



Efficient implementation of advanced Richardson Extrapolation in an atmospheric chemical scheme

Zahari Zlatev¹ · Ivan Dimov² · István Faragó^{3,4} · Krassimir Georgiev² · Ágnes Havasi⁵ 

Received: 24 August 2021 / Accepted: 19 October 2021 / Published online: 21 November 2021
© The Author(s) 2021

Abstract

The numerical treatment of an atmospheric chemical scheme, which contains 56 species, is discussed in this paper. This scheme is often used in studies of air pollution levels in different domains, as, for example, in Europe, by large-scale environmental models containing additionally two other important physical processes—transport of pollutants in the atmosphere (advection) and diffusion phenomena. We shall concentrate our attention on the efficient numerical treatment of the chemical scheme by using Implicit Runge–Kutta Methods combined with accurate and efficient advanced versions of the Richardson Extrapolation. A Variable Stepsize Variable Formula Method is developed in order to achieve high accuracy of the calculated results within a reasonable computational time. Reliable estimations of the computational errors when the proposed numerical methods are used in the treatment of the chemical scheme will be demonstrated by presenting results from several representative runs and comparing these results with “exact” concentrations obtained by applying a very small stepsize during the computations. Results related to the diurnal variations of some of the chemical species will also be presented. The approach used in this paper does not depend on the particular chemical scheme and can easily be applied when other atmospheric chemical schemes are selected.

Keywords Atmospheric chemical schemes · Ordinary differential equations · Implicit Runge–Kutta methods · Richardson Extrapolation

1 Introduction

Long-range transport of air pollutants in the atmosphere is often studied (as, for example, in [29, 31]) by applying complex mathematical models described by non-linear systems of partial differential equations (PDEs), which, after spatial

✉ Ágnes Havasi
havasia2@hotmail.com

Extended author information available on the last page of the article

discretization, lead to huge systems of ordinary differential equations (ODEs). If some splitting procedure is applied, then the chemical reactions together with the deposition and the emission terms can be handled separately, as a non-linear system of ODEs, at every grid-point of the domain in which the discretized system of PDEs is defined. This is done for an arbitrary grid-point. We choose this approach in order to explain in a simple manner how good numerical procedures can be designed for the chemical part of a large air pollution model. Such models can also be used to investigate the impact of future climatic changes on the pollution levels (see [30, 32]).

Different chemical schemes can be implemented in the air pollution models. These schemes play a very important role in the air pollution studies. The number of chemical species involved in the large-scale air pollution models varies as a rule from 20 to about 200 (see, for example, [1, 2, 5–7, 10, 11, 15, 16, 19, 22–24]).

The use of less than 20 chemical species will require crude parameterization of some of the chemical processes and, therefore, the choice of such chemical schemes is in general not advisable when long-range transport of air pollutants in the atmosphere is to be studied.

On the other side, the application of chemical schemes with more than 200 species leads to huge computational tasks when long-term simulations, as those reported in [30, 32], are to be carried out by running the air pollution models over many consecutive years.

A chemical scheme containing 56 species [31] will be used in this study. Moreover, we shall concentrate our attention on the necessity to increase the computational efficiency during the treatment of the chemical scheme and shall investigate the implementation of some reliable and accurate numerical algorithms in the handling of the selected scheme. This can successfully be done by separating the chemical scheme from the other parts of the air pollution models. This approach facilitates very essentially the search for optimal numerical algorithms when the chemical reactions are to be handled numerically. There is an additional benefit of the treatment of the problem in this way: the chemical scheme together with the efficient algorithms used to handle it numerically can easily be implemented in any large-scale air pollution model when the investigation is successfully finished. Finally, reliable information about some interesting phenomena, such as, for example, diurnal variations of some chemical species, can sometimes be easily obtained by using only the chemical scheme. This possibility will be demonstrated in the end of this paper.

2 Mathematical description of the problem

The chemical reactions together with the deposition and emissions terms of a large-scale air pollution model can be described mathematically by the following initial value problem for non-linear ordinary differential equations (ODEs):

$$y' = f(t, y), \quad t \in [a, b], \quad a < b, \quad y \in D \subset \mathbb{R}^s$$

$$\text{and } y(a) = \eta, \quad \eta \in \mathbb{R}^s \quad \text{being a given constant.} \quad (1)$$

Here f is a given function defined in some domain $D \subset \mathbb{R} \times \mathbb{R}^s$ (it will always be assumed in this paper that f is a one-valued function in the whole domain D). Later further assumptions will be made about the smoothness of the function f , see Sect. 4.2. If high-order versions of the Richardson Extrapolation (RE) are to be used, then one must also assume that f is, as well as y , sufficiently many times continuously differentiable. The components of the solution vector $y(t) \in \mathbb{R}^s$ are concentrations of the chemical species at time $t \in [a, b]$. As mentioned above, $s = 56$ is used in this paper, but the main ideas are very general and similar results can easily be obtained when other chemical schemes are used (as, for example, the schemes with $s = 35$ and $s = 168$ from [31]).

Assume that $t_0 = a$, $t_n = t_{n-1} + h_n$ (where $h_n > 0$ for $n = 1, 2, \dots, N$) and $t_N = b$. Denote $y_0 = y(a) = \eta$. Then some numerical method for solving systems of ODEs has to be used to calculate successively approximations $y_1 \approx y(t_1)$, $y_2 \approx y(t_2)$, \dots , $y_N \approx y(t_N)$ of the exact solution of (1) on some non-equidistant set of grid-points $\{t_1, t_2, \dots, t_N\}$. Normally, it is desirable to select at every step as large as possible stepsize h_n with which the calculated by the selected numerical method approximation y_n will satisfy some accuracy requirement, as for example, $\|y(t_n) - y_n\| / \|y(t_n)\| \approx \text{TOL}$, where $\text{TOL} > 0$ is some prescribed in advance error tolerance. Moreover, we say that the order of accuracy of the numerical solution y_n is p when

$$y(t_n) - y_n = O(h^n).$$

This means that two important computational tasks are to be resolved:

- (a) it is necessary to select appropriate numerical methods for solving non-linear systems of ODEs by which sufficiently accurate approximations y_n can be calculated
and
- (b) it is worthwhile to derive and implement reliable error estimators by which the accuracy of the approximations y_n , $n = 1, 2, \dots, N$ can be evaluated in a reliable way.

The solution of these two important tasks will be presented and discussed in the next two sections.

3 Selection of appropriate numerical methods for solving ODEs

The choice of numerical methods, which can efficiently be used in the treatment of atmospheric chemical schemes, is a very difficult task. There are at least three problems, which are creating great difficulties when the atmospheric chemical schemes are treated numerically.

