Geosci. Model Dev., 11, 3089–3108, 2018
https://doi.org/10.5194/gmd-11-3089-2018

Geoscientific Model Development
Open Access

Quasi-Newton methods for atmospheric chemistry simulations: implementation in UKCA UM vn10.8

Emre Esentürk[1,2], **Nathan Luke Abraham**[1,3], **Scott Archer-Nicholls**[1], **Christina Mitsakou**[1,a], **Paul Griffiths**[1,3], **Alex Archibald**[1,3], and **John Pyle**[1,3]

[1]Department of Chemistry, University of Cambridge, Cambridge, UK
[2]Mathematics Institute, University of Warwick, Coventry, UK
[3]National Centre for Atmospheric Science, Cambridge, UK
[a]currently at: the Centre for Radiation, Chemical Environments and Hazards, Public Health England, Chilton, UK

Correspondence: Emre Esentürk (e.esenturk.1@warwick.ac.uk)

Received: 6 February 2018 – Discussion started: 26 February 2018
Revised: 22 May 2018 – Accepted: 8 June 2018 – Published: 1 August 2018

Abstract. A key and expensive part of coupled atmospheric chemistry–climate model simulations is the integration of gas-phase chemistry, which involves dozens of species and hundreds of reactions. These species and reactions form a highly coupled network of differential equations (DEs). There exist orders of magnitude variability in the lifetimes of the different species present in the atmosphere, and so solving these DEs to obtain robust numerical solutions poses a "stiff problem". With newer models having more species and increased complexity, it is now becoming increasingly important to have chemistry solving schemes that reduce time but maintain accuracy. While a sound way to handle stiff systems is by using implicit DE solvers, the computational costs for such solvers are high due to internal iterative algorithms (e.g. Newton–Raphson methods). Here, we propose an approach for implicit DE solvers that improves their convergence speed and robustness with relatively small modification in the code. We achieve this by blending the existing Newton–Raphson (NR) method with quasi-Newton (QN) methods, whereby the QN routine is called only on selected iterations of the solver. We test our approach with numerical experiments on the UK Chemistry and Aerosol (UKCA) model, part of the UK Met Office Unified Model suite, run in both an idealised box-model environment and under realistic 3-D atmospheric conditions. The box-model tests reveal that the proposed method reduces the time spent in the solver routines significantly, with each QN call costing 27 % of a call to the full NR routine. A series of experiments over a range of chemical environments was conducted with the box model to find the optimal iteration steps to call the QN routine which result in the greatest reduction in the total number of NR iterations whilst minimising the chance of causing instabilities and maintaining solver accuracy. The 3-D simulations show that our moderate modification, by means of using a blended method for the chemistry solver, speeds up the chemistry routines by around 13 %, resulting in a net improvement in overall runtime of the full model by approximately 3 % with negligible loss in the accuracy. The blended QN method also improves the robustness of the solver, reducing the number of grid cells which fail to converge after 50 iterations by 40 %. The relative differences in chemical concentrations between the control run and that using the blended QN method are of order $\sim 10^{-7}$ for longer-lived species, such as ozone, and below the threshold for solver convergence (10^{-4}) almost everywhere for shorter-lived species such as the hydroxyl radical.

1 Introduction

With the advent of supercomputers, simulating the atmosphere using computational models has become an integral part of atmospheric science research, complementing experimental measurements, in situ and remote observations. Model predictions are playing an increasingly important role in both purely scientific investigations and public policy making (IPCC, 2013; Glotfelty et al., 2017). In recent years, increasing computational power has enabled the de-

Published by Copernicus Publications on behalf of the European Geosciences Union.

velopment of coupled chemistry–climate models (Morgenstern et al., 2009) which determine the chemical evolution and transport (Lauritzen et al., 2009) of trace atmospheric constituents, such as long-lived greenhouse gases, ozone, nitrogen oxides, volatile organic compounds and aerosol particles, and their influence on the environment, air quality and human health (Heal et al., 2013; Lamarque et al., 2013; O'Connor et al., 2014; Tilmes et al., 2015; Collins et al., 2017). These models require globally accurate predictions over time frames that span decades (Lamarque et al. 2013), involving chemical reactions of species with lifetimes ranging from sub-seconds to centuries (Whitehouse et al., 2004), making the task computationally very expensive.

The UK Chemistry and Aerosols (UKCA) model is part of the Met Office Unified Model (UM) (Cullen, 1993; Hewitt et al., 2011) and works as its chemistry (Morgenstern et al., 2009; O'Connor et al., 2014) and aerosol (Mann et al., 2010) component. Hereafter, we refer to UM-UKCA as the fully coupled chemistry–climate model and refer to the individual submodules as UKCA and UM. Solving the chemistry in UKCA comes at a significant cost as it is one of the most expensive components in the UM-UKCA model. As coupled chemistry–climate models become more complex and the description of chemistry more involved, the need for computationally economic methods will be in higher demand. Hence, it makes sense to investigate ways of increasing the speed of the existing schemes with the goal of little or no sacrifice in accuracy.

Problems of a similar kind appear in other fields such as combustion systems which contain possibly reduced physical dynamics but more intensive chemistry (up to thousands of reactions) (Lu et al., 2009) and aerosol microphysics and dynamics (Mitsakou et al., 2005). Mathematically, these systems are represented by complex networks of coupled differential equations (DEs) which one must solve numerically. There is no universal best numerical method that works for every type of DE. Often one needs to choose the most reasonable method according to the need (e.g. ease of incorporating/modifying in model, solution CPU cost/time, accuracy). The numerical methods available can be conveniently categorised as explicit or implicit. Explicit methods are direct integration methods that work for many types of conventional problems but have worse stability properties, while implicit methods are more involved and indirect in calculations but have superior stability properties (Atkinson, 1989; Sandu et al., 1997; Damian et al., 2002). Generally, explicit methods are quicker than implicit methods at integration of single iteration steps but can fall behind in the total integration cost due to the extra efforts to ensure stability (generally by halving the time steps). When it comes to atmospheric chemistry calculations, the main stumbling block against getting stable solutions is the problem of stiffness, which, broadly speaking, originates from different chemical reactions having orders of magnitude different timescales (Cariolle et al., 2017). If one uses an explicit DE method, the (approximate) concentration values of the next time step are calculated based on the tendencies at the current time. This makes it extremely hard to choose a time step which is short enough to capture the chemical changes and preserve stability but also long enough to make the calculations feasible for computers. A good way to overcome this difficulty is by using an implicit method where tendencies are not based on current values but treated as unknowns to be solved (along with the new concentration values). This greatly increases the stability of solutions at the cost of a series of extra calculations for each time step. But again, there is no single best implicit method which is suitable for all types of stiff problems. In fact, there are families of numerical schemes available for each category (Atkinson, 1989). It is therefore desirable for any proposed new method to be flexible enough so that they can be appended to the existing solver algorithms without substantial change. This is the aim of the proposed method here.

