%absco%
          absco_type : absorption coefficient nlte
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%abco()
                real : absorption coefficients [cm2/molecule]
           Dimension : geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%dabcodT()
                real : derivative of absorption coefficients
                        with respect to temperature
                        [cm2/molecule/K]
           Dimension : geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%fdel
                real : wavenumber fine grid distance [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%fmax
                real : highest wavenumber [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%fmin
                real : lowest wavenumber [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabco()
             integer : index for the point in the wavenumber
                       fine grid where the absorption coefficient
```

```
                        is stored
           Dimension : geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
             integer : number of absorption coeffients
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%Tflag
             logical : =.true. if absorption coefficient
                       T-derivative is calculated
              Origin : absco_calc@abco_m
================================================================================

inprof()%
     inprofiles_type : input profiles
           Dimension : n%inlev
inprof()%alt
                real : altitudes [km]
              Origin : input_main@input_m
inprof()%aerabs()
                real : continuum absorption coefficient [km-1]
           Dimension : n%mw
              Origin : input_contprof@input_m
inprof()%aersca()
                real : continuum scattering coefficient [km-1]
           Dimension : n%mw
              Origin : input_contprof@input_m
inprof()%lat
                real : latitude of the profile [rad]
              Origin : input_main@input_m
inprof()%lon
                real :  longitude of the profile [rad]
                        (increasing in easterly direction)
              Origin : input_main@input_m
inprof()%p
                real : pressure [hPa]
              Origin : input_pTprof@input_m
inprof()%T
                real : temperature [K]
              Origin : input_pTprof@input_m
inprof()%Tvib()
                real : vibrational temperatures [K]
           Dimension : (nlte()%nstate,n%nlte_speci)
              Origin : input_Tvibprof@input_m
inprof()%vmr()
                real : vmr [ppmv]
           Dimension : n%speci
              Origin : input_vmrprof@input_m
              Modify : isomult@input_m

inprof()%latgrad%
  ingrad_profiles_type : gradient profiles along latitude circles
                           (positive to east)
inprof()%latgrad%p
                real : pressure gradient [hPa/km]
              Origin : input_pTgradprof@input_m
```

```
inprof()%latgrad%T
                real : temperature gradient [K/km]
              Origin : input_pTgradprof@input_m
inprof()%latgrad%Tvib()
                real : vibrational temperature gradient [K/km]
           Dimension : (nlte()%nstate,n%nlte_speci)
              Origin : input_Tvibgradprof@input_m
inprof()%latgrad%vmr()
                real : vmr gradient [ppmv/km]
           Dimension : n%speci
              Origin : input_vmrgradprof@input_m
              Modify : isomult@input_m

inprof()%longrad%
  ingrad_profiles_type : gradient profiles along longitude circles
                         (positive to south)
inprof()%longrad%p
                real : pressure gradient [hPa/km]
              Origin : input_pTgradprof@input_m
inprof()%longrad%T
                real : temperature gradient [K/km]
              Origin : input_pTgradprof@input_m
inprof()%longrad%Tvib()
                real : vibrational temperature gradient [K/km]
           Dimension : (nlte()%nstate,n%nlte_speci)
              Origin : input_Tvibgradprof@input_m
inprof()%longrad%vmr()
                real : vmr gradient [ppmv/km]
           Dimension : n%speci
              Origin : input_vmrgradprof@input_m
              Modify : isomult@input_m
================================================================================

inst%
     instrument_type : instrumental specifications
inst%ilsradius
             integer : radius of the apodisation (or ils) function
                       [integer multiples of wgrid%fine]
              Origin : input_main@input_m
inst%noise_seed
             integer : seed value for the noise:
                         >= 0: same noise in two runs
                         < 0: noise always changes
              Origin : input_main@input_m
inst%rms_sinc()
                real : rms value of the noise in the sinc-spectrum
                       (independent grid-values of distance 1/2*opdmax)
           Dimension : n%mw
              Origin : input_main@input_m

inst%gen%
   general_inst_type : ils-model parameters for general instrument
inst%gen%apolin
                real : Part of linear Apodisation
```

```
                    1.0= perfect instrument, 0.0= triangle
             Origin : input_main@input_m
inst%gen%coneifm
               real : maximum inclination of the ray in the
                      interferometer [rad]
             Origin : input_main@input_m
inst%gen%iapokind
            integer : kind of apodisation
             Origin : input_main@input_m
inst%gen%opdmax
               real : maximum path difference [cm]
             Origin : input_main@input_m
inst%gen%phaserr
               real : Phase error [rad]
             Origin : input_main@input_m

inst%envi%
       envisat_type : MIPAS-envisat ils-model parameters
inst%envi%airy
               real : infrared misalignment y-direction [rad]
             Origin : input_main@input_m
inst%envi%airz
               real : infrared misalignment z-direction [rad]
             Origin : input_main@input_m
inst%envi%aly
               real : laser misalignment along y [rad]
             Origin : input_main@input_m
inst%envi%alz
               real : laser misalignment along z [rad]
             Origin : input_main@input_m
inst%envi%by
               real : blur angular width along y [rad]
             Origin : input_main@input_m
inst%envi%bz
               real : blur angular width along z [rad]
             Origin : input_main@input_m
inst%envi%dalphy
               real : full interferometer divergence along y [rad]
             Origin : input_main@input_m
inst%envi%dalphz
               real : full interferometer divergence along z [rad]
             Origin : input_main@input_m
inst%envi%deltat
               real : mismatched delay between electronic response
                      and ADC trigger [s]
inst%envi%delxy
               real : linear shear variation along y [-]
             Origin : input_main@input_m
inst%envi%delxz
               real : linear shear variation along z [-]
             Origin : input_main@input_m
inst%envi%dely
               real : retroreflector linear shear along y [cm]
             Origin : input_main@input_m
```

```
inst%envi%delz
                real : retroreflector linear shear along z [cm]
              Origin : input_main@input_m
inst%envi%dvnull
                real : initial relative speed fluctuation at
                       beginning of scan [cm/s]
              Origin : input_main@input_m
inst%envi%dxnull
                real : initial sampling pertubation [dxnull]
              Origin : input_main@input_m
inst%envi%iapokind
             integer : kind of apodisation
              Origin : input_main@input_m
inst%envi%lfnoise
                real : bandwith laser 1/f noise [Hz]
              Origin : input_main@input_m
inst%envi%lwnoise
                real : bandwith laser white noise [Hz]
              Origin : input_main@input_m
inst%envi%lwvdrift
                real : relative drift rate of laser wvnr [-]
              Origin : input_main@input_m
inst%envi%lwvnr
                real : laser wvnr [cm-1]
              Origin : input_main@input_m
inst%envi%opdmax
                real : maximum path difference [cm]
              Origin : input_main@input_m
inst%envi%pgain
                real : gain slope of IR electrical response [-]
              Origin : input_main@input_m
inst%envi%taus
                real : time constant of exponential attentuation of
                       initial sampling pertubation [s]
              Origin : input_main@input_m
inst%envi%tauv
                real : time constant of exponential attentuation of
                       initial speed fluctuation [s]
              Origin : input_main@input_m
inst%envi%vscan
                real : optical speed of interferometer [cm/s]
              Origin : input_main@input_m

inst%mw()%
         inst_mw_type : microwindow dependent instrument parameters
            Dimension : n%mw
inst%mw()%ails()
                real : read-in apodised instrumental line shape
            Dimension : inst%mw()%n_ails_pts
              Origin : input_ails@input_m
inst%mw()%n_ails_pts
             integer : number of spectral grid points of read-in ails
              Origin : input_ails@input_m
inst%mw()%off()
```

```
                real : instrumental offset per geometry [nW/cm2Srcm-1]
           Dimension : n%obsgeo
              Origin : input_main@input_m
inst%mw()%sca()
                real : instrumental scale per geometry
           Dimension : n%obsgeo
              Origin : input_main@input_m
inst%mw()%shift
                real :  wavenumber shift [cm-1]
              Origin : input_main@input_m
inst%mw()%vertconefov
                real : half vertical fov extension [rad]
              Origin : input_main@input_m
inst%mw()%vertfov()
                real : weighting function for vertical fov
           Dimension : nvertfov
              Origin : input_main@input_m
================================================================================

modprof()%
  modelprofiles_type : model altitudes
           Dimension : n%modlev
modprof()%alt
                real : altitude of model levels [km]
              Origin : make_modelgrid@modlev_m
================================================================================

mol()%
         hitmol_type : HITRAN molecule information
           Dimension : nlingas
mol()%abunalt()
                real : altitude levels of the isotope profiles
           Dimension : mol()%nlev
              Origin : input_isoprof@input_m
mol()%n
             integer : number of isotopes
              Origin : input_hitmol@input_m
mol()%nlev
             integer : number of levels for the isotope profiles
              Origin : input_isoprof@input_m

mol()%iso()%
           iso_type1 : information on isotopes
           Dimension : mol()%n
mol()%iso()%abunhit
                real : hitran isotope abundance
              Origin : input_hitmol@input_m
mol()%iso()%abunprof()
                real : isotope abundance profiles
           Dimension : mol()%nlev
              Origin : input_isoprof@input_m
mol()%iso()%iground
             integer : hitran number of each vibration ground state
           Dimension : mol()%iso()%nground
```