Table 1 Typical minimum and maximum values of four of the 56 chemical species in an interval of 24 h (from 12 o'clock in a given day to 12 o'clock in the next day)

| Chemical species | Minimum value | Maximum value |
|------------------|---------------|---------------|
| OP | 1.0E−25 | 1.6E+04 |
| OH | 3.3E+04 | 2.2E+07 |
| PAN | 9.4E+05 | 1.3E+10 |
| Ozone | 1.4E+12 | 1.8E+12 |

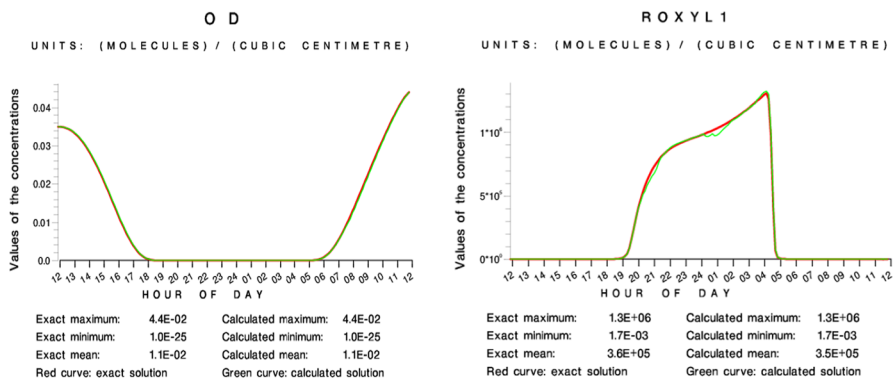


Fig. 1 Diurnal variation of two chemical species. Both the “exact” solution (the solution calculated with a very small stepsize, see the explanations related to the obtaining of the “exact” solution that are given in Sect. 6) and the numerical solution, which is calculated by using the method, which will be discussed in Sect. 5, are given in the two plots

The first problem is caused by the fact that the chemical schemes are very badly scaled. This is demonstrate in Table 1, where the maximum and minimum values of four chemical species are given.

The second problem is caused by the fact that some components of their solution vectors vary very quickly in some parts of the interval $[a, b]$ forming extremely sharp gradients [29, 31]. This is illustrated in Fig. 1 where the diurnal variations of two chemical species are plotted. It is seen that some components of the solution vector $y(t) \in \mathbb{R}^s$ are varying very quickly in some rather short intervals (containing as a rule the time changes from day to night and from night to day).

The third problem is the stiffness of the systems of ODEs, by which chemical schemes are described mathematically. The fact that (1) is a stiff system of ODEs implies that implicit numerical methods have to be selected and used in the solution process. The application of stable Implicit Runge–Kutta Methods (IRKMs), see [3, 4, 12, 14, 17, 18, 21], is a good choice and such methods will be discussed in the remaining part of this section.

We selected three IRKMs and used these methods in the experiments, results of which will be reported in Sect. 6:

- (a) EULERB (the First Order Backward Differentiation Formula),
- (b) DIRK23 (a two-stage third-order Diagonally Implicit Runge–Kutta Method) and
- (c) FIRK35 (a three-stage fifth-order Fully Implicit Runge–Kutta Method).

3.1 Introduction of EULERB

The first of the three selected numerical methods, EULERB, is the well-known and often used First Order Backward Differentiation Formula (called also the Implicit Euler Method or Backward Euler Method). EULERB is a one-stage first-order IRKM that is based on the use of the following formula:

$$y_n = y_{n-1} + h_n f(t_n, y_n), \quad n \in \{1, 2, \dots, N\}. \quad (2)$$

The method is very robust and has good stability properties (L-stable, see [14, 18]), but it is not very accurate.

3.2 Introduction of DIRK23

The second of the three selected numerical methods, DIRK23, is a two-stage third-order Diagonally Implicit Runge–Kutta Method, which is based on the following formulae:

$$y_n = y_{n-1} + 0.5h_n (k_1^n + k_2^n), \quad n \in \{1, 2, \dots, N\}, \quad (3)$$

$$k_1^n = f\left(t_{n-1} + \frac{3 + \sqrt{3}}{6}h_n, \quad y_{n-1} + \frac{3 + \sqrt{3}}{6}h_n k_1^n\right), \quad (4)$$

$$k_2^n = f\left(t_{n-1} + \frac{3 - \sqrt{3}}{6}h_n, \quad y_{n-1} - \frac{\sqrt{3}}{3}h_n k_1^n + \frac{3 + \sqrt{3}}{6}h_n k_2^n\right). \quad (5)$$

DIRK23 has good stability properties (being an A-stable method, [18]) and it is more accurate than EULERB, but it is more time-consuming too. At every time-step it is necessary to handle successively the two non-linear systems of algebraic Eqs. (4) and (5), each of which contains s equations, while only one such system is to be solved when EULERB is used.

3.3 Introduction of FIRK35

The third of the three selected numerical methods, FIRK35, is a three-stage fifth-order Fully Implicit Runge–Kutta Method based on the following formulae:

$$y_n = y_{n-1} + \frac{16 - \sqrt{6}}{36} h_n k_1^n + \frac{16 + \sqrt{6}}{36} h_n k_2^n + \frac{1}{9} h_n k_3^n, \quad n \in \{1, 2, \dots, N\}, \quad (6)$$

$$k_1^n = f \left(t_{n-1} + \frac{4 - \sqrt{6}}{10} h_n, \quad y_{n-1} + \frac{88 - 7\sqrt{6}}{360} h_n k_1^n + \frac{296 - 169\sqrt{6}}{1800} h_n k_2^n + \frac{-2 + 3\sqrt{6}}{225} h_n k_3^n \right), \quad (7)$$

$$k_2^n = f \left(t_{n-1} + \frac{4 + \sqrt{6}}{10} h_n, \quad y_{n-1} + \frac{296 + 169\sqrt{6}}{1800} h_n k_1^n + \frac{88 + 7\sqrt{6}}{360} h_n k_2^n + \frac{-2 - 3\sqrt{6}}{225} h_n k_3^n \right), \quad (8)$$

$$k_3^n = f \left(t_{n-1} + h_n, \quad y_{n-1} + \frac{16 - \sqrt{6}}{36} h_n k_1^n + \frac{16 + \sqrt{6}}{36} h_n k_2^n + \frac{1}{9} h_n k_3^n \right). \quad (9)$$

FIRK35 is much more accurate than EULERB and DIRK23. It has excellent stability properties, being L-stable, [14, 18]. However, it is much more time-consuming than the other two numerical methods, because one has to handle the systems of 3s non-linear algebraic equations formed by (7), (8), and (9) at every step n, while systems of s non-linear algebraic equations are to be solved at each step when EULERB and DIRK23 are used.

4 Applying advanced versions of the Richardson Extrapolation

Any of the introduced in the previous section IRKMs can be used in combination with nine advanced versions of the Richardson Extrapolation, which were introduced and studied in [33, 34]. These versions can be applied in order to increase the order of accuracy and/or to obtain reliable error estimations.

The theory of the Advanced Richardson Extrapolation can be based on the four important theorems, which are listed below without proofs (the proofs can be found in [33, 34]).

It should be mentioned here that all results presented in [33, 34] and formulated below are valid for one-step methods for solving systems of ODEs. However, these theorems are also valid for Runge–Kutta methods, because the class of the Runge–Kutta methods is a sub-class of the class of one-step methods.

The first theorem is telling us how to obtain each of the nine advanced versions of the Richardson Extrapolation.