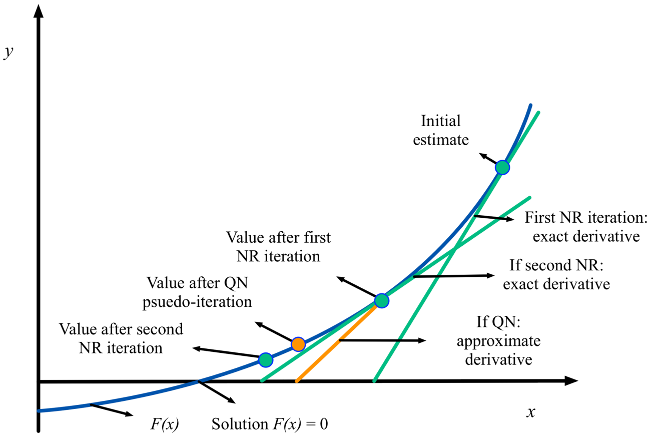
y
x
Initial estimate
First NR iteration: exact derivative
Value after first NR iteration
If second NR: exact derivative
Value after QN psuedo-iteration
If QN: approximate derivative
Value after second NR iteration
F(x)
Solution F(x) = 0

Figure 1. Illustration of application of the QN method (adopted in our work) to find the root of a function of one variable.

As will be detailed further in the text, a common feature of the many currently available implicit schemes is the solution of large systems of nonlinear differential equations iteratively (Ortega and Rheinboldt, 1970; Brandt, 1977; Kelley, 1995). At each time step, expensive subroutines have to be called several times; this is the main source of computational cost of the chemical time integration. These subroutines typically include (i) construction of a Jacobian (derivative of a function in higher dimensions) and (ii) a Newton–Raphson-type iterative algorithm to solve the nonlinear algebraic equations (associated with the nonlinear differential equations). To overcome the high costs, methods that avoid or reduce Jacobian construction have gained popularity in recent years (Brown and Saad, 1990; Chan and Jackson, 1984; Knoll and Keyes, 2004; Viallet et al., 2016, and the references therein). Our motivation for this work is somewhat similar in that we use approximations of the Jacobian to reduce the costs of the solver.

Here, we develop an approach which reduces the costs of expensive routines by partly recycling the information generated within the iterations. The method is based on exploiting this information in a way that enables one to take extra steps forward for the desired solution without going through the costly parts of the cycle. The approach is an adaptation of the quasi-Newton (QN) methods (Broyden, 1965; Shanno, 1970; Fletcher, 1970; Goldfarb, 1970; Davidon, 1991), fused into the classical Newton–Raphson (NR) method, which are commonly used for solving large systems of nonlinear algebraic equations.

The main idea behind the QN method is illustrated in Fig. 1. The objective of finding species concentrations after a short time interval can be transformed into finding the roots of a nonlinear function, which, in Fig. 1, is represented as a function $F(x)$ of a single variable x. Numerically, the task of finding the root of the function can be achieved by the NR algorithm which is based on finding the x intercepts following the tangent lines of values of the function (the green lines in Fig. 1). The root is obtained by simply re-evaluating the function at each x intercept and iterating the process. The QN method uses an approximation for the tangent line (instead of an exact derivative), the orange line in Fig. 1, so that computing the "new" x intercept is quicker. In higher dimensions (e.g. when solving for multiple chemical species), finding the exact derivative is equivalent to calculating the Jacobian matrix, while the QN method uses an approximate Jacobian, saving considerable computation time. A key point of the implementation is that the additional internal QN iterations do not replace the NR iterations completely. Rather each QN iteration works in and is fed by the current NR iteration.

Our adaptation of the QN method uses an "inverse update" approximation (Kvaalen, 1991) instead of the more commonly used "forward updates" (Broyden, 1965). We demonstrate that the approach improves the convergence rate significantly with respect to the number of main NR iterations and saves computational time. We further argue that using our mixed-method approach makes the algorithm more robust against "stiff environments" as it reduces the probability of the solver failing to converge on a solution and restarting using a shorter time step. We also test how the solutions (chemical concentrations of species) are affected over a long period of integration. We show that the differences in prognostic variables between our suggested QN method and the classical NR method are negligible and do not grow in time.

The structure of this article is as follows. In Sect. 2, we describe the UM-UKCA model and give a brief summary of its basic features. We then outline the current algorithm that handles the reaction kinetics by solving systems of nonlinear ordinary differential equations (ODEs) followed by our suggested modification using quasi-Newton methods. We further discuss why and how this modification works, its advantages and its possible dangers. In Sect. 3, we report results of our computational experiments carried out under both a controlled box-model environment and as part of the full 3-D Met Office UM-UKCA model. We compare the results of the code-modified runs with the control runs from the perspective of computational savings and differences in the concentrations/mixing ratios of chemical species, and discuss related matters with regard to parallel computing clusters. In Sect. 4, we conclude the paper by summarising and highlighting our results and pointing to possible future directions.

2 The UKCA model

UM-UKCA, originally developed by the National Centre for Atmospheric Science and the UK Met Office, was designed as a framework for atmospheric chemistry and aerosol computations that operates under the Met Office Unified Model (UM) platform and models atmospheric chemistry and aerosol fields that can feed back onto the model dynamics via the model radiation scheme (Morgenstern et al., 2009; O'Connor et al., 2014). It computes a number of possible physical–chemical processes taking place in the atmosphere such as radiation, photolysis, emissions, wet/dry deposition and clouds. It is coupled to the UM transport dynamics sequentially; that is, transport routines and chemistry–aerosol routines are performed one after another (operator splitting) with adjustable frequency. Currently in its global configuration, for transport, a time step of 20 min is used, whilst a chemical time step of 1 h is used to update the new concentrations of species in the model.

A number of chemical schemes are available in UKCA for modelling different parts of the atmosphere (troposphere, stratosphere, etc.) with varying model details (e.g. radiative feedback switched on/off). In this paper, we use the more general stratospheric–tropospheric coupled scheme with and without an online aerosol mode (either using GLOMAP mode (Mann et al., 2010) or aerosol climatologies) to demonstrate our results. The pure stratospheric–tropospheric mode (StratTrop) contains 75 species and consists of 283 chemical reactions (Banerjee et al., 2016). When GLOMAP-mode aerosols are activated, 12 additional tracers are added to the system and a total of 306 reactions represent the atmospheric chemistry. The StratTrop chemical mechanism is solved using an implicit backward Euler scheme under the ASAD framework (Carver et al., 1997; Wild and Prather, 2000), as described in detail below, while photolysis is computed using the Fast-JX scheme (Wild et al., 2000). The details of these schemes can be found in Abraham et al. (2012). The UM-UKCA version used here is vn10.6.1, in the Global Atmosphere 7.1 configuration, which is a development of the UM-UKCA GA6 configuration (Walters et al., 2017).