```
              Origin : input_hitmol@input_m
mol()%iso()%nground
             integer : number of vibrational ground states
              Origin : input_hitmol@input_m
mol()%iso()%nstate
             integer : number of vibr. states
              Origin : input_hitmol@input_m
mol()%iso()%qcoef1
                real : Gamache coefficients for partiton-sum
                       calculation(70-500K)
           Dimension : 4
              Origin : input_hitmol@input_m
mol()%iso()%qcoef2
                real : Gamache coefficients for partiton-sum
                       calculation(>500K)
           Dimension : 4
              Origin : input_hitmol@input_m
mol()%iso()%q296
                real : partition sums at 296 K
              Origin : input_hitmol@input_m
mol()%iso()%wmol
                real : molecular weight [g/mol]
              Origin : input_hitmol@input_m

mol()%iso()%state()%
         state_type1 : state vector
           Dimension : mol()%iso()%nstate
mol()%iso()%state()%E
                real : state energy
              Origin : input_hitmol@input_m
================================================================================

mw()%
    microwindow_type : mw definition
           Dimension : n%mw
mw()%igasiso()
             integer : molecule/isotope identifier of each species
           Dimension : mw()%nspeci
              Origin : input_mwdef@input_m
mw()%ispeci()
             integer : general species number of each species in mw
           Dimension : mw()%nspeci
              Origin : speciorder@input_m
mw()%nsmw
             integer : number of sub-microwindows
              Origin : ini_sub_mw@inismw_m
mw()%nspeci
             integer : number of species per mw
              Origin : input_mwdef@input_m
mw()%occuobs()
             logical : occupation matrix for the observations
           Dimension : n%obsgeo
              Origin : ini_sub_mw@inismw_m
mw()%occusim()
```

```
             logical : occupation matrix for the simulations
           Dimension : n%simgeo
              Origin : make_occusim@modgeo_m
mw()%shifthitwvnr()
                real : shift vs HITRAN data for each species and microwindow
           Dimension : mw()%nspeci
              Origin : input_mwdef@input_m
mw()%sim_obs()
             logical : which simulations are necessary for which observations
           Dimension : (n%simgeo,n%obsgeo)
              Origin : make_occusim@modgeo_m
mw()%w1
                real : begin wavenumber of mw [cm-1]
              Origin : input_mwdef@input_m
mw()%w2
                real : end wavenumber of mw  [cm-1]
              Origin : input_mwdef@input_m
mw()%w1_ils
                real : begin wavenumber of ils-extended mw [cm-1]
              Origin : extend_mw@input_m
mw()%w2_ils
                real : end wavenumber of ils-extended mw [cm-1]
              Origin : extend_mw@input_m

mw()%geo()%
         mw_geo_type : geometries for external sub-mws
           Dimension : n%obsgeo
mw()%geo()%nsmw
             integer : number of external sub-mws
              Origin : input_mwdef@input_m

mw()%geo()%smw()%
      mw_geosmw_type : external sub-mws
           Dimension : mw()%geo()%nsmw
mw()%geo()%smw()%w1
                real : begin wavenumber of external sub-mw [cm-1]
              Origin : input_mwdef@input_m
mw()%geo()%smw()%w2
                real : end wavenumber of external sub-mw [cm-1]
              Origin : input_mwdef@input_m
mw()%geo()%smw()%w1_ils
                real : begin wavenumber of ils-extended external mw [cm-1]
              Origin : extend_mw@input_m
mw()%geo()%smw()%w2_ils
                real : end wavenumber of ils-extended external mw [cm-1]
              Origin : extend_mw@input_m

mw()%smw()%
         sub_mw_type : internal sub-microwindows
mw()%smw()%occuobs()
             logical : occupation matrix for the observations
           Dimension : n%obsgeo
              Origin : ini_sub_mw@inismw_m
mw()%smw()%occusim()
```

```
            logical : occupation matrix for the simulations
          Dimension : n%simgeo
             Origin : make_occusim@modgeo_m
mw()%smw()%sim_obs()
            logical : which simulations are necessary for which observations
          Dimension : (n%simgeo,n%obsgeo)
             Origin : make_occusim@modgeo_m
mw()%smw()%w1
               real : begin wavenumber of internal sub-mw [cm-1]
             Origin : ini_sub_mw@inismw_m
mw()%smw()%w2
               real : end wavenumber of internal sub-mw  [cm-1]
             Origin : ini_sub_mw@inismw_m
mw()%smw()%w1_ils
               real : begin wavenumber of ils-extended internal sub-mw [cm-1]
             Origin : ini_sub_mw@inismw_m
mw()%smw()%w2_ils
               real : end wavenumber of ils-extended internal sub-mw [cm-1]
             Origin : ini_sub_mw@inismw_m
================================================================================

n%
       numbers_type : numbers
n%baselev
            integer : number of base levels
             Origin : input_main@input_m
n%inlev
            integer : number of input levels
             Origin : input_main@input_m
n%lspeci
            integer : number of line-data species
             Origin : speciorder@input_m
n%modlev
            integer : number of model levels
             Origin : make_modelgrid@modlev_m
n%mw
            integer : number of microwindows
             Origin : input_mwdef@input_m
n%nlte_speci
            integer : number of nlte-species
             Origin : input_main@input_m
n%obsgeo
            integer : number of observation geometries
             Origin : input_main@input_m
n%simgeo
            integer : number of simulated geometries
             Origin : make_modelgeo@modlev_m
n%speci
            integer : number of species
             Origin : speciorder@input_m
n%tot_nlte_state
            integer : total number of nlte-states in input
             Origin : input_main@input_m
n%xspeci
```

```
             integer : number of cross-section data species
              Origin : speciorder@input_m
================================================================================

nlte()%
        non_lte_type : nlte definition
           Dimension : n%nlte_speci
nlte()%igasiso
             integer : nlte species identifier (10*mo+iso)
              Origin : input_main@input_m
nlte()%istate()
             integer : hitran vibrational state number
           Dimension : nlte()%nstate
              Origin : input_main@input_m
nlte()%itrans()
             integer : transition
           Dimension : (nlte()%ntrans,2)
              Origin : input_main@input_m
nlte()%model_speci
             integer : =1 if the species is a nlte-model species
                       =0 if not
              Origin : input_main@input_m
nlte()%nstate
             integer : number of vibrational states
              Origin : input_main@input_m
nlte()%ntrans
             integer : number of transitions for which
                       nlte should be considered
              Origin : input_main@input_m
================================================================================

obs%
   obs_geometry_type : observation geometry
obs%alt()
               real : observer altitude
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%angle()
               real : nadir/elevation angles [rad]
           Dimension : n%obsgeo
              Origin : input_main@input_m, make_modelgeo@modgeo_m,
                       raytrace_ctrl@rayctl_m
obs%aziview()
               real : viewing azimuth angle [rad] (south=0,
                       direction S->O->N->W)
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%lat()
               real : latitudes of obverver/tangent points [rad]
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%length
               real : path length for homogeneous path
                       calculation [km]
```

```
              Origin : input_main@input_m
obs%lon()
                real : longitude of obverver/tangent points [rad]
                       (increasing in easterly direction)
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%Tback
                real : background temperature
                       (if negative: only transmission calculation)
              Origin : input_main@input_m
obs%ztang()
                real : tangent altitudes [km]
           Dimension : n%obsgeo
              Origin : input_main@input_m, raytrace_ctrl@rayctl_m
================================================================================

outdat%
        outdata_type : output data vector
outdat%n_mw
             integer : number of microwindows
              Origin : ini_output@iniout_m
outdat%wdel
                real : wavenumber grid distance
              Origin : ini_output@iniout_m

outdat%mw()%
          outmw_type : output data for each microwindow
           Dimension : n%mw
outdat%mw()%n_geo
             integer : number geometries
              Origin : ini_output@iniout_m

outdat%mw()%geo()%
         outgeo_type : output data for each geometry
           Dimension : outdat%mw()%n_geo
outdat%mw()%geo()%n_smw
             integer : number of external sub-microwindows for each geometry
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%
            out_type : output data for each sub-microwindow
           Dimension : outdat%mw()%geo()%n_smw
outdat%mw()%geo()%smw()%dspec_dapo()
                real : derivative of coarse grid
                       spectrum with respect to linear apodisation
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dlos()
                real : derivative of coarse grid
                       spectrum with respect to line of sight
                       elevation angle
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_doff()
```

```
                real : derivative of coarse grid
                       spectrum with respect to offset
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dpha()
                real : derivative of coarse grid
                       spectrum with respect to phase
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dsca()
                real : derivative of coarse grid
                       spectrum with respect to scale
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dshift()
                real : derivative of coarse grid
                       spectrum with respect to shift
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%n_pts
             integer : number of spectral points
outdat%mw()%geo()%smw()%spec()
                real : spectrum on coarse grid
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%w1
                real : first wavenumber of output spectrum
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_daerabs()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to continuum
                       absorption coefficient
           Dimension : para%n_aerabs_para
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_daerabs()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dbrdaeronum()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to broadband aerosol
                       number density coefficients
           Dimension : para%n_brdaeronum_para
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dMie(:,:)%
          dspec_type : derivative of coarse grid
                       spectrum with respect to Mie parameters
```

```
           Dimension : brdaero%nMie,para%n_Mie_para
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dp()%
         dspec_type : derivative of coarse grid
                      spectrum with respect to pressure
           Dimension : para%n_p_para
outdat%mw()%geo()%smw()%dspec_dp()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dT()%
         dspec_type : derivative of coarse grid
                      spectrum with respect to temperature
           Dimension : para%n_T_para
outdat%mw()%geo()%smw()%dspec_dT()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dTgrad()%
         dspec_type : derivative of coarse grid
                      spectrum with respect to temperature-gradient
           Dimension : para%n_Tgrad_para
outdat%mw()%geo()%smw()%dspec_dTgrad()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dTvib()%
         dspec_type : derivative of coarse grid
                      spectrum with respect to vibrational temperature
           Dimension : (para%n_Tvib_para_max,para%n_nlte_state)
outdat%mw()%geo()%smw()%dspec_dTvib()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dvmr()%
         dspec_type : derivative of coarse grid
                      spectrum with respect to vmr
           Dimension : (para%n_vmr_para_max,para%n_vmr_speci)
outdat%mw()%geo()%smw()%dspec_dvmr()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dvmrgrad()%
         dspec_type : derivative of coarse grid
```