The second theorem is providing results related to the accuracy of the advanced versions of the Richardson Extrapolation.

The third theorem is establishing rules for the calculation of reliable error estimations by using the advanced versions of the Richardson Extrapolation.

The fourth theorem is telling us how to obtain information about the stability properties of the different versions of the Richardson Extrapolation.

4.1 Derivation of nine advanced versions of the Richardson Extrapolation

Theorem 1 Let $y_0^{[q]} = y(a) = \eta$ for $\forall q \in \{0, 1, \dots, 8\}$. Assume that some integer $q \in \{0, 1, \dots, 8\}$ has been selected and that $y_{n-1}^{[q]} \in \mathbb{R}^s$ is available. Calculate first $q + 2$ auxiliary vectors $z_n^{[0]}, z_n^{[1]}, \dots, z_n^{[q+1]}$ in \mathbb{R}^s by using $y_{n-1}^{[q]}$ together with any one-step method and by performing respectively 1, 2, ..., 2^{q+1} steps with stepsizes $h, h/2, \dots, h/2^{q+1}$. Then vector $y_n^{[q]}$ (the q -Times Repeated Richardson Extrapolation, the q TRRE) can be obtained by using the following two formulae:

$$y_n^{[q]} = \frac{P_n^{[q]}(z_n^{[q+1]}, z_n^{[q]}, \dots, z_n^{[0]})}{S^{[q]}}, \tag{10}$$

$$S^{[q]} = \prod_{j=1}^{q+1} (2^{p+j-1} - 1). \tag{11}$$

The first term in $P_n^{[q]}$ is obtained by multiplying the first term in $S^{[q]}$ with $z_n^{[q+1]}$, the second term in $P_n^{[q]}$ is obtained by multiplying the second term in $S^{[q]}$ with $z_n^{[q]}$ and one must continue in this way. The last term in $P_n^{[q]}$ is obtained by multiplying the last term in $S^{[q]}$ with $z_n^{[0]}$.

Note that any of the three IRKMs discussed in the previous sections can be used in the calculation of the approximations $z_n^{[0]}, z_n^{[1]}, \dots, z_n^{[q+1]}$, but one can also apply any other IRKM or even an arbitrary one-step method as stated in Theorem 1.

Three examples for advanced Richardson Extrapolation formulae are given below (all other formulae can be found in [33, 34]).

Classical Richardson Extrapolation, CRE:

$$y_n^{[0]} = \frac{P_n^{[0]}}{S^{[0]}} = \frac{2^p z_n^{[1]} - z_n^{[0]}}{2^p - 1}. \tag{12}$$

Repeated Richardson Extrapolation, RRE:

$$y_n^{[1]} = \frac{P_n^{[1]}}{S^{[1]}} = \frac{2^{2p+1} z_n^{[2]} - 3(2^p) z_n^{[1]} + z_n^{[0]}}{2^{2p+1} - 3(2^p) + 1}. \tag{13}$$

Two-Times Repeated Richardson Extrapolation, 2TRRE:

$$y_n^{[2]} = \frac{P_n^{[2]}}{S^{[2]}} = \frac{2^{3p+3} z_n^{[3]} - 7(2^{2p+1}) z_n^{[2]} + 7(2^p) z_n^{[1]} - z_n^{[0]}}{2^{3p+3} - 7(2^{2p+1}) + 7(2^p) - 1}. \tag{14}$$

4.2 Accuracy of the advanced versions of the Richardson Extrapolation

Theorem 2 Consider an arbitrary one-step method for solving non-linear systems of ODEs $dy/dt = f(t, y)$. Assume that

- (a) the order of accuracy of the selected one-step method is $p \geq 1$
and
(b) it is used together with a q TRRE where $0 \leq q \leq 8$.

Then the order of accuracy of the approximation $y_n^{[q]}$ (where vector $y_n^{[q]}$ is calculated as shown in Theorem 1) is at least $p + q + 1$ when the function f in the right-hand-side of the system of ODEs $dy/dt = f(t, y)$ is at least $p + q + 1$ times continuously differentiable.

The nine advanced versions of the Richardson Extrapolation, the abbreviations used in this paper (as well as in [33, 34]) and their orders of accuracy, according to Theorem 2, are listed in Table 2.

4.3 Error estimators for the advanced versions of the Richardson Extrapolation

Theorem 3 Assume that $z_n^{[0]}$, $z_n^{[1]}$, $y_n^{[q-1]}$ and $y_n^{[q]}$ are calculated as shown in Theorem 1 as well as that the order of accuracy of the underlying one-step method is p . Then error estimations at step n , where $n = 1, 2, \dots, N$, can be obtained by using the following two formulae:

Table 2 The nine versions of the Richardson Extrapolation, the abbreviations that will be used in the remaining part of this paper and the orders of accuracy are given in this table

| q | Names of the nine versions | Abbreviation | Order of accuracy |
|-----|---|--------------|-------------------|
| 0 | Classical Richardson Extrapolation | CRE | $p + 1$ |
| 1 | Repeated Richardson Extrapolation | RRE | $p + 2$ |
| 2 | Two Times Repeated Richardson Extrapolation | 2TRRE | $p + 3$ |
| 3 | Three Times Repeated Richardson Extrapolation | 3TRRE | $p + 4$ |
| 4 | Four Times Repeated Richardson Extrapolation | 4TRRE | $p + 5$ |
| 5 | Five Times Repeated Richardson Extrapolation | 5TRRE | $p + 6$ |
| 6 | Six Times Repeated Richardson Extrapolation | 6TRRE | $p + 7$ |
| 7 | Seven Times Repeated Richardson Extrapolation | 7TRRE | $p + 8$ |
| 8 | Eight Times Repeated Richardson Extrapolation | 8TRRE | $p + 9$ |

The integer p used in the last column of the table is the order of accuracy of the underlying numerical method for solving numerically systems of ODEs. This means that $p = 1$, $p = 3$ and $p = 5$ when EULERB, DIRK23 and FIRK35 are respectively used. The Classical Richardson Extrapolation (the CRE) was introduced in [20]. Some properties of the Repeated Richardson Extrapolation (the RRE) were studied in [8]

$$\text{EST}_n^{[p0]} = \frac{z_n^{[1]} - z_n^{[0]}}{2^p - 1} \quad (15)$$

and

$$\text{EST}_n^{[pq]} = y_n^{[q]} - y_n^{[q-1]}, \quad q \in \{1, 2, \dots, 8\}. \quad (16)$$

Remark 1 Formula (16) is explaining the general rule that is used in the calculation of $\text{EST}_n^{[pq]}$, $q \in \{1, 2, \dots, 8\}$, but in the practical computations there is no need to calculate vector $y_n^{[q-1]}$. The difference in the right-hand-side of (16) can easily be expressed by applying, as in (15), the vectors $z_n^{[k]}$, $k = 0, 1, \dots, q + 1$, which must nevertheless be calculated and used in the computation of $y_n^{[q]}$, see Theorem 1.

Two examples, the error estimations for $q = 1$ and $q = 2$, are given below. All formulae related to the error estimations, when different versions of the Richardson Extrapolation are used with $q > 2$, are given in [33].