In addition to the full 3-D UM-UKCA model, we also use a box-model version of UKCA (hereafter referred to as UKCA_BOX) to gain better control of the chemistry part of our simulations. UKCA_BOX is designed as a development tool using the same UKCA code, branched from version 10.1

of the UM-UKCA, but with the rest of the UM-UKCA model removed and replaced with inputs that feed the UKCA code with the same information as if it were a single grid cell in the full 3-D model. The box model uses the same StratTrop (CheST) chemical mechanism, ASAD chemical solver and Fast-JX photolysis scheme as the full 3-D model but does not have any emissions, deposition or transport. As it runs for only a single grid cell, it can be run cheaply on a single processor across many test cases. Thus, it is ideal for testing and optimising the chemical solver in UKCA over a wide range of idealised chemical environments.

In the following sections, we discuss the chemical time integration schemes in the UKCA package for determining the new tracer concentrations and chemical tendencies. All numerical schemes are implemented using the Fortran 95 language. The code is available in the UM-UKCA trunk from version 10.8. Branches are also available in vn10.7 and vn10.6.1.

2.1 Chemical evolution in the UKCA

The time integration for the gas-phase chemistry in UKCA is carried out by the ASAD package which provides a flexible framework for adding and removing new reactions/species (Carver et al., 1997; Wild and Prather, 2000). The UKCA version of the ASAD package uses a backward Euler numerical scheme to compute the new species concentrations at the next chemical time step. One of the reasons for this choice is that the relevant timescales of the reactions of species vary over many orders of magnitudes depending on the location and time of the reactions, which makes the system extremely stiff. The backward Euler method is an implicit scheme which has superior numerical stability properties to almost all other explicit or semi-explicit methods and hence works particularly well with stiff systems (Atkinson, 1989). This enables the use of longer time steps and makes long time integrations feasible. The drawback is that, as in all implicit schemes, it demands that systems of nonlinear algebraic equations are solved at each time step, requiring extra calculations and so increasing the computational cost significantly.

These heavy costs can be partly reduced by exploiting the fact that the coupling among species is “loose” in the sense that each species reacts with several other species but not all. This makes the Jacobian sparse and allows for the use of sparse matrix methods which significantly cuts costs. This approach was implemented in the UM-UKCA model (see Morgenstern et al., 2009).

2.2 Numerical implementation in the existing solver

The reaction kinetics in the atmosphere can be represented, mathematically, as a system of nonlinear ODEs where the initial values are prescribed. Emissions and dry/wet deposition enter these equations as source and sink terms. The task of determining the change in chemical species concentrations is equivalent to solving the coupled nonlinear system numerically.

Let $\boldsymbol{c}\,(t) = (c_1\,(t)\,c_2\,(t)\,, \ldots, c_N\,(t))$ denote the vector of species concentrations at a given time. Then, the species evolve according to

$$\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}t} = \boldsymbol{f}\,(\boldsymbol{c}) = \mathbf{P}(\boldsymbol{c}) - \mathbf{L}\,(\boldsymbol{c}) + \mathbf{E} - \mathbf{D}_{\text{wet}}(\boldsymbol{c}) - \mathbf{D}_{\text{dry}}(\boldsymbol{c}) \tag{1}$$

$$\boldsymbol{c}(0) = \boldsymbol{a}, \tag{2}$$

where $\boldsymbol{f}$ is the nonlinear vector function (tendencies) given by the production and loss terms $\mathbf{P}, \mathbf{L}$, emissions $\mathbf{E}$, and wet and dry depositions $\mathbf{D}_{\text{wet}}$, $\mathbf{D}_{\text{dry}}$. The vector $\boldsymbol{a}$ is the initial concentration. The variables in bold–italic font are understood to be vectors. In the current implementation, emissions are treated separately during the boundary-layer mixing step, and dry deposition occurs throughout the boundary layer.

To solve Eq. (1) numerically using a backward Euler scheme, we discretise the time variable, so the discrete equation takes the form

$$\frac{\boldsymbol{c}(t_* + \Delta t\,) - \boldsymbol{c}(t_*)}{\Delta t} = \boldsymbol{f}\,(\boldsymbol{c}(t_* + \Delta t\,))\,, \tag{3}$$

where t_* is the current time and Δt is the difference between the next chemical time step and current time. The unknown $\boldsymbol{c}(t_* + \Delta t\,)$, the vector of species concentrations at the next chemical time step, appears on both sides of the nonlinear equation which can be solved numerically using a NR algorithm.

2.3 Newton–Raphson scheme

Here, we give a brief description of the NR method, which will prepare the ground for discussion of our contribution. Setting $t = t_* + \Delta t$ and $\boldsymbol{c}\,(t_*) = \boldsymbol{c}_*$ for brevity, we first write the discretised ODE (Eq. 3) in the standard form of an algebraic equation (AE); that is,

$$\mathbf{F}\,(\boldsymbol{c}(t)) = \frac{\boldsymbol{c}(t) - \boldsymbol{c}_*}{\Delta t} - \boldsymbol{f}\,(\boldsymbol{c}(t)) = 0. \tag{4}$$

The NR scheme starts with an initial guess (e.g. solution from the previous time step or a first-order predictor) followed by an iteration algorithm in which the following system of linear equations is solved:

$$\mathbf{J}(\boldsymbol{c}^k)(\Delta\boldsymbol{c}^k) = -\mathbf{F}(\boldsymbol{c}^k), \tag{5}$$

where $\mathbf{J}(\boldsymbol{c}^k)$ (or simply $\mathbf{J}^k$) is the Jacobian at the kth iterate and $\Delta\boldsymbol{c}^k = \boldsymbol{c}^{k+1} - \boldsymbol{c}^k$ is the increment (still within the same chemical time step). At each iteration, by solving a linear equation of the form of Eq. (5), our initial guess will be improved and approach the actual solution of Eq. (4) as the procedure is repeated (Atkinson, 1989).

The linear equation (Eq. 5) can also be written in the form

$$\left(\Delta\boldsymbol{c}^k\right) = \mathbf{H}(\boldsymbol{c}^k)\mathbf{F}(\boldsymbol{c}^k), \tag{6}$$

where $\mathbf{H}(\boldsymbol{c}^k)$ (or simply $\mathbf{H}^k$) is the negative of the inverse of the Jacobian $(-\mathbf{J}^k)^{-1}$. This form will be particularly useful when we explain our improvement of the current method.