```
                        spectrum with respect to vmr-gradient
            Dimension : (para%n_vmrgrad_para_max,para%n_vmrgrad_speci)
outdat%mw()%geo()%smw()%dspec_dvmrgrad()%spec()
                 real : derivative spectra
            Dimension : outdat%mw()%geo()%smw()%n_pts
               Origin : ini_output@iniout_m
outdat%dangle_dztang()
                 real : derivative of nadir angle wrt tangent altitude [rad/km]
            Dimension : n%obsgeo
               Origin : ini_output@iniout_m
outdat%ztang()
                 real : tangent altitudes
            Dimension : n%obsgeo
               Origin : ini_output@iniout_m
================================================================================

outheader
       character(80) : header for output files
               Origin : input_main@input_m
================================================================================

para%
        retpara_type : retrieval parameters
para%alt_aerabs()
                 real : altitudes of continuum paramete5rs
            Dimension : para%n_aerabs_para
               Origin : ini_para_aerabs@inipar_m
para%brdaeronum()
                 real : parameters for broadband aerosol number density
            Dimension : para%n_brdaeronum_para
               Origin : ini_para_num@inipar_m
para%i_nlte_iso
              integer : global nlte_isotope number
                        for which Tvib derivatives are calculated
               Origin : ini_para_Tvib@inipar_m
para%i_nlte_speci
              integer : global species number for which Tvib
                        derivatives are calculated
               Origin : ini_para_Tvib@inipar_m
para%i_nlte_state()
              integer : pointer to the global nlte state number
            Dimension : para%n_nlte_state
               Origin : ini_para_Tvib@inipar_m
para%i_vmrgrad_speci()
              integer : pointer to global species numbering
            Dimension : para%n_vmrgrad_speci
               Origin : ini_para_vmrgrad@inipar_m
para%i_vmr_speci()
              integer : pointer to global species numbering
            Dimension : para%n_vmr_speci
               Origin : ini_para_vmr@inipar_m
para%Mie(:,:)
                 real : Mie-parameter parameters
            Dimension : (brdaero%nMie,para%n_Mie_para)
```

```
              Origin : ini_para_Mie@inipar_m
para%n_aerabs_para
             integer : number of aerosol abs.coef. parameters
              Origin : ini_para_aerabs@inipar_m
para%n_brdaeronum_para
             integer : number of broadband aerosol number density parameters
              Origin : ini_para_num@inipar_m
para%nlte_speci
             integer : pointer to the nlte-species of nlte()%..input variable
              Origin : ini_para_Tvib@inipar_m
para%n_Mie_para
             integer : number of Mie parameters
              Origin : ini_para_Mie@inipar_m
para%n_nlte_state
             integer : number of T-vib states for which derivatives
                       should be calculated
              Origin : ini_para_Tvib@inipar_m
para%n_p_para
             integer : number of p parameters
              Origin : ini_para_p@inipar_m
para%n_Tgrad_para
             integer : number of T gradient parameters
              Origin : ini_para_Tgrad@inipar_m
para%n_T_para
             integer : number of T parameters
              Origin : ini_para_T@inipar_m
para%n_Tvib_para_max
             integer : maximum number of Tvib parameters
              Origin : ini_para_Tvib@inipar_m
para%n_vmrgrad_speci
             integer : number of species for which
                       vmr-gradient is retrieved
              Origin : ini_para_vmrgrad@inipar_m
para%n_vmrgrad_para_max
             integer : maximum number of vmr-gradient parameters
              Origin : ini_para_vmrgrad@inipar_m
para%n_vmr_para_max
             integer : maximum number of vmr parameters
              Origin : ini_para_vmr@inipar_m
para%n_vmr_speci
             integer : number of species for which vmr is retrieved
              Origin : ini_para_vmr@inipar_m
para%p()
                real : parameters for p profile
           Dimension : para%n_p_para
              Origin : ini_para_p@inipar_m
para%T()
                real : parameters for T profile
           Dimension : para%n_T_para
              Origin : ini_para_T@inipar_m
para%Tgrad()
                real : parameters for T gradient profiles
           Dimension : para%n_Tgrad_para
              Origin : ini_para_Tgrad@inipar_m
```

```
para%mw()%
     paraaerabs_type : microwindows for each of which an
                       aerosol abs.coef. profile is parametrized
           Dimension : n%mw
para%mw()%aerabs()
                real : parameters for aerosol absorption coef. profile
           Dimension : para%n_aerabs_para
              Origin : ini_para_aerabs@inipar_m

para%speci()%
        paravmr_type : species of vmr parameters (index is on
                       global species numbering !!)
           Dimension : n%speci
para%speci()%n_vmrgrad_para
             integer : number of vmr-gradient parameters
              Origin : ini_para_vmrgrad@inipar_m
para%speci()%n_vmr_para
             integer : number of vmr parameters
              Origin : ini_para_vmr@inipar_m
para%speci()%vmr()
                real : parameters for vmr profile
           Dimension : para%speci()%n_vmr_para
              Origin : ini_para_vmr@inipar_m
para%speci()%vmrgrad()
                real : parameters for vmr-gradient profile
           Dimension : para%speci()%n_vmrgrad_para
              Origin : ini_para_vmrgrad@inipar_m

para%state()%
       paranlte_type : nlte-states (index is on global state numbering
           Dimension : para%n_nlte_state
para%state()%n_Tvib_para
             integer : number of Tvib parameters
              Origin : ini_para_Tvib@inipar_m
para%state()%Tvib()
                real : parameters for Tvib profile
           Dimension : para%state()%n_Tvib_para
              Origin : ini_para_Tvib@inipar_m
================================================================================

Sails%
        outdata_type : spectrum data vector with ails (and fov)
                       convolved spectrum and derivatives
                       with internal sub-microwindow indexing
Sails%n_mw
             integer : number of microwindows
              Origin : alloc_Sails@radtra_m
Sails%wdel
                real : wavenumber grid distance
              Origin : alloc_Sails@radtra_m

Sails%mw()%
          outmw_type : data for each microwindow
```

```
           Dimension : n%mw
Sails%mw()%n_geo
             integer : number geometries
              Origin : alloc_Sails@radtra_m

Sails%mw()%geo()%
         outgeo_type : data for each geometry
           Dimension : Sails%mw()%n_geo
Sails%mw()%geo()%n_smw
             integer : number of external sub-microwindows for each geometry
              Origin : alloc_Sails@radtra_m
Sails%mw()%geo()%l_smw()
             logical : occupation vector, which internal sub-mw
                       belongs to each geometry
           Dimension : Sails%mw()%geo()%n_smw
              Origin : alloc_Sails@radtra_m

Sails%mw()%geo()%smw()%
            out_type : output data for each sub-microwindow
           Dimension : Sails%mw()%geo()%n_smw
Sails%mw()%geo()%smw()%dspec_dapo()
                real : derivative of coarse grid
                       spectrum with respect to linear apodisation
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
Sails%mw()%geo()%smw()%dspec_dlos()
                real : derivative of coarse grid
                       spectrum with respect to line of sight
                       elevation angle
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
Sails%mw()%geo()%smw()%dspec_doff()
                real : derivative of coarse grid
                       spectrum with respect to offset
           Dimension : Sails%mw()%smw()%n_pts
              Origin : off_deri@offsca_m
Sails%mw()%geo()%smw()%dspec_dpha()
                real : derivative of coarse grid
                       spectrum with respect to phase
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
Sails%mw()%geo()%smw()%dspec_dsca()
                real : derivative of coarse grid
                       spectrum with respect to scale
           Dimension : Sails%mw()%smw()%n_pts
              Origin : sca_deri@offsca_m
Sails%mw()%geo()%smw()%dspec_dshift()
                real : derivative of coarse grid
                       spectrum with respect to shift
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
```

```
              Modify : sca@offsca_m
Sails%mw()%geo()%smw()%n_pts
             integer : number of spectral points
              Origin : alloc_Sails@radtra_m
Sails%mw()%geo()%smw()%spec()
                real : spectrum on coarse grid
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m, off@offsca_m
Sails%mw()%geo()%smw()%w1
                real : first wavenumber of output spectrum
              Origin : alloc_Sails@radtra_m

Sails%mw()%geo()%smw()%dspec_daerabs()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to continuum
                       absorption coefficient
           Dimension : para%n_aerabs_para
Sails%mw()%geo()%smw()%dspec_daerabs()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dbrdaeronum()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to broadband aerosol
                       number density coefficients
           Dimension : para%n_brdaeronum_para
              Origin : ini_output@iniout_m
Sails%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

Sails%mw()%geo()%smw()%dspec_dMie(:,:)%
          dspec_type : derivative of coarse grid
                       spectrum with respect to Mie parameters
           Dimension : brdaero%nMie,para%n_Mie_para
              Origin : ini_output@iniout_m
Sails%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

Sails%mw()%geo()%smw()%dspec_dp()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to pressure
           Dimension : para%n_p_para
Sails%mw()%geo()%smw()%dspec_dp()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
```