Error estimation for the Repeated Richardson Extrapolation, RRE:

$$\text{EST}_n^{[p1]} = \frac{2^p z_n^{[2]} - (2^p + 1)z_n^{[1]} + z_n^{[0]}}{2^{2p+1} - 3(2^p) + 1}. \quad (17)$$

Error estimation for the Two-Times Repeated Richardson Extrapolation, 2TRRE:

$$\text{EST}_n^{[p2]} = \frac{2^{2p+1}z_n^{[3]} - (2^{2p+1} + 3 \cdot 2^p)z_n^{[2]} + (3 \cdot 2^p + 1)z_n^{[1]} - z_n^{[0]}}{2^{3p+3} - 7(2^{2p+1}) + 7(2^p) - 1}. \quad (18)$$

4.4 Stability properties of the advanced versions of the Richardson Extrapolation

Theorem 4 If $0 \leq q \leq 8$ and $\nu \in \mathbb{C}^-$, then the value of the stability function of the q TRRE can be calculated by using the following formula:

$$R^{[q]}(\nu) = \frac{\bar{P}^{[q]}(R(\nu))}{S^{[q]}}, \quad (19)$$

where $S^{[q]}$ is defined by (11) and $R(\nu)$ is the value of the stability function of the underlying numerical method for solving systems of ODEs.

The formulae for all functions $\bar{P}^{[q]}(R(\nu))$, $q \in \{0, 1, \dots, 8\}$, are listed in [34]. The important issue is that if we know the value $R(\nu)$ of the stability function of the underlying one-step method for some $\nu \in \mathbb{C}^-$, then we are able to calculate, by using (19), the values of the stability functions $R^{[q]}(\nu)$ of all advanced versions of the Richardson Extrapolation, i.e. the values of the stability functions for all $q \in \{0, 1, \dots, 8\}$.

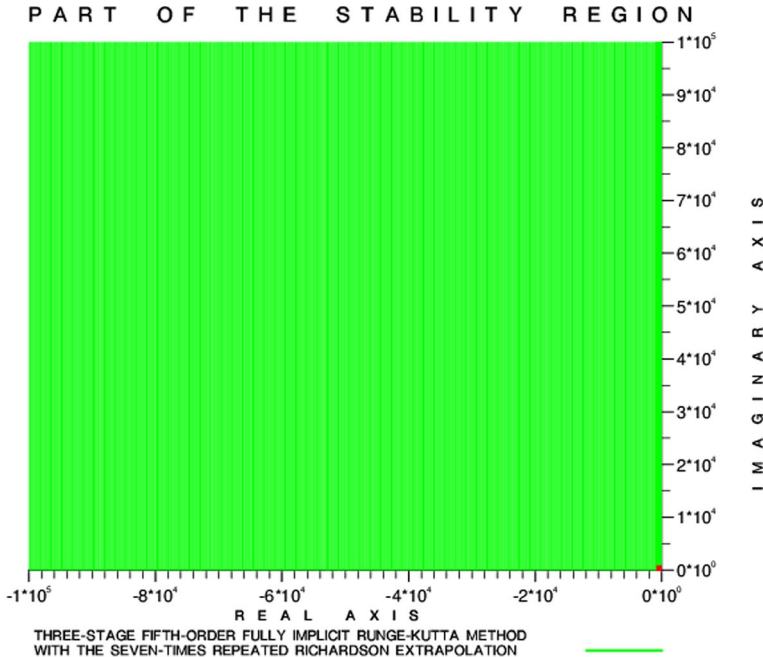


Fig. 2 Part of the absolute stability region of the three-stage fifth-order Fully Implicit Runge–Kutta Method (FIRK35) combined with the Seven Times Repeated Richardson Extrapolation (7TRRE). There are only some points in the small square near the origin of the coordinate system where $|R^{[7]}(\nu)| \geq 1$. This means that instability may occur if the Jacobian matrix of function f of the system $dy/dt = f(t, y)$ has eigenvalues close to the origin of the coordinate system (Color figure online)

There are many different stability definitions (see [3, 4, 13, 14, 17, 21]). The result presented in (19) is valid for any stability function used in the different definitions. However, the classical definition for absolute stability, which was introduced by G. Dahlquist in 1963, [9], is used in all calculations, which are discussed in the remaining part of this section. Some results from these calculations are presented in Fig. 2. It is explained below how these results were obtained.

If we calculate the values of $R^{[q]}(\nu)$ for some $q \in \{0, 1, \dots, 8\}$ on a sufficiently dense regular set of points $\nu \in \mathbb{C}^-$, then we shall be able to obtain reliable information about the stability of the qTRRE in the continuous domain containing this set of points.

The idea sketched in the previous paragraph was implemented in the following way. Consider the square SQ in the negative part of the complex plane with vertices the coordinates of which are given by $(0.0, 0.0)$, $(0.0, 10^5i)$, $(-10^5, 10^5i)$, $(-10^5, 0.0)$. Discretize the square SQ by using an equidistant set PSQ of grid-points on the lines parallel to the coordinate axes such that the distance between two neighbour grid-points (either in the vertical direction or in the horizontal direction) is $d = 0.001$. It is clear that the total number of grid-points in PSQ is $O(10^{16})$. The values of the function $R^{[q]}(\nu)$ were calculated for all $\nu \in \text{PSQ}$ by using each of the three

underlying Implicit Runge–Kutta Methods (IRKMs) for solving ODEs from Sect. 3 with all $q \in \{0, 1, \dots, 8\}$, i.e. with all nine advanced versions of the Richardson Extrapolation discussed in Sect. 4. The results from the 27 runs (the runs with the three underlying methods for solving ODEs, each of these methods being combined with all nine versions of the Richardson Extrapolation) were very similar. It is clear that one should expect the computations to remain stable when the condition $\left| R^{[q]}(\nu) \right| < 1.0$ is satisfied. This condition was fulfilled in all runs for all points in the set PSQ, excepting only a few points near the origin of the coordinate system.

One example, which is an illustration of the conclusion made above, is given in Fig. 2. The three-stage fifth-order Fully Implicit Runge–Kutta Method, the FIRK35, is used in this example together with the Seven-Times Repeated Richardson Extrapolation, the 7TRRE, to obtain the results, which are shown in the plot given in Fig. 2. It shows the domain where one should expect the computational process to remain stable because $\left| R^{[q]}(\nu) \right| < 1.0$ for all points in this domain. The inequality $\left| R^{[q]}(\nu) \right| \geq 1.0$ holds only for a few points located near the origin of the coordinate system. This means that one must be careful if the Jacobian matrix of function f has small eigenvalues with negative real parts that are close to the imaginary axis.

As mentioned above very similar results, as those shown in Fig. 2, were obtained in the other 26 runs.

It should also be mentioned that the inequality $\left| R^{[0]}(\nu) \right| < 1.0$ is satisfied for $\forall \nu \in \text{PSQ}$ when the Classical Richardson Extrapolation, CRE, is combined with EULERB and DIRK23. This indicates that the combinations of CRE with EULERB and DIRK23 are stable in the whole square SQ.