In the current UKCA implementation, each major calculation step of the ODE solution algorithm is carried out by a separate routine as shown in Fig. 2a. The main solving engine begins by calculating the current tendencies (right-hand side of Eq. 1) using the updated chemical concentrations from the previous time step (Step 1 in Fig. 2a). Then an initial predictor guess (forward Euler type) is calculated to be used in the following iterative loop. After that, the Jacobian is calculated using the exact quadratic form of the nonlinear reaction rates (Step 2). This step is followed by the solution of the linear Eq. (6) (Step 3). After the new increment $(\Delta\boldsymbol{c}^k)$ is calculated, convergence is tested to determine whether $\Delta\boldsymbol{c}^k$ is within our tolerance limit (which is set to a relative change of 10^{-4} in the current version). If the routine passes the convergence test, the solver exits and concentrations at the next time step are output; otherwise, the process repeats until it converges on a stable solution. If the solution fails to converge after a set number of iterations (50 in the current version), is unstable or diverges, the routine will exit and repeat using a smaller time step (typically by halving the time step). The expensive parts of the above procedure are, particularly, Steps 2 and 3 (Fig. 2a), and our goal is to reduce the number of calls to these steps as we show in the next section.

2.4 Quasi-Newton algorithm

We noted above that the expensive parts of the chemical integration are the Jacobian construction and solution of a system of linear equations at each iteration. Our strategy is based on the idea of using QN methods to minimise the number of iterations in the main NR solving loop, thereby reducing the number of Jacobian reconstructions and linear systems to be solved.

In QN methods, the use of exact Jacobian at every iteration is abandoned. Instead it is approximated in a way that will satisfy certain imposed conditions. The ideas behind these (secant) methods, which date back to Broyden (1965), Shanno (1970), Fletcher (1970), Goldfarb (1970) and Davidon (1991) resemble using the inverse quotient of a function (of one variable) to replace the reciprocal of the exact derivative of the same function (see Fig. 1). The price of this avoidance is a slowdown in convergence (not quadratic as in the NR algorithm but still super-linear). In general, this strategy is more profitable since the slowdown in the convergence rate can be compensated by the substantial time gain obtained from bypassing the other costly steps compared to the time lost in the number of iterations.

Our implementation is somewhat different from the standard quasi-Newton methods in that Newton–Raphson iterations are not completely replaced by the QN iterations. Rather, QN iterations are fused into the existing NR loop and implemented only if a chosen criterion is met. In this sense, the new algorithm is a mixed method which uses both NR and QN methods as needed. This way keeps the changes to the existing algorithm minimal and makes the method flexible and practical to use. Despite this relatively small change in the algorithm, the computational gain in return is considerable.

Diagrammatically (see Fig. 2b), the approach works as follows. If the desired convergence has not taken place after the end of the Newton–Raphson iteration, then instead of moving on to the next iteration and reconstructing the Jacobian from scratch (Step 2), we make a pseudo-iteration and form an "effective approximation" for the inverse of the Jacobian using the concentrations already computed (Step 5). Step 6 follows in which we resolve for the newer concentration values making use of the information available from Step 3. So, a full NR iteration is effectively replaced by a QN pseudo-iteration taking much less time. These measures are quantified in Sect. 3.1.

In the above description, we refer to the "effective approximation" of the inverse of the Jacobian. However, in practice, we do not strictly construct an approximate "inverse" since taking the inverse of a matrix brings more expense. Rather, the remnants of the main NR iteration (the Jacobian from Step 2, concentrations from Step 3) are recycled and used in the approximation scheme for the inverse of the Jacobian (Broyden approximation). Schematically, after the main Newton–Raphson route, we perform Steps 4–6 shown in Fig. 2b, which is formalised below.

We use a particular, Broyden-type inverse approximation scheme (Kvaalen, 1991), which is given by the following form

$$\mathbf{H}^k_{\mathbf{app}} = \mathbf{H}^k - \frac{\Delta\boldsymbol{c}^k + \mathbf{H}^k(\Delta\mathbf{F}(\boldsymbol{c}^k))}{(\Delta\boldsymbol{c}^k)^{\mathrm{T}}(\Delta\boldsymbol{c}^k)}(\Delta\mathbf{F}(\boldsymbol{c}^k))^{\mathrm{T}}, \tag{7}$$

where $\Delta\boldsymbol{c}^k = \mathbf{H}^k\mathbf{F}(\boldsymbol{c}^k)$ and $\Delta\mathbf{F}(\boldsymbol{c}^k) = \mathbf{F}(\boldsymbol{c}^k + \Delta\boldsymbol{c}^k) - \mathbf{F}(\boldsymbol{c}^k)$ are the increments from the kth main iteration step, $\mathbf{H}^k$ is the inverse of the (exact) Jacobian in the main step (at the kth iteration), and $\mathbf{H}^k_{\mathrm{app}}$ is the negative of the approximate inverse of the Jacobian in the pseudo-iteration after the kth iteration. The superscript "$\ldots^{\mathrm{T}}$" denotes the transpose of a matrix. Although the above relation requires us to know the inverse of the Jacobian, for our purposes, we do not need to compute it explicitly.

Once $\mathbf{H}^k_{\mathrm{app}}$ is determined, the new pseudo-increment $\delta\boldsymbol{c}^k$ is given by the relation

$$\delta\boldsymbol{c}^k = \mathbf{H}^k_{\mathrm{app}}\mathbf{F}(\boldsymbol{c}^k + \Delta\boldsymbol{c}^k). \tag{8}$$

Taking $\mathbf{H}^k_{\mathrm{app}}$ from Eq. (7) and placing it into Eq. (8) gives

$$\delta\boldsymbol{c}^k = \mathbf{H}^k\mathbf{F}(\boldsymbol{c}^k + \Delta\boldsymbol{c}^k) - (\Delta\boldsymbol{c}^k + \mathbf{H}^k(\Delta\mathbf{F}(\boldsymbol{c}^k))a^k, \tag{9}$$

where $a^k = \frac{(\Delta\mathbf{F}(\boldsymbol{c}^k))^{\mathrm{T}}\mathbf{F}(\boldsymbol{c}^k+\Delta\boldsymbol{c}^k)}{(\Delta\boldsymbol{c}^k)^{\mathrm{T}}(\Delta\boldsymbol{c}^k)}$. Now, recalling $\Delta\mathbf{F}(\boldsymbol{c}^k) = \mathbf{F}(\boldsymbol{c}^k + \Delta\boldsymbol{c}^k) - \mathbf{F}(\boldsymbol{c}^k)$, and using the linearity of $\mathbf{H}^k$ and noting