```
Sails%mw()%geo()%smw()%dspec_dT()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to temperature
           Dimension : para%n_T_para
Sails%mw()%geo()%smw()%dspec_dT()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dTgrad()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to temperature-gradient
           Dimension : para%n_Tgrad_para
Sails%mw()%geo()%smw()%dspec_dTgrad()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dTvib()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vibrational temperature
           Dimension : (para%n_Tvib_para_max,para%n_nlte_state)
Sails%mw()%geo()%smw()%dspec_dTvib()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dvmr()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vmr
           Dimension : (para%n_vmr_para_max,para%n_vmr_speci)
Sails%mw()%geo()%smw()%dspec_dvmr()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dvmrgrad()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vmr-gradient
           Dimension : (para%n_vmrgrad_para_max,para%n_vmrgrad_speci)
Sails%mw()%geo()%smw()%dspec_dvmrgrad()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
================================================================================

sim%
   sim_geometry_type : simulated geometries
```

```
sim%alt()
                real : observer altitude
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%angle()
                real : nadir angles [rad]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m, raytrace_ctrl@rayctl_m
sim%aziview()
                real : viewing azimuth angle [rad]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%dangle_dztang()
                real : derivative of nadir angle with respect to tangent
                       altitude [rad/km]
           Dimension : n%simgeo
              Origin : dangle_dztang@rayctl_m
sim%lat()
                real : latitudes of obverver/tangent points [rad]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%length
                real : path length for homogeneous path calculation [km]
              Origin : make_modelgeo@modgeo_m
sim%lobs()
             logical : =T if simulation is also an observation
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%lon()
                real : latitudes of obverver/tangent points [rad]
                       (increasing in easterly direction)
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%ztang()
                real : tangent altitudes [km]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m, calc_ztang@modlev_m,
                       raytrace_ctrl@rayctl_m
================================================================================

speci()%
          speci_type : species definition
           Dimension : n%speci
speci()%hwhm_f_max
                real : maximum air-broadened half width [cm-1]
              Origin : readlines@inspec_m
speci()%hwhm_f_min
                real : minimum air-broadened half width [cm-1]
              Origin : readlines@inspec_m
speci()%isoprof
             integer : profile identifier (hitran iso number) or
                       (0) for all isotopes not included
                       in another species
              Origin : speciorder@input_m
```

```
speci()%itotiso()
             integer : isotope numbers the species consists of
           Dimension : mxiso
              Origin : speciorder@input_m
speci()%mo
             integer : hitran molecule number
              Origin : speciorder@input_m
speci()%n_nlte_iso
             integer : number of nlte isotopes
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%ntotiso
             integer : total number of isotopes the species
                       consists of
              Origin : speciorder@input_m

speci()%cross%
      x_section_type : cross-section data
speci()%cross%n
             integer : original gasnumber belonging to each of
                       the n%xspeci xsection gases
              Origin : make_x@xinput_m
speci()%cross%nxmw
             integer : number of (internal x-section) mws
              Origin : make_x@xinput_m

speci()%cross%xmw()%
            xmw_type : internal x-section microwindows
           Dimension : speci()%cross%nxmw
speci()%cross%xmw()%dw
                real : wavenumber grid point distance
              Origin : make_x@xinput_m
speci()%cross%xmw()%mw
             integer : original mw belonging to each
                       internal x-section mw
              Origin : make_x@xinput_m
speci()%cross%xmw()%mw_orbit
             integer : original mw belonging to each
                       internal x-section mw in orbit (DFD)
              Origin :
speci()%cross%xmw()%ngrid
             integer : number of wavenumber grid points
              Origin : make_x@xinput_m
speci()%cross%xmw()%npT
             integer : number of x-section p-T measurements
              Origin : make_x@xinput_m
speci()%cross%xmw()%p
                real : pressure of each x-section p-T measurement [hPa]
           Dimension : speci()%cross%xmw()%npt
              Origin : make_x@xinput_m
speci()%cross%xmw()%T
                real : temperature of each x-section p-T measurement [K]
           Dimension : speci()%cross%xmw()%npT
              Origin : make_x@xinput_m
speci()%cross%xmw()%w1
```

```
                real : begin wavenumb. of each intern. xsct. mw
              Origin : make_x@xinput_m
speci()%cross%xmw()%w2
                real : end wavenumb. of each intern. xsct. mw
              Origin : make_x@xinput_m
speci()%cross%xmw()%x(:,:)
                real : measured cross sections for
                       each wavenuber grid point
                       and each pT measurement
           Dimension : speci()%cross%xmw()%ngrid,
                       speci()%cross%xmw()%npT
              Origin : readx@xinput_m

speci()%iso()%
           iso_type2 : isotope (spectroscopic data tree)
           Dimension : 0:speci()%n_nlte_iso
speci()%iso()%i
             integer : hitran isotope number
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%nlte_speci
             integer : pointer to the nlte-species of nlte()%..
                       input variable
              Origin : alloc_nlte@inspec_m
speci()%iso()%n_nlte_bands
             integer : number of nlte bands
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m

speci()%iso()%band()%
          band_type : band (spectroscopic data tree)
          Dimension : 0:speci()%iso()%n_nlte_bands
speci()%iso()%band()%ivl
             integer : lower state global quanta index for nlte band
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%ivl_state
             integer : pointer to the nlte_state for lower state
                       (=0 for ground state)
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%ivu
             integer : upper state global quanta index for nlte band
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%ivu_state
             integer : pointer to the nlte_state for upper state
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%n_lm_branches
             integer : number of line-mixing branches
              Origin : allocno_lm@inspec_m, alloc_lm@inspec_m

speci()%iso()%band()%branch()%
        branch_type : branch (spectroscopic data tree)
          Dimension : 0:speci()%iso()%band()%n_lm_branches
speci()%iso()%band()%branch()%b0_rot
                real : rotational moment of molecule
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
```

```
speci()%iso()%band()%branch()%coefrm_f()
                real : coef. of a-T parametrisation (air)
           Dimension : (3,4)
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%coefrm_s()
                real : coef. of a-T parametrisation (self)
           Dimension : (3,4)
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%mw_l1()
             integer : first line for each mw
           Dimension : n%mw
              Origin : pointmw@inspec_m
speci()%iso()%band()%branch()%mw_l2()
             integer : last line for each mw
           Dimension : n%mw
              Origin : pointmw@inspec_m
speci()%iso()%band()%branch()%mw_orbit_l1()
             integer : first line for each mw in orbit
           Dimension : n%mw (for orbit DFD)
              Origin :
speci()%iso()%band()%branch()%mw_orbit_l2()
             integer : last line for each mw in orbit
           Dimension : n%mw (for orbit DFD)
              Origin :
speci()%iso()%band()%branch()%nlines
             integer : number of lines
              Origin : allocopy1@inspec_m, allocopy2@inspec_m,
                       alloc_lm@inspec_m
              Modify : delete_lmlines@inspec_m
speci()%iso()%band()%branch()%symprop_f()
                real : beta factors
           Dimension : 2
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%symprop_s()
                real : beta factors
           Dimension : 2
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m

speci()%iso()%band()%branch()%line()%
           line_type : line specific data
           Dimension : speci()%iso()%band()%branch()%nlines
speci()%iso()%band()%branch()%line()%elow
                real : lower state energy [cm-1]
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%expt
                real : coef. of T-dependence of air-broadened halfwidth
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%hwhm_f
```

```
                real : air-broadened half width [cm-1]
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%hwhm_s
                real : self-broadened half width [cm-1]
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%iso
             integer : hitran isotope number
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%ivl
             integer : lower state global quanta index
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%ivu
             integer : upper state global quanta index
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%lnsint
                real : log(line intensity)
              Origin : (readlines@inspec_m, isoabun@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%mo
             integer : hitran molecule number
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%pshift
                real : pressure shift
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%ql
         character(9) : lower state local quanta
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%qu
         character(9) : upper state local quanta
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%sint
                real : line intensity  [cm-1/(molec*cm-2)] multiplied
                       by T-idependent part for optimized calculation
                       (sint * exp( hck * elow / T0hit) /
                       (1 - exp( -hck * w / T0hit)) * q296
              Origin : (readlines@inspec_m, read_lmdata@inspec_m,
                       isoabun@inspec_m), input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%w
                real : line wavenumber [cm-1]
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m

speci()%iso()%band()%branch()%lmline()%
          lmline_type : line-mixing spect.data
            Dimension : speci()%iso()%band()%branch()%nlines
```

```
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%coefy_f()
                real : coef. of y T-parametrisation (air-broad.)
           Dimension : 4
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%coefy_s()
                real : coef. of y T-parametrisation (self-broad.)
           Dimension : 4
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%csym
        character(2) : symmetry of i and f state
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%jrot
             integer : rotational quantum number
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%trans
                real : transition probability =sqrt(hitran_R)
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
================================================================================

sw%
         switch_type : switches
sw%addnoise
             logical : =.true. if noise should be added to the spectrum
              Origin : input_main@input_m
sw%alloc_deri
             integer : switch for allocation grade of deri% variable
                        =0  partly allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_geo
             integer : switch for allocation grade of geo% variable
                        =0  not allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_modprof
             integer : switch for allocation grade of modprof% variable
                        =0  not allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_mw
             integer : switch for allocation grade of mw% variable
                        =0  partly allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_outdat
             integer : switch for allocation grade of outdat% variable
                        =0  not allocated
```