It must be emphasized that the investigation of the stability of the advanced versions of the Richardson Extrapolation is very important and must always be carried out. The following fact explains why this is so. The combination of the Classical Richardson Extrapolation, the CRE, with the well-known Trapezoidal Rule (which is A-stable), results in an unstable numerical method, see [9].

5 Development of a variable stepsize variable formula method

It is clear (see the last column in Table 2) that the accuracy of the computational process can be increased very considerably when the value of parameter q is increased. For example, the order of accuracy is becoming 14 when the 8TRRE is used together with FIRK35. However, it is necessary to pay something for the high accuracy: the computational work is also increased very considerably. This explains why it is worthwhile to carry out the computations with an attempt to keep both the error of the calculated solution and the number of the computational operations sufficiently small. This aim can be achieved by selecting, at each step $n \in \{1, 2, \dots, N\}$ of the computational process, as large as possible stepsize h_n and a version q TRRE with as small as possible value of parameter $q \in \{0, 1, \dots, 8\}$ under the essential requirement that the computational error remains sufficiently small (according to a prescribed error tolerance $\text{TOL} > 0$). This means that it is necessary to develop a

Variable Stepsize Variable Formula Method (a VSVFM) in an attempt to optimize the computational process.

VSVFMs have been studied in many papers. Some ideas from [25–28] are used in the derivation of the algorithm, which is described below.

The major rules used in our Variable Stepsize Variable Formula Method, our VSVFM, which is based on the application of any of the three underlying IRKMs presented in Sect. 3 together with the nine advanced versions of the Richardson Extrapolation from Sect. 4, are sketched below.

A key parameter RATIO is calculated at every time-step by using the formula:

$$\text{RATIO} = \delta \left(\frac{\text{TOL}}{\text{EST}_n^{[pq]}} \right)^{\frac{1}{p+q+1}}, \quad n = 1, 2, \dots, N, \quad q = 0, 1, \dots, 8, \quad p = 1, 3 \text{ or } 5. \quad (20)$$

The value of the integer q is telling us what is the particular version of the Richardson Extrapolation which is used at step n . The integer p is the order of the underlying numerical method for solving systems of ODEs. The coefficient δ ($\delta < 1.0$), from the right-hand-side of (20), is a precaution factor used always in the preparation of VSVFMs (see, for example, [18]); $\delta = 0.9$ is selected in all experiments results of which are presented in this paper.

The five major actions, which are to be performed at the end of any step n ($n = 1, 2, \dots, N$), can be formulated as shown in Table 3. It is clearly seen that all these actions depend essentially on the value of parameter RATIO.

It should be emphasized that only the most important actions are described in Table 3. Several additional rules must also be used. For example, no new increase of the stepsize is in general allowed several steps after increasing it. This rule is useful, because the theory, on which the numerical methods for solving systems of ODEs are based, is strictly speaking valid only in the case where a constant stepsize is used. Therefore, the stepsize should not be varied too often and/or with a too large amount. There is a parameter WAIT in the program, which is used to keep the same stepsize at least during two consecutive steps. However, if the check of parameter RATIO indicates that either Case 4 or Case 5 (see Table 3) has taken place, then the step is always rejected and the computations are repeated by applying a reduced stepsize.

6 Numerical results

The chemical scheme with 56 species was run over a time-interval of 24 h, starting at 12 o'clock at a given day and finishing at 12 o'clock on the next day. This is a very relevant and important choice, because this interval contains the periods of changes from day-time to night-time and from night-time to day-time when some chemical species vary very quickly (see the plots given in Fig. 1) and can cause great problems for the numerical methods. The time is measured in seconds. This means that the time-interval, which is actually used in all computations, was $[a, b] = [43200, 129600]$. This interval was divided into 168 sub-intervals (the

Table 3 Major rules for accepting or rejecting the approximation obtained at the current step n

| Case | RATIO | Action |
|--------|----------------------------------|--|
| Case 1 | $0.9 \leq \text{RATIO} \leq 1.5$ | Keep the same stepsize by setting $h_{n+1} = h_n$ Increase the order of accuracy when $\text{RATIO} < 1.0$ by selecting more accurate version of the Richardson Extrapolation (replacing q by $q + 1$ if $q < 8$) Decrease the order of accuracy when $\text{RATIO} > 1.25$ by selecting less accurate version of the Richardson Extrapolation (replacing q by $q - 1$ if $q > 0$) Continue with the calculation of y_{n+1} Increase the stepsize by setting $h_{n+1} = 1.25 * h_n$ / |
| Case 2 | $1.5 < \text{RATIO} \leq 4.0$ | Increase the order of accuracy when $\text{RATIO} > 2.0$ by selecting a more accurate version of the Richardson Extrapolation (replacing q by $q + 1$ if $q < 8$) Continue with the calculation of y_{n+1} |
| Case 3 | $4.0 \leq \text{RATIO}$ | Increase the stepsize by setting $h_{n+1} = 1.5 * h_n$ Increase the order of accuracy when $\text{RATIO} > 6.0$ by selecting more accurate version of the Richardson Extrapolation (replacing q by $q + 1$ if $q < 8$) Continue with the calculation of y_{n+1} |
| Case 4 | $0.1 \leq \text{RATIO} < 0.9$ | Decrease the stepsize by setting $h_n = 0.5 * h_n$ Increase the order of accuracy when $\text{RATIO} < 0.25$ by selecting more accurate version of the Richardson Extrapolation (replacing q by $q + 1$ if $q < 8$) Reject the step and repeat it by calculating a new y_n |
| Case 5 | $\text{RATIO} < 0.1$ | Decrease the stepsize by setting $h_n = 0.25 * h_n$ Increase the order of accuracy when $\text{RATIO} < 0.05$ by selecting more accurate version of the Richardson Extrapolation (replacing q by $q + 1$ if $q < 8$) Reject the step and repeat it by calculating a new y_n |

Table 4 Results obtained by running DIRK23 combined with different versions of the Richardson Extrapolation used as a VSVFM

| TOL | Successful steps | Rejected steps | Exact maximal error | Estimated maximal error |
|------------|------------------|----------------|---------------------|-------------------------|
| 10^{-7} | 342 | 0 | $9.203 * 10^{-8}$ | $6.384 * 10^{-8}$ |
| 10^{-8} | 428 | 0 | $1.906 * 10^{-8}$ | $4.655 * 10^{-9}$ |
| 10^{-9} | 362 | 0 | $3.128 * 10^{-9}$ | $4.791 * 10^{-10}$ |
| 10^{-10} | 925 | 0 | $3.932 * 10^{-10}$ | $3.703 * 10^{-11}$ |
| 10^{-11} | 3231 | 10 | $9.882 * 10^{-12}$ | $3.473 * 10^{-12}$ |

Five different values of the error tolerance TOL are used and the code is always trying to keep the order of the local error estimation (which is calculated during the different runs) approximately equal to TOL