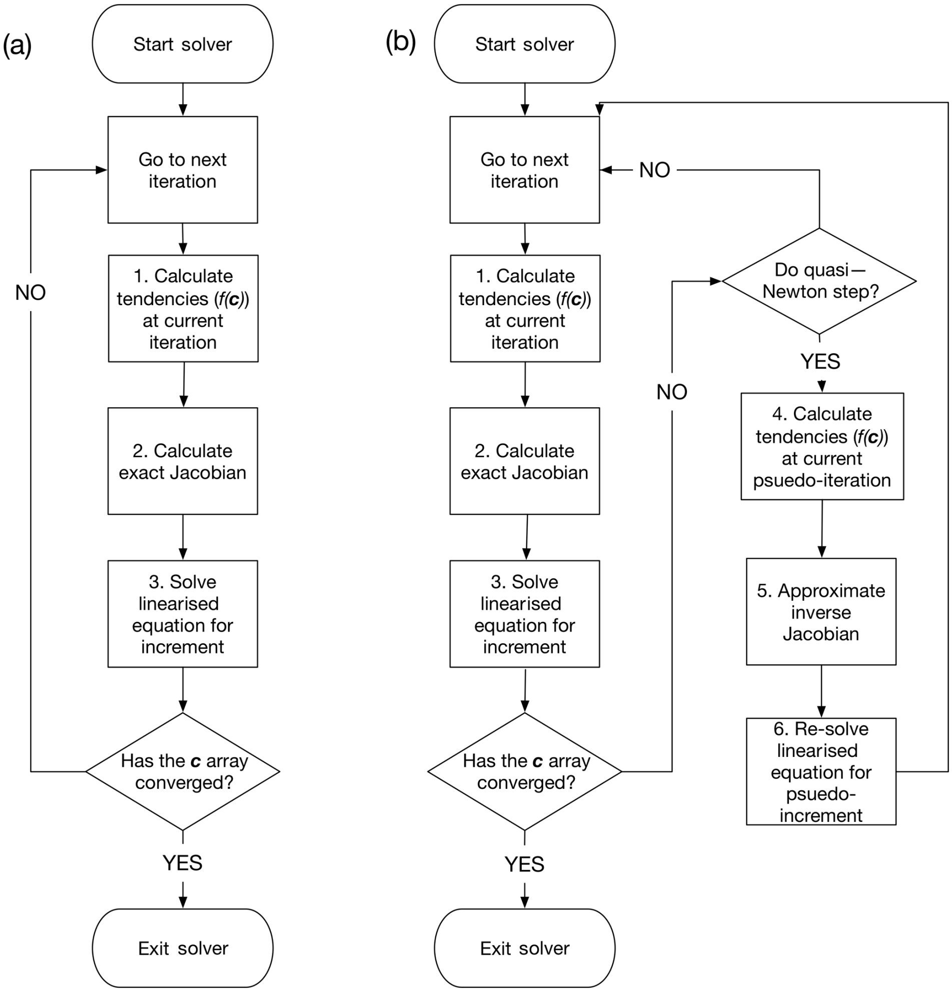
(a)
Start solver
Go to next iteration
1. Calculate tendencies (f(c)) at current iteration
2. Calculate exact Jacobian
3. Solve linearised equation for increment
Has the c array converged?
NO
YES
Exit solver
(b)
Start solver
Go to next iteration
NO
1. Calculate tendencies (f(c)) at current iteration
2. Calculate exact Jacobian
3. Solve linearised equation for increment
Has the c array converged?
YES
Exit solver
NO
Do quasi— Newton step?
YES
4. Calculate tendencies (f(c)) at current psuedo-iteration
5. Approximate inverse Jacobian
6. Re-solve linearised equation for psuedo-increment

Figure 2. Flowchart showing steps taken to numerically solve the nonlinear chemical equations using the Newton–Raphson method: as carried out in the standard version of ASAD in the UKCA chemical transport model **(a)** and in our modified version incorporating a "quasi-Newton iteration" **(b)**.

that $\Delta \boldsymbol{c}^k = \mathbf{H}^k \mathbf{F}(\boldsymbol{c}^k)$, the terms simplify:

$$\begin{aligned}\delta \boldsymbol{c}^k &= \mathbf{H}^k \mathbf{F}(\boldsymbol{c}^k + \Delta \boldsymbol{c}^k) - \mathbf{H}^k \mathbf{F}(\boldsymbol{c}^k + \Delta \boldsymbol{c}^k) a^k \\ &= (1 - a^k) \mathbf{H}^k \mathbf{F}(\boldsymbol{c}^k + \Delta \boldsymbol{c}^k). \end{aligned} \tag{10}$$

If compared, we see that Eq. (9) has the same form as Eq. (6), which can also be written in the form of Eq. (5) as

$$\mathbf{J}(\boldsymbol{c}^k)(\delta \boldsymbol{c}^k) = (1 - a^k) \mathbf{F}(\boldsymbol{c}^k + \Delta \boldsymbol{c}^k). \tag{11}$$

Now, crucially, the information of the reduced (row echelon) matrix obtained from the original Jacobian through Gaussian elimination in Step 3 (Eq. 5) is still available and can readily be used to solve the linear Eq. (10), where the only difference from Eq. (5) is on the right-hand side. This bypasses the need for computing $\mathbf{H}^k_{\text{app}}$ explicitly, saving memory and time. In effect, the method accomplishes two tasks at once: reducing the combined steps of reconstructing a new Jacobian and solving a new linear equation (in a new NR iteration) into a single step of solving a modified linear equation (in a pseudo-iteration) based on the information already available within that (main) iteration. In practical terms, this means that $\sim N^3$ numerical operations that are normally needed to solve a linear system are now reduced to $\sim N^2$ operations, which gives substantial savings within the routine. An example of the implementation of these changes is given in the pseudo-code provided in Appendix A.

3 Numerical results

In this section, we compare our results with the new method (quasi-Newton) and without (classical Newton–Raphson) when implemented in the current version of the UKCA solver. We consider the effectiveness of the algorithm on a single processor with, i.e. UKCA_BOX, as well as on a high-performance parallel computing (HPC) platform (ARCHER) with the full 3-D UM simulations. In both cases, our analysis will be two-fold: comparison of computational performance (savings, robustness, etc.) and comparison of predicted model values. We show that, although the chemistry step alone takes 5 to 10 % of the entire computations, there is a noticeable speed-up when the chemistry component is modified in the way suggested without causing any significant error in prognostic variable values. This also improves the robustness of the computation by reducing the number of cases during the course of entire chemical integration for which the time step has to be halved in order to converge on a solution.