```
                    =1  totally allocated
             Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_Sails
            integer : switch for allocation grade of Sails% variable
                    =0  not allocated
                    =1  totally allocated
sw%alloc_sim
            integer : switch for allocation grade of sim% variable
                    =0  not allocated
                    =1  totally allocated
sw%baselev
            integer : switch for base-levels defining the
                    layering for the forward calculation
                    0= the input-profile levels are used exclusively
                    1= the input-profile levels are used and additional
                        levels with respect to criteria 7.3-7.6 are added
                    2= the levels under $7.32 are used exclusively
                    3= the levels under $7.32 are used and additional
                        levels with respect to criteria 7.3-7.6 are added
                    4= the levels are set up automatically with respect
                        to criteria 7.3-7.6
             Origin : input_main@input_m
sw%brdaero
            logical : consideration of broadband-aerosol
             Origin : input_main@input_m
sw%deriaer
            logical : continuum derivative
             Origin : input_main@input_m
sw%deriapo
            logical : linear apodisation derivative
             Origin : input_main@input_m
sw%derilos
            logical : line of sight derivative
             Origin : input_main@input_m
sw%deriMie
            logical : Mie parameter derivative
             Origin : input_main@input_m
sw%derinum
            logical : number density of broadband-aerosol derivative
             Origin : input_main@input_m
sw%derioff
            logical : offset derivative
             Origin : input_main@input_m
sw%derip
            logical : pressure derivative
             Origin : input_main@input_m
sw%deripha
            logical : phase derivative
             Origin : input_main@input_m
sw%derisca
            logical : scale derivative
             Origin : input_main@input_m
sw%derishift
            logical : wavenumber shift derivative
```

```
              Origin : input_main@input_m
sw%deriT
             logical : temperature derivative
              Origin : input_main@input_m
sw%deriTgrad
             logical : Temperature gradient derivative
              Origin : input_main@input_m
sw%deriTvib
             logical : Tvib derivative
              Origin : input_main@input_m
sw%derivmr
             logical : vmr derivative
              Origin : input_main@input_m
sw%derivmrgrad
             logical : vmr gradient derivative
              Origin : input_main@input_m
sw%derix
             logical : nlte-model parameter derivative
              Origin : input_main@input_m
sw%firstrun
             logical : =.true. for first time run of forward-model
              Origin : kopra
sw%fov
             logical : field of view calculation
              Origin : input_main@input_m
sw%fovils
             logical : field of view effect on instrumetal profile
              Origin : input_main@input_m
sw%fq_equ1
             logical : =.true.: fq set to 1.0
              Origin : input_main@input_m
sw%horigrad
             logical : horizontal gradient calculation
              Origin : input_main@input_m
sw%horigrad_p
             logical : horizontal gradient calculation for p
              Origin : input_main@input_m
sw%horigrad_T
             logical : horizontal gradient calculation for T
              Origin : input_main@input_m
sw%horigrad_Tvib
             logical : horizontal gradient calculation for Tvib
              Origin : input_main@input_m
sw%horigrad_vmr
             logical : horizontal gradient calculation for vmr
              Origin : input_main@input_m
sw%hydrostat
             integer : hydrostatic equilibrium
                       (0=no,1=level-pressures,2=level-altitudes)
              Origin : input_main@input_m
sw%ilscalc
             integer : mode of ils calculation
                       1= circular aperture with phase and lin. apo. errror
                       2= ESA parametrization
```

```
                    3= read in for each microwindow
              Origin : input_main@input_m
sw%mainspeci
             integer : species-number of main gas
                       (the Curtis-Godson T of which will be
                       used for the Planck function)
                       >1 = C-G T of this species will be used
                       0  = no main gas: mix C_G T's of all gases
                       -1 = C-G T of air will be used
              Origin : make_mainspeci@input_m
sw%Miemod
             logical : should the Mie-model be used for broadband-aerosol
              Origin : input_main@input_m
sw%mix
             logical : line mixing
              Origin : input_main@input_m
sw%mix_ddros
             integer : direct diagonalisation (1), Rosenkranz (2)
              Origin : input_main@input_m
sw%mix_qpr
             integer : only Q- (1), Q- and PR- coupling (2)
              Origin : input_main@input_m
sw%mode_obs
             integer : mode of observation
                        1= satellite / limb / tangent altitude
                        2= satellite / limb / nadir angle and observer altitude
                        3= balloon / limb / tangent altitude and observer altitude
                        4= balloon / limb / nadir angle and observer altitude
                        5= upward / nadir angle and observer altitude
                        6= upward+limb / nadir angle and observer altitude
                        7= homogeneous path (cuvette)
              Origin : input_main@input_m
sw%new_absco
             logical : absorption coefficients should be determined new
              Origin : kopra
sw%new_modelgeo
             logical : model geometries should be determined new
              Origin : kopra
sw%new_modelgrid
             logical : model levels should be determined new
              Origin : kopra
sw%nlte
             logical : nlte calculation
              Origin : input_main@input_m
sw%nlte_model
             logical : nlte-model calculation
              Origin : input_main@input_m
sw%outcoarse
             logical : switch for output on coarse grid
              Origin : input_main@input_m
sw%outfine
             integer : switch for output on fine grid
                       0 = no output on fine grid
                       1 = output on non-equidistant fine grid
```

```
                    2 = output on equidistant fine grid
             Origin : input_main@input_m
sw%paraaer
            logical : use continuum parameters
             Origin : input_main@input_m
sw%paraapo
            logical : use linear apodisation parameters
             Origin : input_main@input_m
sw%paralos
            logical : use line of sight parameters
             Origin : input_main@input_m
sw%paraMie
            logical : use Mie model parameters
             Origin : input_main@input_m
sw%paranum  logical : use broadband-aerosol number density parameters
             Origin : input_main@input_m
sw%paraoff
            logical : use offset parameters
             Origin : input_main@input_m
sw%parap
            logical : use pressure parameters
             Origin : input_main@input_m
sw%parapha
            logical : use phase parameters
             Origin : input_main@input_m
sw%parasca
            logical : use scale parameters
             Origin : input_main@input_m
sw%parashift
            logical : use wavenumber shift parameters
             Origin : input_main@input_m
sw%paraT
            logical : use temperature parameters
             Origin : input_main@input_m
sw%paraTgrad
            logical : use Temperature gradient parameters
             Origin : input_main@input_m
sw%paraTvib
            logical : use Tvib parameters
             Origin : input_main@input_m
sw%paravmr
            logical : use vmr parameters
             Origin : input_main@input_m
sw%paravmrgrad
            logical : use vmr gradient parameters
             Origin : input_main@input_m
sw%parax
            logical : use nlte-model parameter derivatives
             Origin : input_main@input_m
sw%rs_nlte
            logical : calculate rot/spin nlte
             Origin : input_main@input_m
sw%testout
            integer : test output grade
```

```
              Origin : input_main@input_m
sw%weighting_fct
             logical : calculation of weighting functions
              Origin : input_main@input_m
================================================================================

wgrid%
       wavegrid_type : wavenumber grid
wgrid%coarse
                real : coarse grid [cm-1]
              Origin : input_main@input_m
wgrid%fine
                real : fine grid [cm-1]
              Origin : input_main@input_m
wgrid%iratio
                real : ratio of fine and coarse grid
              Origin : input_main@input_m
================================================================================
```

Graphical user interface

M. Linder

Abstract: The graphical user interfaces *kopragui* and *specplot* are described. *kopragui* allows to edit *kopra* input files, start *kopra* jobs, and plot spectra and input profiles. *specplot* allows to comfortably and flexibly plot spectra.

1 *kopragui* - a useful tool for kopra users

1.1 Description

kopragui is a graphical user interface developed to support and simplify the use of *kopra* (**K**arlsruhe **o**ptimized and **p**recised **r**adiative transfer **a**lgorithm). The program offers the following opportunities:

- **Inspection, creation and modification of *kopra* input files.**

  Parameters of *kopra* input files can be inspected. As there are dependences between some input variables, it can be difficult to edit a *kopra* input file. The modification of a variable possibly implies changes of other variables. Especially, if extensive input files are to be created or modified, it might be difficult to keep an overview on the correctness of all parameters. *kopragui* allows to edit input files in an easy way. The user is lead through windows with all the input variables he has to enter. The values are checked for correctness, and, where possible, for plausibility.

- **Start of kopra**

  kopra can be executed as batch job in the *lsf* cluster. The *lsf* host can be selected automatically by *lsf* or user-defined. The user is informed about the result of the job by email.

- **Plot of spectra**

  spectra as produced by *kopra* and the residual difference of two spectra can be plotted.

- **Plot and manipulation of profiles**

  kopra needs several profiles as input, such as temperature, pressure, or gas vmr profiles. These profiles have an important influence on the creation of spectra by *kopra*. *kopragui* allows to plot one, or compare several profiles and to print them.
  Pressure, temperature, and vmr profiles can be manipulated sectionwise by arithmetic operations. In this way, profiles can be scaled and/or shifted by an additive offset.

- **Information**

  The *Info* button offers information about the use of *kopragui*, the latest modifications in *kopragui*, the location of a web site with informations about *kopra* and about the author.

1.2 Installation and call

kopragui has been developed in JAVA 1.2 under UNIX (SOLARIS 5.7). The program is delivered as compressed file. You need to have a version of JAVA's JDK 1.2 installed. (see http://java.sun.com/)

1.2.1 Installation

You need 1 MB free disk space to install *kopragui*. Install *kopragui* in the following way:

- Copy *kopragui.tar.gz* (788 361 KB) e.g into your home directory.
- `gunzip` kopragui.tar.gz

- `tar` xvf kopragui.tar
  A directory *kopragui* has been created in your working directory.
- `rm` kopragui.tar
- Insert the path of your kopragui installation directory in the shell script *kopragui/kopragui*:
  - `cd` kopragui
  - Load the file *kopragui* in a text editor and insert the installation directory:

    ```
    set kopragui_path= ...
    ```
- Insert the path of your JDK 1.2 directory in the same shell script:

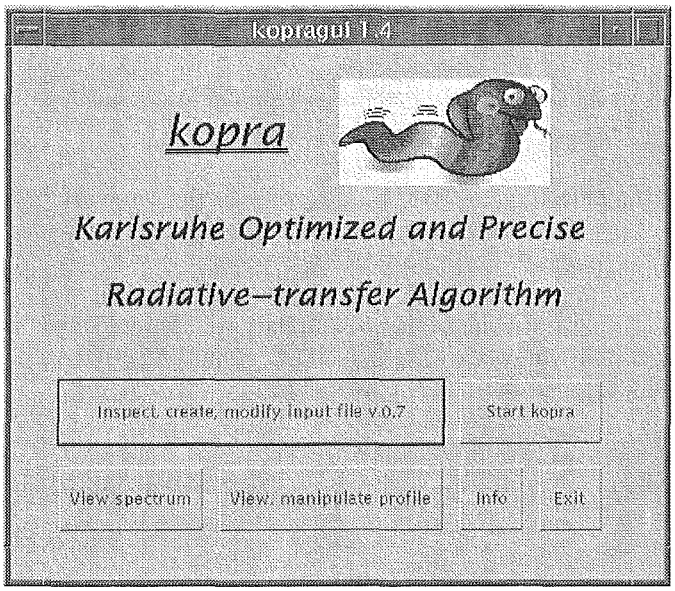
  ```
  set java_path= ...
  ```
- Insert the path of your *kopragui* installation directory in the path variable of your *.cshrc*, if you work with C shell or *.profile*. Alternatively, you can create the following link in your *bin*directory:

  ```
  ln -s <installation-directory>/kopragui/kopragui
                                  <HOME-directory>/bin/kopragui
  ```
- `rehash`

After the installation you can call *kopragui* in any directory.

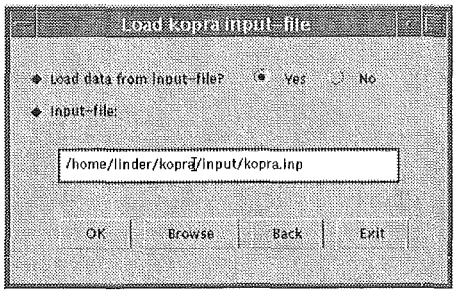
1.2.2 Call

After *kopragui* was called, the following window appears:



1.3 Inspection, creation and modification of *kopra* input files

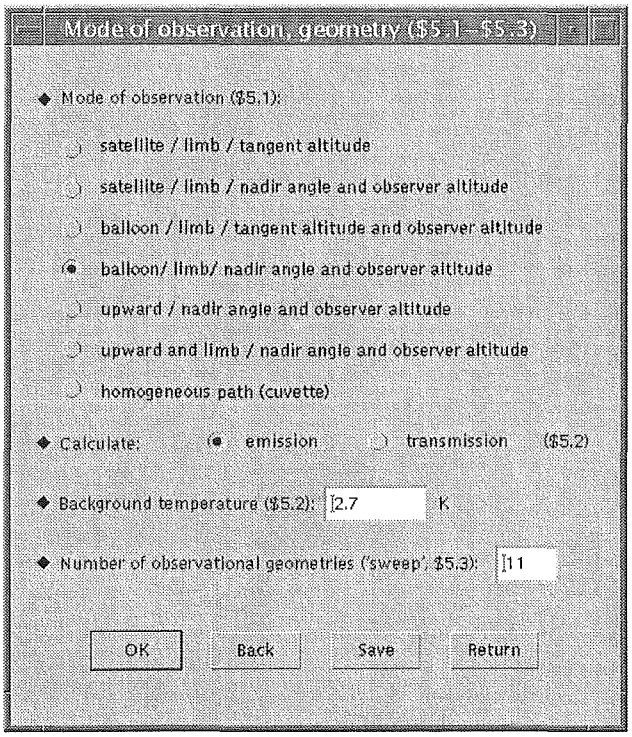
kopragui offers the opportunity to inspect the parameters of an existing *kopra* input file. Moreover a new input file can be created or an existing file can be modified. After pressing the button *Inspect, create, modify input file v.0.7*, the following window asks the user either to create a new *kopra* input file or to modify an existing one:



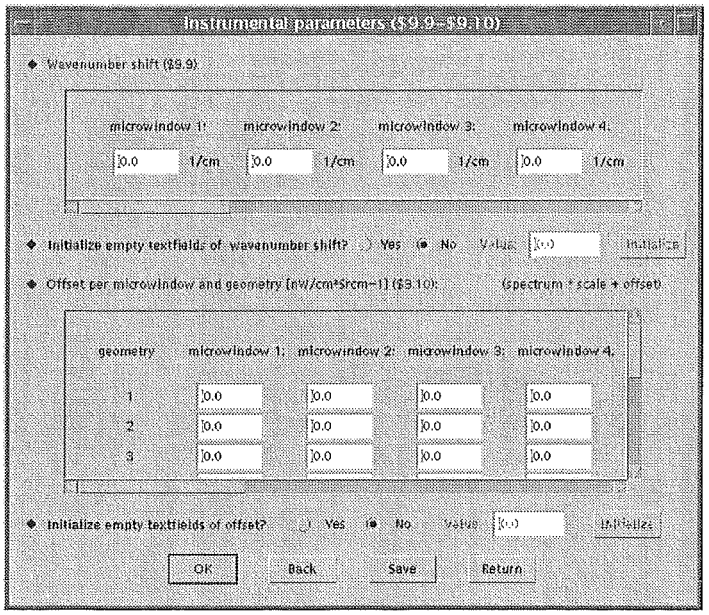
An input file which should be read can be of any version, but a file of version 0.7 is created (see paragraph *$* at the beginning of the file). In case of an error, a message with appropriate hints appears.

kopragui leads the user through a sequence of windows in which all necessary data have to be entered.

The window, shown below, is an example for such an input window. In the first option, the user has to choose the mode of observation, whether the measurement is made from a satellite, balloon, ground, ... This option corresponds to variable *$5.1* of the *kopra* input file. The next two options refer to the background temperature of the experiment (*$5.2*). If *emission* is selected, the background temperature has to be entered. The option *transmission* decides on the sign of *$5.2*. In the last textfield of this window, the user has to enter the number of observational geometries (*$5.3*).



The numerous vectors and matrices are integrated in panels with scrollbars. The following window refers to *instrumental parameters (\$9)*. First the vector *wavenumber shift per microwindow (\$9.9)* and then, the matrix *offset per microwindow and geometry (\$9.10)* has to be defined. Empty textfields can be initialized automatically with a user-defined value.



The data is checked for correctness, and, where it is useful, for plausibility. If an error occurs, a message with information helps the user to correct it. In some cases, *help* windows are available and can be called by pressing the *help* button. In most cases, the windows are self-explaining. The user is asked to enter only relevant data. Textfields for non-relevant data are blocked, and non-relevant windows don't appear.

At the beginning, the user can choose a window to start with. If he has to change a value for example refering to *computational accuracy (\$7)*, he can miss out the preceding windows and directly start editing the value which has to be modified.

The current state of the file can be saved from any window (button *Save*). In this case, the values belonging to the windows, which have not been confirmed by the *OK* button are **not** checked for errors.

The user has the possibility to go back from one window to the previous window by pressing the *Back* button.

It is always possible, to return to this selection window for choice of another input window or to quit (button *Return*).

If file names have to be entered, *kopragui* checks, whether these files exist and are

readable. A file browser supporting the search for files is available.

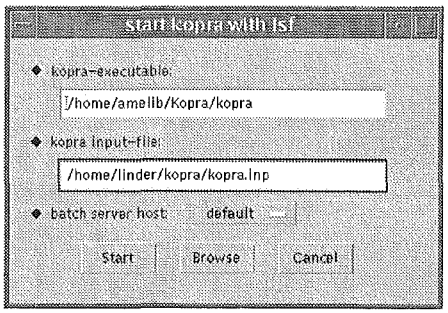
A date in the header for output files *($4.1)* is made actual automatically, if it is written in the correct format (window *Output files and directories*).

The newly created or modified *kopra* input file can be saved under a user-defined name. It has version *0.7*

1.4 Start of *kopra*