Table 5 Results obtained by running FIRK35 combined with different versions of the Richardson Extrapolation used as a VSVFM

| TOL | Successful steps | Rejected steps | Exact maximal error | Estimated maximal error |
|------------|------------------|----------------|---------------------|-------------------------|
| 10^{-10} | 349 | 0 | 1.528E–11 | 4.770E–11 |
| 10^{-11} | 359 | 0 | 5.188E–12 | 7.807E–12 |
| 10^{-12} | 430 | 3 | 7.031E–13 | 8.479E–13 |
| 10^{-13} | 532 | 6 | 7.531E–14 | 8.337E–14 |
| 10^{-14} | 571 | 8 | 1.014E–14 | 9.979E–15 |
| 10^{-15} | 786 | 9 | 4.947E–15 | 9.475E–16 |
| 10^{-16} | 1074 | 13 | 3.184E–16 | 9.933E–17 |
| 10^{-17} | 619 | 9 | 2.593E–17 | 4.544E–19 |
| 10^{-18} | 1245 | 9 | 1.602E–17 | 6.809E–19 |
| 10^{-19} | 1381 | 20 | 1.059E–18 | 9.332E–20 |
| 10^{-20} | 1023 | 17 | 5.133E–19 | 9.507E–21 |

Eleven different values of the error tolerance TOL are used and the code is trying to keep the order of the local error estimation (which is calculated during the different runs) approximately equal to TOL

length of each sub-interval being approximately 514.285 seconds) and the “exact” solution was calculated at the end of each sub-interval by applying a constant step-size $h = 10^{-5}$ and by running FIRK35 combined with the 8TRRE.

6.1 Accuracy results

All three numerical methods from Sect. 3 were run with different values of the error tolerance TOL, but the code was always forced to reach the end-point of each of the 168 sub-intervals and the achieved accuracy at the end of each of the 168 sub-intervals of the interval $[a, b]$ was checked by using the formula $\|y^{\text{exact}} - y^{\text{calculated}}\|_2 / \max(\|y^{\text{exact}}\|_2, 10^{-6})$.

Table 6 Results obtained by running EULERB combined with different versions of the Richardson Extrapolation used as a VSVFM

| | Successful steps | Rejected steps | Exact maximal error | Estimated maximal error |
|-----------|------------------|----------------|---------------------|-------------------------|
| 10^{-2} | 345 | 0 | $1.054 * 10^{-1}$ | $8.428 * 10^{-3}$ |
| 10^{-3} | 348 | 0 | $4.092 * 10^{-2}$ | $7.153 * 10^{-4}$ |
| 10^{-4} | 411 | 4 | $9.614 * 10^{-3}$ | $8.191 * 10^{-5}$ |
| 10^{-5} | 434 | 10 | $2.048 * 10^{-3}$ | $6.522 * 10^{-6}$ |
| 10^{-6} | 456 | 2 | $7.588 * 10^{-5}$ | $9.217 * 10^{-7}$ |

Five different values of the error tolerance TOL are used and the code is trying to keep the order of the local error estimation (which is calculated during the different runs) approximately equal to TOL

Table 7 Numbers of calls of different versions of the Richardson Extrapolation when two values of the error tolerance TOL are used in runs where the underlying method is FIRK35

| TOL | CRE | RRE | 2TRRE | 3TRRE | 4TRRE | 5TRRE | 6TRRE | 7TRRE | 8TRRE |
|------------|-----|-----|-------|-------|-------|-------|-------|-------|-------|
| 10^{-11} | 213 | 93 | 21 | 15 | 8 | 3 | 1 | 3 | 2 |
| 10^{-20} | 3 | 156 | 96 | 266 | 221 | 136 | 95 | 49 | 18 |

We started with DIRK23 and used five values of parameter TOL. The results, which are given in Table 4 show clearly that the VSVFM algorithm, the major properties of which were described in Table 3, is working very well in this situation.

Much more accurate results can be achieved, as shown in Table 5, when FIRK35 is used with eleven values of the error tolerance TOL.

Finally, if one is not interested in achieving high accuracy, then EULERB can be used with large values of the error tolerance. This is demonstrated in Table 6, where five values of TOL are used.

The results shown in Tables 4, 5 and 6 indicate that the number of steps is in general not changed too much when the value of the error tolerance TOL is varied. The figures presented in Table 7 explain why this is so: decreasing the value of the error tolerance, i.e. the increase of the accuracy requirements, leads in most of the cases not to a decrease of the stepsize, but to the selection of more accurate versions of the Richardson Extrapolation. This is important in the efforts to achieve some computational balance, because the use of Variable Step Size Variable Formula Methods, VSVFMs, require some extra checks and calculations at each step, which are needed in the efforts to find the answers to the following questions:

- Is the step successful or should it be rejected?
- Should the stepsize be increased and, if this is the case, by how much should it be increased?
- Should the stepsize be decreased and, if this is the case, by how much should it be decreased?

It is important to avoid, if possible, the use of expensive Richardson Extrapolation versions. The results from many runs (not only the runs reported in this section) indicate that the most expensive versions of the Richardson Extrapolation are used only a few times even in the cases where the error tolerance is extremely small. This is illustrated in Table 7. If the error tolerance is $TOL = 10^{-11}$, then the most time consuming version of the Richardson Extrapolation, the 8TRRE, is called only two times (see the first line of Table 7). Even if $TOL = 10^{-20}$, 8TRRE was called only 18 times. This is very small number compared with the total number of steps (compare the figures given in the second line of Table 7).

It should be mentioned here that all runs, results from which were presented in this sub-section, were performed by using extended computer precision (i.e. working with about 32 significant digits of the real numbers).

6.2 Diurnal variations of some chemical species

The diurnal variations of two chemical species were shown in Fig. 1 in order to illustrate the fact that sharp gradients may appear in some short intervals (during changes from night-time to day time and from day time to night-time). The diurnal variations do not necessarily lead to the creation of sharp gradients. This fact is demonstrated in some of the plots shown in Fig. 3, where the diurnal variations of six of the 56 chemical species are presented. As in Fig. 1, two curves are given in each plot: one of these curves is presenting the numerical solution calculated by the code (by using error tolerance $TOL = 0.001$ in this particular case), the other one the “exact solution” calculated by using a very small stepsize. It is seen that the calculated solution is very close to the “exact solution” on the whole interval of 24 h (86,400 s).

7 Concluding remarks

Remark 2 About the stability of the computational process. The three underlying numerical methods, the three IRKMs, have excellent stability properties (see Sect. 3). However, there is a danger that the combinations of any of these methods with some of the versions of the Richardson Extrapolation may result in unstable computational process (see Sect. 4 and the comments related to Fig. 2). Our experiments show that the combination of any of the underlying numerical methods with any of the nine versions of the Richardson Extrapolation is producing stable results. The instability problems are avoided because the VSVMF is changing automatically the stepsize and/or the value of q every time when numerical difficulties appear.