3.1 UKCA_BOX simulations

To test the performance of the QN approximation method on performance of the UKCA chemistry solver, we first tested the changes in UKCA_BOX. UKCA_BOX allows us to test the performance of the QN methods under a highly controlled environment, and optimise the options for the solver based on a variety of chemical conditions.

Four standard test cases were set up for these experiments to test the behaviour of the box model in different chemical environments: Urban, Rural, Marine and Stratosphere (Strat). The initial conditions for these test cases were extracted for July from a 10-year run of the full UM-UKCA model for the year 2000 at 1.875° × 1.25° resolution, equivalent to the experiments conducted in Sect. 3.2. For the Urban, Rural and Marine scenarios, average surface chemical fields, temperature, pressure and specific humidity were extracted at surface locations over the Beijing megacity, the continental USA and the Pacific Ocean, respectively (see Table 1 for details). All UKCA_BOX experiments were run on a single processor core. The Strat scenario used zonally averaged chemical and meteorological fields at 40° N and 32 km. Full details of the scenarios are given in the Supplement. The Urban scenario is initialised with the most complex mix of chemical components and is therefore the most challenging to solve. For this reason, the analysis in the paper will focus on the Urban scenario. Results from the other scenarios are included in the Supplement.

The UKCA_BOX uses the Fast-JX photolysis scheme (Wild et al., 2000), comparable to that used in the full UM-UKCA model (Telford et al., 2013). For the purposes of these experiments, a simplified setup was used whereby photolysis turns "on" and "off" every 12 h of integration, using precalculated photolysis rates. This was done to minimise the computation of photolysis rates and create idealised scenarios with an abrupt step change at "dawn" and "dusk" to test the stability of the solver. Photolysis rates were taken from an offline run of the 1-D column Fast-JX scheme at 12:00 UTC on 1 July, 40° N at 0 and 32 km in clear-sky conditions for the Urban, Rural, Marine and Strat scenarios, respectively. Each experiment ran for 5 days with a 60 min time step (the same as the chemical time step used in the full 3-D UM-UKCA model). Without emissions, deposition or transport, the chemical evolution is completely determined by the initial conditions. Each scenario starts in a state of disequilibrium, then slowly "winds down" over the 5 days of integration.

As discussed in the previous section, the QN method is cheaper than the full NR method because it does not recalculate the full Jacobian at each iteration (Table 2). On average, one QN iteration takes 27 % of the time of a full NR iteration. Since the QN method reduces the number of NR iterations required to converge, the time taken will therefore generally be reduced. However, the QN method is not as exact as the NR method, and so there is not a one-to-one efficiency: calling the QN method many times may only reduce the number of NR iterations required by a few, and in some cases calling the QN method too many times can result in a net increase in computational burden. Finding the most efficient setup therefore becomes an optimisation problem: how can we gain the maximum reduction in NR iterations, with as few calls to the QN method as possible? In particular, we are interested in reducing the number of iterations required for the solver during the most challenging chemical states when the equations are most stiff. This will reduce the range of time taken for cores to solve each part of the domain, therefore reducing time spent waiting for all cores to catch up to the same time in the full 3-D model.

To test the range of options, we devised nine experiments for each scenario, as summarised in Table 3. The control (CNTL) experiment does not call the QN method and is identical to the solver in the release version of UKCA. The other scenarios call the QN method after one or more NR-iterations, as given by the numbers in the names of experiments in Table 3. For example, QN1 calls the QN Newton method after the first NR iteration only, QN2–3 calls it after the second and third NR iterations, and QN1+ calls the QN method after every NR iteration. In general, the first iteration of the solver is where the solution is most likely to diverge and cause stability problems, and so a dampening factor of 0.5 is applied to the QN method, as is also done on the first iteration for the NR method. As shown by the flow structure of this development (Fig. 2b), the QN method is only called if the solution has not already converged.

Figure 3 shows chemical concentrations for a selection of chemical tracers from the box model, comparing the CNTL experiment with the QN experiments, for the Urban scenario. Similar figures for the other scenarios are included in the Supplement. In this scenario, the mix of NO_x and VOCs results in production of O_3 for the first day, then a slow loss of O_3 over the next four days as concentrations of short-lived tracers decay due to the lack of fresh emissions (Fig. 3a). Overall, these results show the QN method is very accurate with negligible divergence from the CNTL experiment. The fractional differences are largest for short-lived tracers, such as OH, but are at most of the order 10^{-5} or less (Fig. 3f). For longer-lived species, such as O_3 or NO_2, fractional changes are typically $< 10^{-8}$ (Fig. 3c, i). Differences between the

Table 1. Summary of data points from UM model runs used to initialise UKCA_BOX scenarios, parameters describing atmospheric conditions of each scenario and initial concentrations of select chemical species. In each case, data are extracted from a 10-year July average run of the UM-UKCA model for the year 2000.

Scenario	Location	Height above ground (km)	Pressure (hPa)	Temperature (K)	Specific humidity ($kg\,kg^{-1}$)	O_3 (ppbv)	NO_x (ppbv)	HCHO (ppbv)
Urban	40° N, 116.4° E	0	983	300	0.0147	46.5	20.5	3.37
Rural	40° N, 260° E	0	926	304	0.0101	49.8	2.5	1.95
Marine	40° N, 180° E	0	1017	292	0.0121	25.0	0.30	0.39
Stratosphere (Strat)	Zonal average at 40° N	32	8.61	240	3.44×10^{-6}	9102	15.7	0.09

Table 2. Wall-clock times for running 1000 calls for the NR iterations and QN iteration within the UKCA_BOX model run on a single processor core.

	Full Newton–Raphson method	Quasi-Newton method	Ratio
CPU time for 1000 calls	160 ± 3.1 ms	42 ± 0.71 ms	0.2625
Wall-clock time for 1000 calls	157 ± 1.8 ms	42.9 ± 0.15 ms	0.273

CNTL and QN scenarios do not grow over time; rather, they tend to be largest in periods which are challenging to solve (at the start of the simulation and around dawn and dusk) and then decay to zero.