kopragui allows the user to start *kopra* with a given input file. If an input file has been edited recently, its file name is inserted automatically, but can also be changed. The *kopra* job is executed in the *lsf* cluster. The batch server can be selected automatically by *lsf* or defined by the user. The user is informed about the result of job by email as soon as the job has finished.

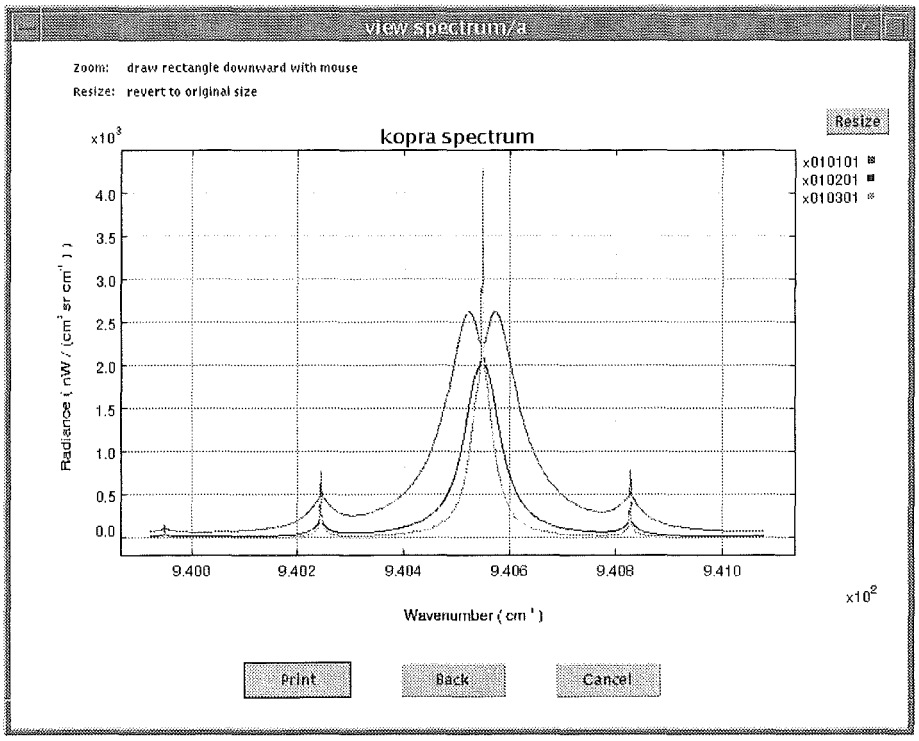
The next figure shows the control window for the execution of *kopra*:



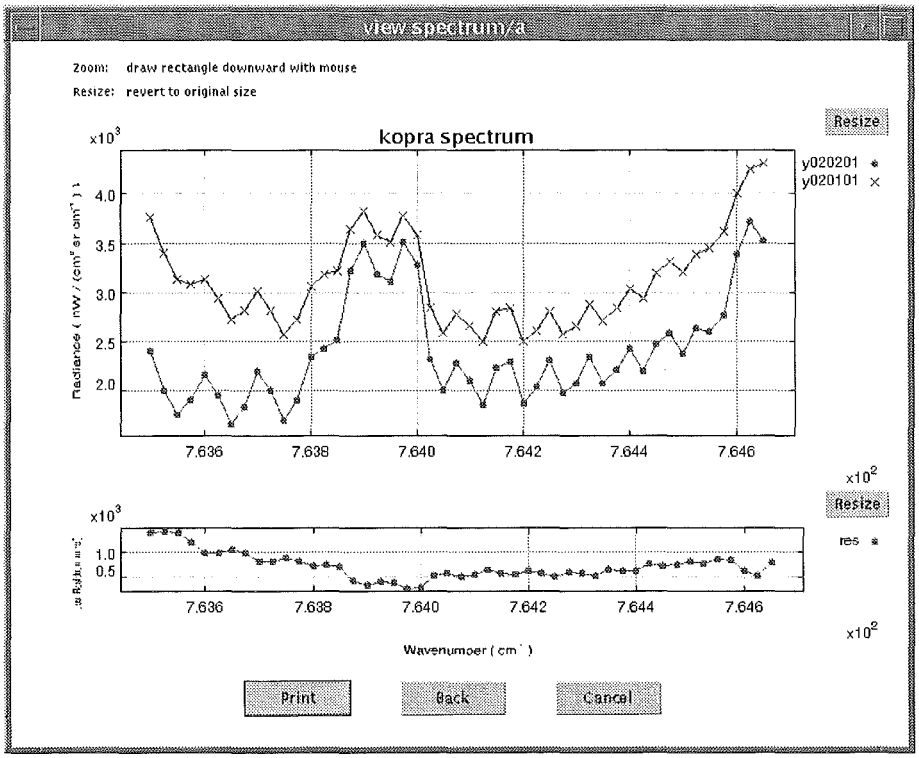
1.5 Plot of spectra

kopra spectra can be plotted and compared in a diagram. The user has the option to add the absolute or relative residual, if he wants to compare two spectra. The header of the plot and the curve labels can be edited. The axis labels can be chosen to be in german or english. The curves can be drawn with or without markers at each grid point. This makes it easier to distinguish several graphs in black and white prints. Parts of the spectrum/a can be zoomed interactively with the mouse.

The following figure shows three spectra in a single plot:



In the next diagram two spectra are compared and the absolute residual is added. The curves are plotted with markers.

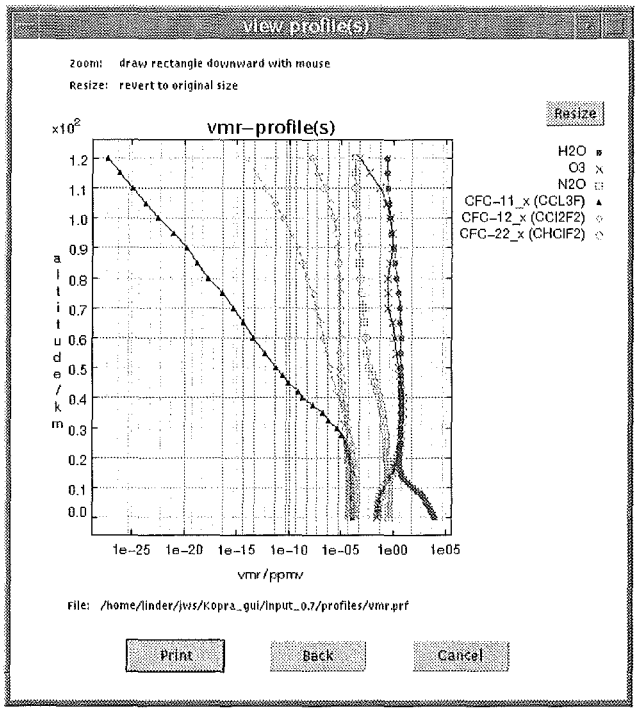


Alternatively, the IDL tool *specplot* can be called to plot *kopra* or *rfm* spectra.

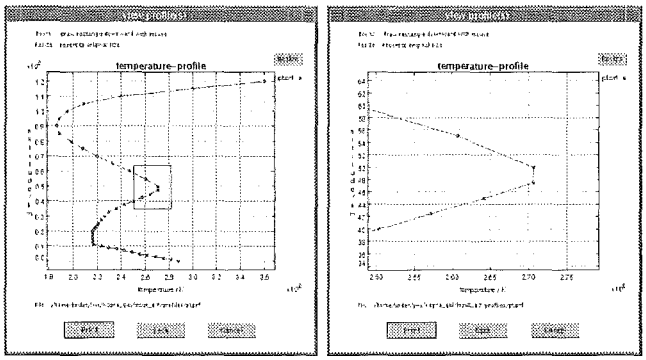
1.6 Plot of profiles

kopragui offers the opportunity to view one or serveral different profiles in a plot or to compare profiles of the same kind in different files.The profiles in a plot are drawn in different colors. Markers can be added at each grid point. Logarithmic axes can be chosen if it is useful.

In the following figure six different vmr gas profiles are plotted with logarithmic x-axis:



It is possible to zoom an area of the diagram by drawing a rectangle with a mouse button, as the next figure shows:



Profile plots can be printed or saved in postscript or encapsulated postscript format.

The following *kopra* profiles can be **plotted**:

- **pressure and temperature profiles**

  Several pressure or temperature profiles can be plotted and compared in a diagram. The pressure axis can be linear or logarithmic.

- **gas profiles**

  All gas profiles of a file can be plotted. Alternatively, gas profiles which are common to different files can be compared. The vmr axis can be linear or logarithmic.

- **vibrational temperature profiles**

  All vibrational temperatures of a file can be plotted in a diagramm. A kinetic temperature profile can be added to the plot.

- **aerosol absorption and scattering coefficient profiles**

  The aerosol absorption and scattering coefficients can be plotted for each microwindow and all altitude levels. Alternatively the profiles can be plotted for all microwindows and a specified altitude level. In this case, the corresponding wavenumbers are read from a *kopra* input file, which has to be specified.

- **isotope (relative) abundance profiles**

  The (relative) abundances of isotopes of a "line data" gas given in a file can be drawn in a plot, dependent on the altitude level. There is an option to extrapolate the graphs to given altitude borders. This may be useful if the abundances are given for one or only few altitude levels. The axis for the abundances can be linear or logarithmic.

- **pressure and temperature gradient profiles**

  The pressure and temperature latitude and longitude gradients can be plotted in a diagram. The pressure axis can be linear or logarithmic.

- **gas gradient profiles**

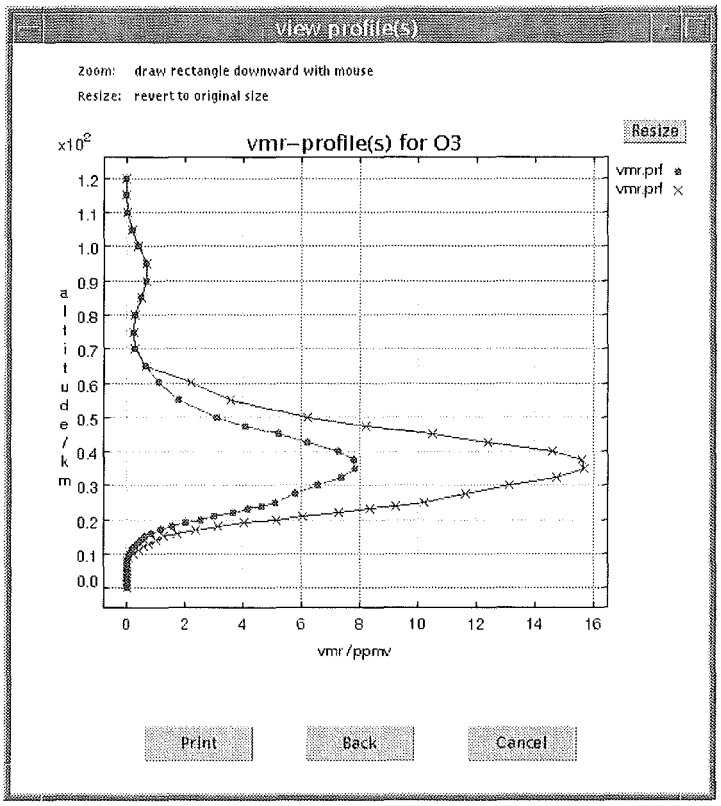
  The latitude and longitude gradients of a gas can be plotted.

- **vibrational temperature gradient profiles**

  The latitude and longitude vibrational temperature gradients of a state can be plotted.