Remark 3 About the use of other underlying numerical methods for solving ODEs. The results presented in Sect. 6 were obtained by applying the same rules for changing the stepsize and the version of the Richardson Extrapolation when the three different underlying methods for solving ODEs, which were introduced in Sect. 3, are

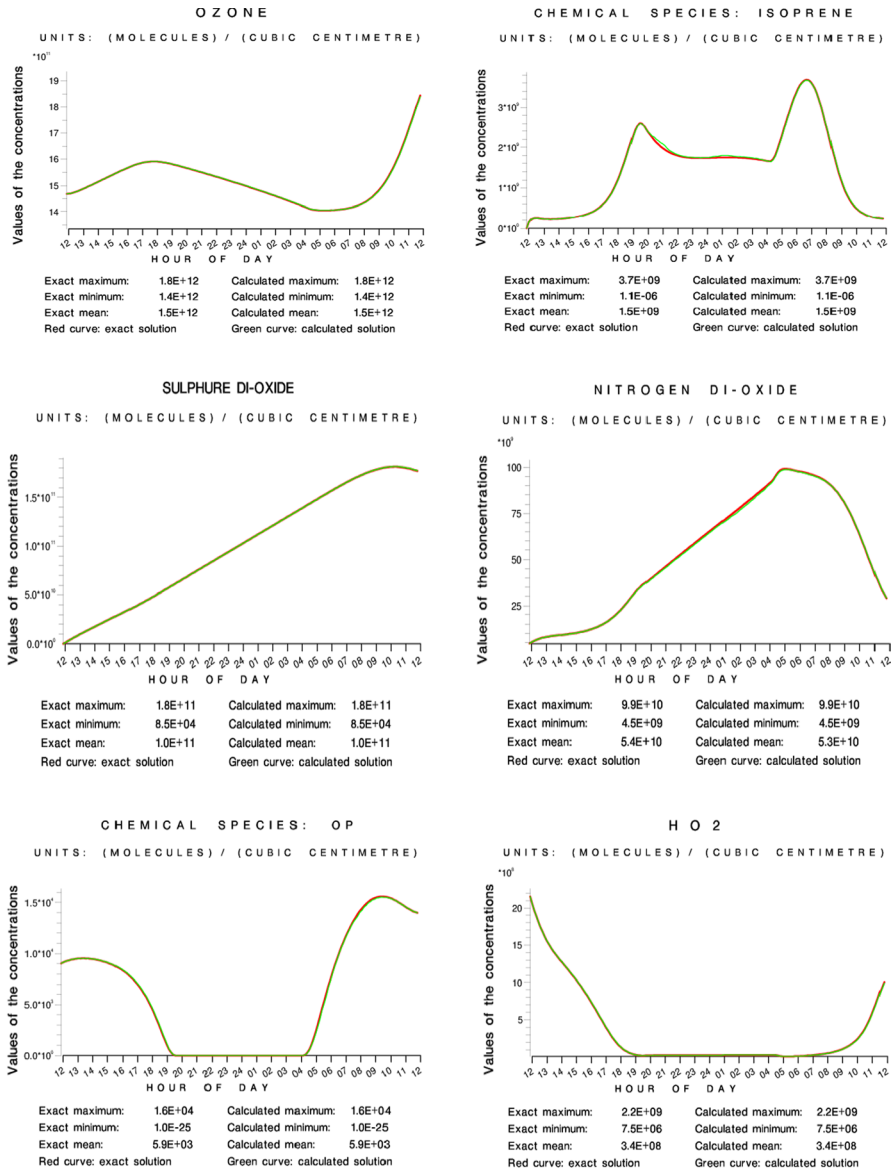


Fig. 3 Diurnal variation of six chemical species. Both the “exact” solution (the solution calculated with a very small stepsize) and the numerical solution (calculated by using the VSVFM, which was discussed in Sect. 5) are given in the six plots

used. This fact indicates that the strategy for varying the stepsize and the Richardson Extrapolation versions does not depend too much on the underlying methods and other underlying methods can also be selected and successfully used. However, it must be emphasized here that it is important to investigate the stability properties of the

combinations of the selected underlying method with all nine versions of the Richardson Extrapolation (the same rules as those explained after Theorem 4 may be used).

Remark 4 About the reduction of the number of advanced versions of the Richardson Extrapolation used in the code. The numerical results indicate that the computationally expensive versions of the Richardson Extrapolation are not used often when the algorithm sketched in Table 3 is used. If one wishes to be sure that these versions will never be used during the runs, then the algorithm described in Table 3 could be slightly modified in order to prevent the use of expensive versions. If the required accuracy is not very high and/or if the solved problem is not very difficult, then it might be worthwhile to introduce a parameter $\text{MAX}_q \leq 8$ so that if, for example, $\text{MAX}_q = 2$, then only the first three versions of the Richardson Extrapolation (with $q = 0$, $q = 1$ and $q = 2$) will be used in the run. Some other values of parameter MAX_q , with $q \in \{0, 1, \dots, 8\}$, can also be selected and used.

Remark 5 About the used chemical scheme. We were mainly interested in the efficient implementation of advanced versions of the Richardson Extrapolation together with three underlying Implicit Runge–Kutta Methods. Therefore, the properties of the used chemical scheme were shortly described. The important issue is that the described algorithms are not depending very much on the particular chemical scheme that is used and can also be applied if other chemical schemes are selected. Some more details about the treatment of the chemical schemes in the air pollution models can be found in [29, 31]. Information about other atmospheric chemical schemes, which can be used in large-scale air pollution models can be found, for example, in [1, 2, 5–7, 10, 11, 15, 16, 19, 22–24].

Acknowledgements This work was completed in the ELTE Institutional Excellence Program (TKP2020-IKA-05) financed by the Hungarian Ministry of Human Resources. Application Domain Specific Highly Reliable IT Solutions" project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme. This work of K. Georgiev was accomplished with the support by the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014–2020) and co-financed by the European Union through the European structural and Investment funds. The work of I. Dimov is partially funded by the research project KP-06-N52/5 "Effective methods for modeling, optimization and decision making" funded by NSF of Bulgaria.