Time series of the number of iterations required to converge for the Urban scenario are shown in Fig. 4. Similar figures for the Rural, Marine and Strat scenarios are included in the Supplement (Figs. S1 and S2; S3 and S4; S5 and S6, respectively), which in general are found to converge in fewer iterations than the Urban case. The dashed blue line shows the number of NR iterations required to reach a stable solution at each time step, the red line shows the number of QN iterations required, and the black line shows the estimated NR-equivalent number of iterations taken to solve, using the result that QN iterations take on average 27 % of the computational time to solve compared to the NR method (Table 2). The first time step is the most difficult to solve, as the initial chemical concentrations are typically far from a steady state having been taken from monthly average values from model cells. After that, the dawn and dusk periods, the time steps immediately after photolysis is turned on and off, respectively, are the next most challenging, as changing photolysis rates causes an abrupt change in the lifetimes of many species. The inclusion of the QN method can be seen to improve the solver when the net NR-equivalent iterations (black line) are lowered compared to the CNTL scenario, and is optimal when this can be achieved with the minimum number of QN pseudo-iterations (red line, Fig. 4). While the UKCA_BOX model only solves a single case at any one time step, each core in the 3-D model will solve for many grid cells at each time step, and can only move on to the next time step once all have converged. In other words, the 3-D model is only as fast as its slowest grid cell. For this reason, the cases where the new methods reduce iteration count at the more challenging time steps (at dawn and dusk) are considered a stronger indication that they will improve integration time in the full 3-D model rather than the average.

The Urban scenario is the most challenging of the test cases to solve, due to the high initial concentrations of reactive tracers (Fig. 4). The CNTL scenario takes 12 full NR iterations to solve the first time step, then between 4 and 7 for each time step thereafter, needing 4.36 iterations on average (Fig. 4a). More iterations are required at dawn and dusk, with a maximum of seven NR iterations required at dusk. Calling the QN pseudo-iteration on the first iteration (QN1, QN1–2, QN1–3 and QN1+; Fig. 4b–e) reduces the number of NR iterations required to reach a stable solution on most time steps but increases the number of NR iterations at dawn on most days, therefore increasing the computational costs at these time steps compared to the CNTL run. The experiments with the QN method first called on the second iteration (QN2, QN2–3 and QN2+; Fig. 4f–h) consistently reduce the number of NR iterations required to reach a stable solution. Experiment QN2–3 is the most efficient of the three, reducing the number of NR iterations required, at the dusk time steps to 6, and to 3.52 on average, giving a net average of 3.89 NR-equivalent iterations counting each QN pseudo-iteration as 27 % of a full NR iteration. Experiment QN2+ shows diminishing returns compared to QN2–3, calling more QN pseudo-iterations for no reduction in NR iterations on most time steps. The experiment with QN called on the third iteration only (QN3; Fig. 3i) shows only marginal improvement compared to the CNTL scenario. Overall, QN2–3 most consistently reduces the net iteration count on average at dawn and dusk in the Urban test case. In some of the other scenarios, QN1–3 performed most efficiently (see the Supplement). However, in the Urban scenario, the QN2–3 experiment performs better at the dawn time steps, when the QN1–3 experiment performs worse than the CNTL run. QN1–3 therefore shows signs of reduced robustness during the periods which

Table 3. Summary of experiments conducted using UKCA_BOX. The control (CNTL) experiment does not call the QN method. The other experiments call the QN method after one or more NR iterations.

Call QN method on	CNTL	QN1	QN1–2	QN1–3	QN1+	QN2	QN2–3	QN2+	QN3
First iteration:	No	Yes	Yes	Yes	Yes	No	No	No	No
Second iteration:	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Third iteration:	No	No	No	Yes	Yes	No	Yes	Yes	Yes
> Third iteration:	No	No	No	No	Yes	No	No	Yes	No

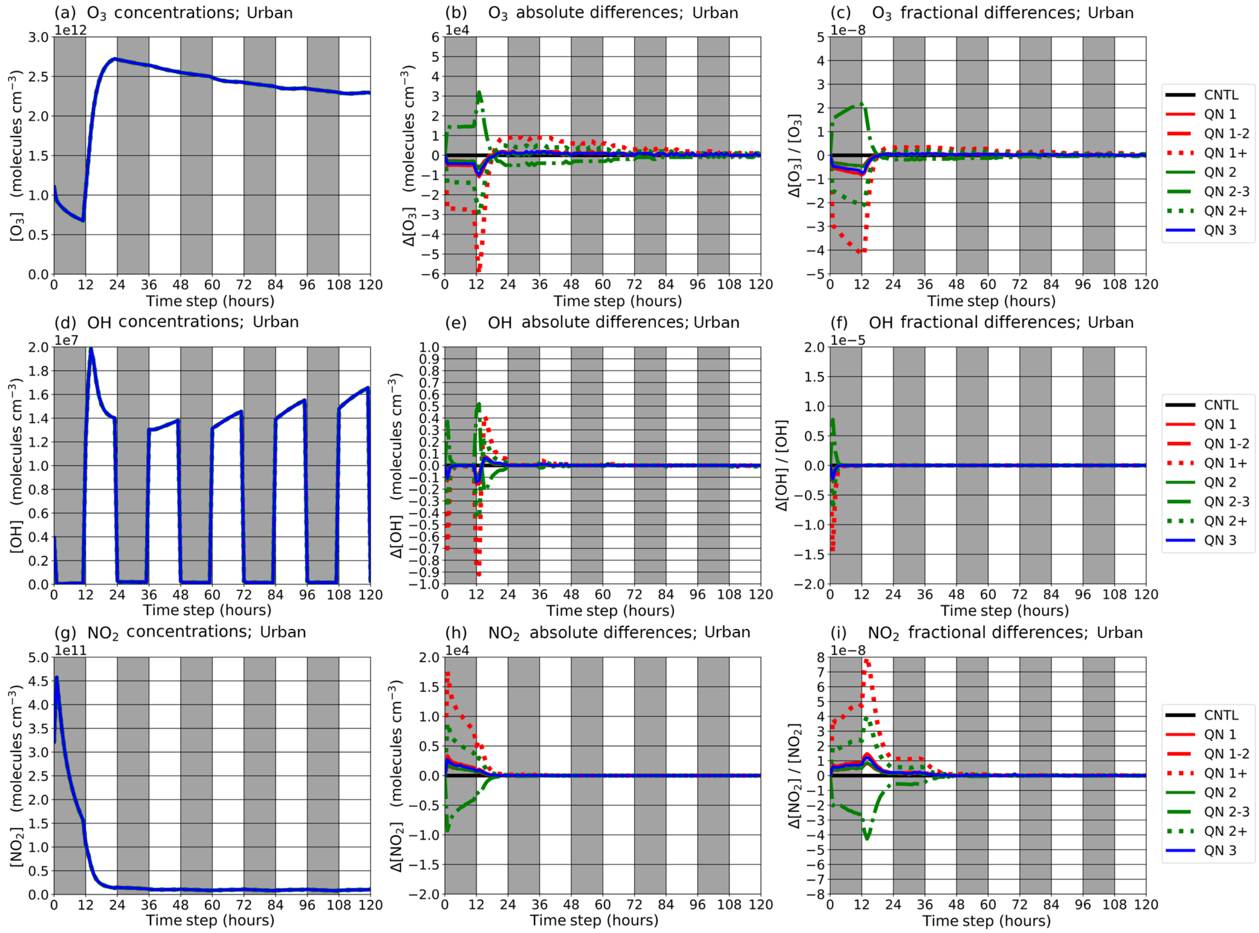

Figure 3. Concentrations of O_3, OH and NO_2 in molecules cm^{-3} from UKCA_BOX simulations of the Urban scenario. The left panels show absolute concentrations from all scenarios, with differences too small to be observed by eye. The centre and rightmost panels show absolute and fractional differences between the CNTL and QN experiments, respectively. The white bands show periods with photolysis on, and grey bands show periods with photolysis off.

are most challenging to solve, meaning it is unlikely to be able to handle the wide range of chemical states that will be simulated in the 3-D model runs. We therefore use the QN2–3 setup for the 3-D model runs, as UKCA_BOX results suggest it shows the most consistent improvements over the CNTL scenario.