It is possible to **manipulate** pressure, temperature and vmr profiles sectionwise with arithmetic operations +,-,*,/. Using this option, profiles can be scaled and/or shifted by an additive offset. The manipulated profiles can be saved.

The next figure, shows an O_3 profile which has been multiplied by 3 between 10 and 60 km and the original O_3 profile in a plot:



1.7 Information

kopragui offers information about the following items:

- use of *kopragui*
- latest modifications of *kopragui*
- a web site with informations about *kopra*
- the author

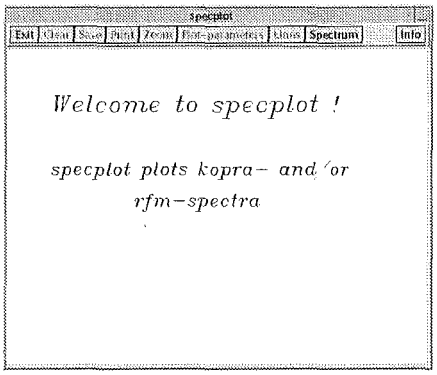
2 *specplot*

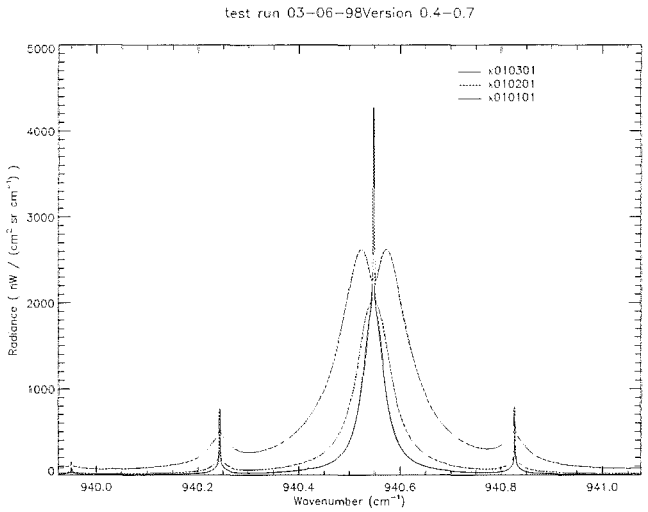
specplot is an interactive plot tool for *kopra* and/or *rfm* spectra:
specplot offers the following opportunities:

- **plot of *kopra* and/or *rfm*spectra**

  Up to 15 spectra can be plotted in one diagramm. The file selection box allows to select several spectra at a time. The maximum data range is set automatically.

  The figure below shows three spectra in a plot:





- **selectable curve attributes: colors, linestyles, markers,...**

  The curve attributes are initialized, but can be set individually by the user. The curves can be distinguished either by color or by linestyle. They can be plotted with up to 15 different *colors.*

  The following *linestyles* are available:

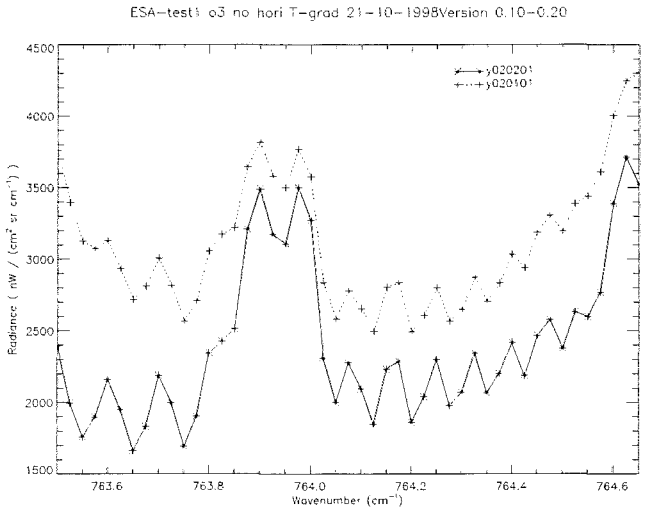
  solid (—), dotted (.....), dashed (_ _ _ _), dash dot (_._._._), dash dot dot dot (_.._.._), long dashes (– –)

  Markers can be set at each gridpoint. The following markers are available:

  plus sign (+), asterisk (*), period (.), diamond, triangle, square, x, circle (o).

  The next figure shows two spectra with different linestyles and markers at each grid point:

- **plot of absolute or relative residual**



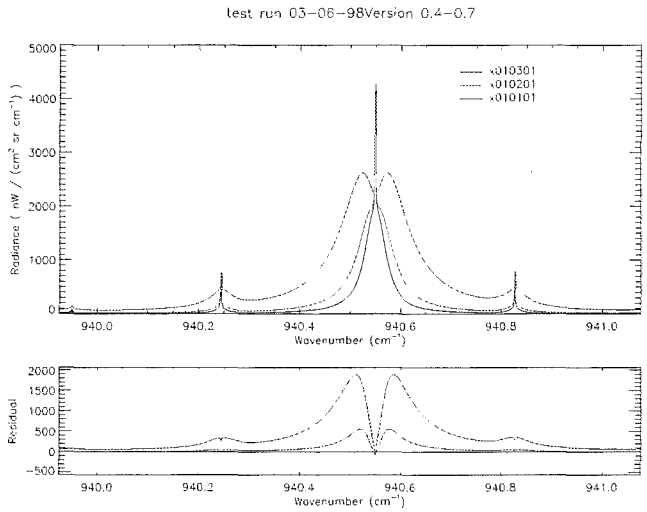
The residual is calculated and added to the plot. The user has the choice to plot the absolute or relative residual. Alternatively, the plot of residuals can be switched off. The first spectrum is reference spectrum. The residuals are calculated according to the rule:

absolute residuals: $(n - ref)$
relative residuals: $(n - ref)/ref$

where ref is the first spectrum and n is the nth spectrum.

If the compared spectra have different gridpoints, specplot uses linear interpolation to calculate the residual. A zero line can be added to the residual.

The figure below shows three spectra with their absolute residuals:



- **zoom**

  Parts of the plot can be zoomed with the mouse according to the following rule:

  - A rectangle is drawn with the *left* mouse button, indicating the area, which should be zoomed.
  - The position of the rectangle can be changed with the *middle* mouse button.
  - The zoom command is send with the *right* mouse button.

  The zoom is enabled by default, but can be disabled by clicking the corresponding option in the menu *zoom*.

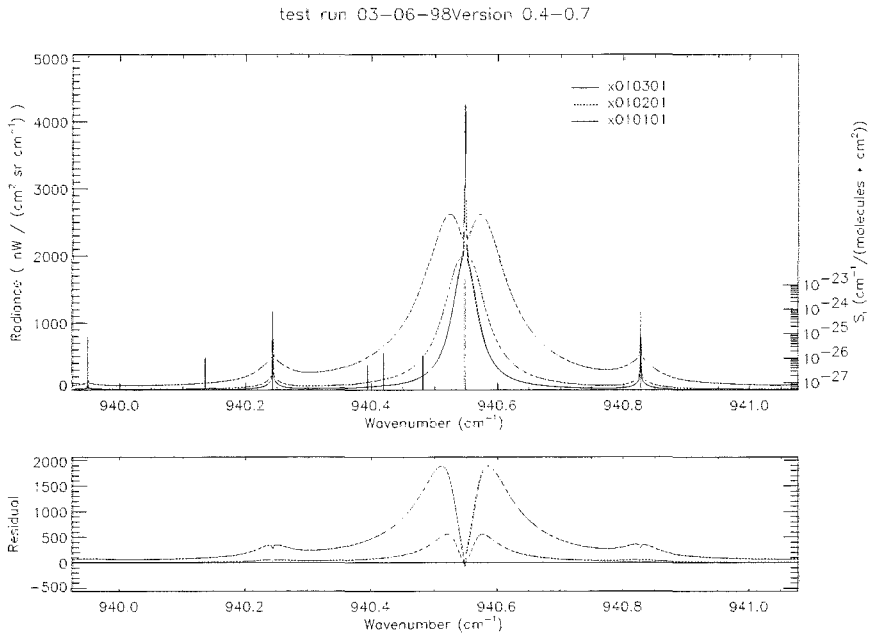
- **changeable legend parameters**

  The legend entries correspond to the file names of the spectra, but can be modified. The positions of the legend entries can be set individually.

- **plot of line intensities**

  Lines of gases (HITRAN line data gases, non-HITRAN line data gases, pseudo line data gases) can be added to the plot. The line intensities are drawn according to a seperate logarithmic y-axis on the lower right side of the diagram. The number of lines can be reduced by setting a threshold intensity, so that only lines with greater intensities will be plotted. The data is read from a database.

The figure below shows three spectra with CO_2 lines in the given wavenumber range:



- **hard copy**

  The plot can be saved in one of the following formats:

  postscript (ps), encapsulated postscript (eps) or encapsulated postscript interchange format (epsi).

- **print**

  The plot can be printed on paper or transparency in black+white or color.