Funding Open access funding provided by Eötvös Loránd University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. I. Ackerman, H. Hass, B. Schell, F.S. Binkowski, Regional modelling of particulate matter with MADE. *Environ. Manag. Health* **10**(4), 201–208 (1999)
2. P. Borrell, P.M. Borrell, *Transport and Transformation of Pollutants in the Troposphere* (Springer, Berlin, 2000)
3. K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations* (Clarendon Press, Oxford, 1995)
4. J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods* (Wiley, New York, 1987). <https://doi.org/10.1002/9781119121534>
5. G.R. Carmichael, L.P. Peters, An Eulerian transport-transformation-removal model for sulphur-dioxide and sulphate. *Atmos. Environ.* **18**, 937–952 (1984). [https://doi.org/10.1016/0004-6981\(84\)90070-2](https://doi.org/10.1016/0004-6981(84)90070-2)
6. G.R. Carmichael, L.P. Peters, A second generation model for regional-scale transport-chemistry-deposition. *Atmos. Environ.* **20**, 173–188 (1986). [https://doi.org/10.1016/0004-6981\(86\)90218-0](https://doi.org/10.1016/0004-6981(86)90218-0)
7. J.S. Chang, R.A. Brost, I.S.A. Isaksen, S. Madronich, P. Middleton, W.R. Stockwell, C.J. Walcek, A three dimensional Eulerian acid deposition model: physical concepts and formulation. *J. Geophys. Res.* **92**, 14691–14700 (1987). <https://doi.org/10.1029/JD092iD12p14681>
8. E. Christiansen, H.G. Petersen, Estimation of convergence orders in repeated Richardson Extrapolation. *BIT Numer. Math.* **29**, 48–59 (1989). <https://doi.org/10.1007/BF01932705>
9. G. Dahlquist, A special stability problem for linear multistep methods. *BIT Numer. Math.* **3**, 27–43 (1963). <https://doi.org/10.1007/BF01963532>
10. A. Ebel, M. Memmesheimer, H.J. Jacobs, Regional modelling of tropospheric ozone distributions and budgets, in *Atmospheric Ozone Dynamics: I. Global Environmental Change*. NATO ASI Series, vol. 53, ed. by C. Varotsos (1997), pp. 37–57. ISBN: 978-3-642-60797-4
11. M.W. Gery, G.Z. Whitten, J.P. Killus, M.C. Dodge, A photochemical kinetics mechanism for urban and regional modelling. *J. Geophys. Res.* **94**, 12925–12956 (1989). <https://doi.org/10.1029/JD094iD10p12925>
12. G.H. Golub, J.M. Ortega, *Scientific Computing and Differential Equations* (Academic Press, Boston, 1992)
13. E. Hairer, S.P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations: I. Nonstiff Problems* (Springer, Berlin, 1987). <https://doi.org/10.1002/zamm.19880680638>
14. E. Hairer, G. Wanner, *Solving Ordinary Differential Equations: II Stiff and Differential-Algebraic Problems* (Springer, Berlin, 1991)
15. H. Hass, A. Ebel, H. Feldmann, H.J. Jacobs, M. Memmesheimer, Evaluation studies with a regional chemical model (EURAD) using air quality data from the EMEP monitoring network. *Atmos. Environ.* **27A**, 867–887 (1993). [https://doi.org/10.1016/0960-1686\(93\)90007-L](https://doi.org/10.1016/0960-1686(93)90007-L)
16. H. Hass, H.J. Jacobs, M. Memmesheimer, Analysis of a regional model (EURAD) near surface gas concentration predictions using observations from networks. *Meteorol. Atmos. Phys.* **57**, 173–200 (1995). <https://doi.org/10.1007/BF01044160>
17. W. Hundsdorfer, J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction* (Springer, Berlin, 2003)
18. J.D. Lambert, *Numerical Methods for Ordinary Differential Equations* (Wiley, New York, 1991)
19. L.K. Peters, C.M. Berkowitz, G.R. Carmichael, R.C. Easter, G. Fairweather, S.J. Ghan, J.M. Haies, L.R. Leung, W.R. Pennell, F.A. Potra, R.D. Saylor, T.T. Tsang, The current state and future direction of Eulerian models in simulating tropospheric chemistry and transport of trace species: a review. *Atmos. Environ.* **29**, 189–222 (1995). [https://doi.org/10.1016/1352-2310\(94\)00235-D](https://doi.org/10.1016/1352-2310(94)00235-D)
20. L.F. Richardson, The deferred approach to the limit, I—single lattice. *Philos. Trans. R. Soc. Lond. Ser. A* **226**, 299–349 (1927). <https://doi.org/10.1098/rsta.1927.0008>
21. L.F. Shampine, *Numerical Solution of Ordinary Differential Equations* (Chapman & Hall, New York, 1994). <https://doi.org/10.1201/9780203745328>
22. W.R. Stockwell, F. Kirchner, M. Kuhn, A new mechanism for regional atmospheric chemistry modelling. *J. Geophys. Res.* **102**, 25847–25879 (1997). <https://doi.org/10.1029/97JD00849>
23. A. Venkatram, P.K. Karamchandani, P.K. Misra, Testing a comprehensive acid deposition model. *Atmos. Environ.* **22**, 737–747 (1988). [https://doi.org/10.1016/0004-6981\(88\)90011-X](https://doi.org/10.1016/0004-6981(88)90011-X)

24. P. Zannetti, *Air Pollution Modelling: Theories, Computational Methods and Available Software* (Springer, New York, 1990). <https://doi.org/10.1007/978-1-4757-4465-1>
25. Z. Zlatev, Stability properties of variable stepsize variable formula methods. *Numer. Math.* **31**, 175–182 (1978). <https://doi.org/10.1007/BF01397474>
26. Z. Zlatev, Zero-stability properties of the three-ordinates variable stepsize variable formula methods. *Numer. Math.* **37**, 157–166 (1991). <https://doi.org/10.1007/BF01398250>
27. Z. Zlatev, Consistency and convergence of general multistep variable stepsize variable formula methods for ordinary differential equations. *Computing* **31**, 47–67 (1993). <https://doi.org/10.1007/BF02247936>
28. Z. Zlatev, Advances in the theory of variable stepsize variable formula methods for ordinary differential equations. *Appl. Math. Appl.* **31**, 209–249 (1989). [https://doi.org/10.1016/0096-3003\(89\)90120-3](https://doi.org/10.1016/0096-3003(89)90120-3)
29. Z. Zlatev. *Computer Treatment of Large Air Pollution Models* (Kluwer Academic Publishers, Dordrecht, 1995) (now distributed by Springer, Berlin). <https://link.springer.com/book/10.1007/2F978-94-011-0311-4>
30. Z. Zlatev, Impact of future climate changes on high ozone levels in European Suburban Areas. *Clim. Change* **101**, 447–483 (2010). <https://doi.org/10.1007/s10584-009-9699-7>
31. Z. Zlatev, I. Dimov, *Computational and Numerical Challenges in Environmental Modelling* (Elsevier, Amsterdam, 2006)
32. Z. Zlatev, I. Dimov, I. Faragó, K. Georgiev, Á. Havasi, Large-scale air pollution modeling in europe under different climatic scenarios. *Int. J. Big Data Min. Glob. Warm.* **1**(2), 28 (2019). <https://doi.org/10.1142/S2630534819500098>
33. Z. Zlatev, I. Dimov, I. Faragó, K. Georgiev, Á. Havasi, Explicit Runge-Kutta methods combined with advanced versions of the richardson extrapolation. *Comput. Methods Appl. Math.* **20**(4), 739–762 (2020). <https://doi.org/10.1515/cmam-2019-0016>
34. Z. Zlatev, I. Dimov, I. Faragó, K. Georgiev, Á. Havasi, Studying the stability properties of nine versions of the Richardson Extrapolation (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Zahari Zlatev¹ · Ivan Dimov² · István Faragó^{3,4} · Krassimir Georgiev² · Ágnes Havasi⁵ 

¹ Department of Environmental Science, Aarhus University, Roskilde, Denmark

² Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria

³ MTA-ELTE Numerical Analysis and Large Networks Research Group, Budapest, Hungary

⁴ Institute of Mathematics, Budapest University of Technology and Economics, Egrý J. u. 1., 1111 Budapest, Hungary

⁵ Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University, Budapest, Hungary