3.2 UM-UKCA simulations

In this section, we report our results for the full 3-D global UM-UKCA simulations with the QN method implemented (on the original ASAD solver code) and without (classical NR method). We discuss these results from the perspectives of model performance (computational savings and stability) and prognostic evaluations (comparison of model physical values). All simulations were performed using version 10.6.1 of the model, applying the GA7.1 configuration at

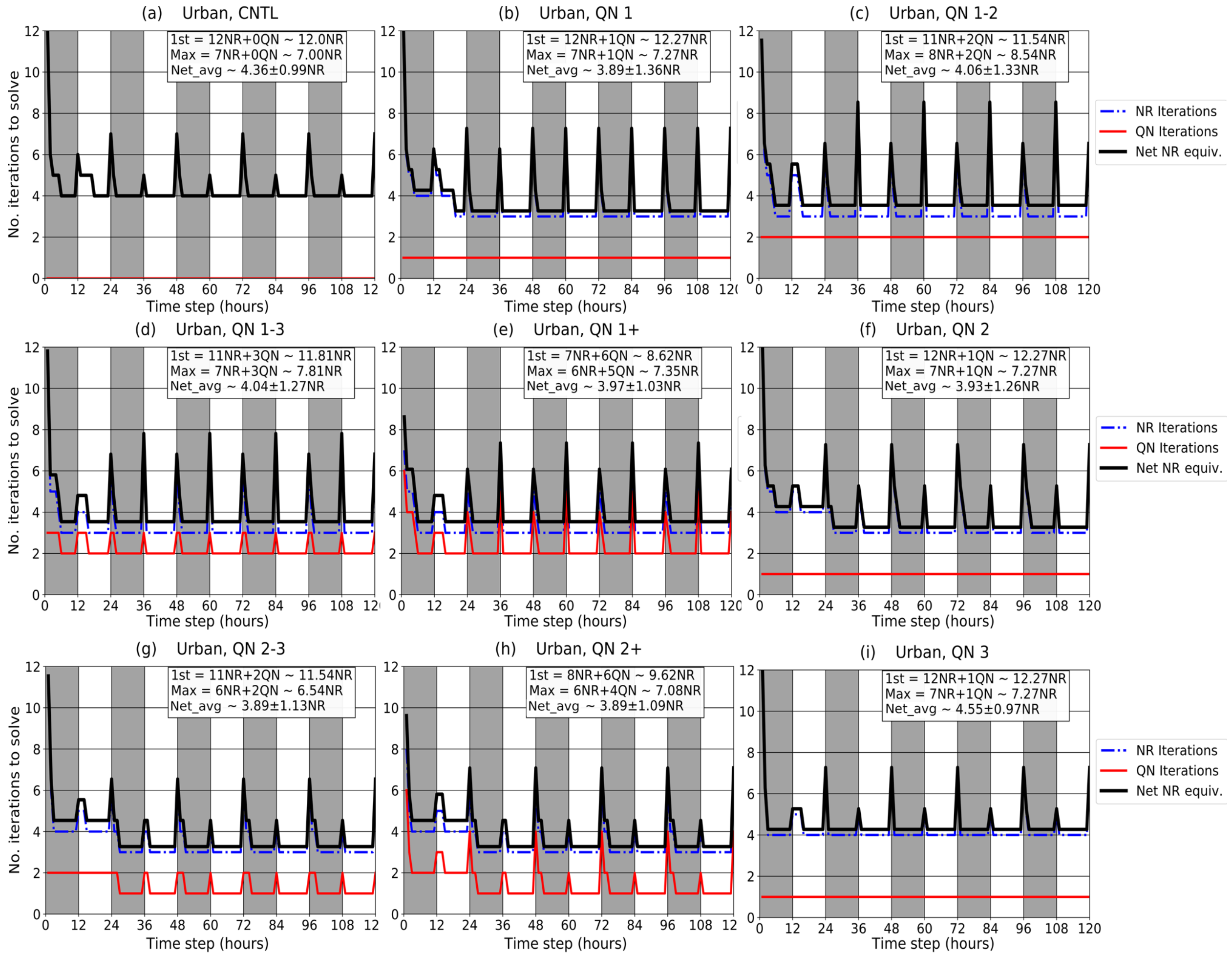

Figure 4. Plots of solver iteration (convergence) numbers for the original full NR method and QN methods, with QN pseudo-iterations only called on particular iteration(s). The CNTL scenario **(a)** only solves with NR iterations, and is equivalent to the solver in the release version of UKCA. The other eight panels call QN pseudo-iterations on one or more iterations at each time step. The blue dashed lines show the number of NR iterations required to converge on a stable solution, while the red line shows the number of QN pseudo-iterations required, and the black line total net number of NR-equivalent iterations to solved, calculated as NR + 0.27QN. The white bands show periods with photolysis on, and grey bands show periods with photolysis off. The text in each panel gives the number of NR and QN iterations required to converge on the first time step, the most difficult time step after the first, and on average across the whole period in NR-equivalent iterations.

Table 4. Computational speed-up using the QN method in comparison to the regular Newton–Raphson method.

Chemistry	Number of cores	Simulation	Mean wall-clock time for one month (s) ±2 × standard error	Speed-up (%)
StratTrop	432	CNTL QN2–3	3525.7 ± 10.6 3444.2 ± 9.5	2.31 ± 0.01
StratTrop+GLOMAP	432	CNTL QN2–3	4805.7 ± 20.6 4664.7 ± 14.2	2.93 ± 0.02

1.875° × 1.25° resolution with 85 vertical levels up to 85 km (N96L85). Emissions were the year 2000 CMIP5 emissions for all runs (Lamarque et al., 2013). Surface sea temperatures (SST) were as in Banzon et al. (2018) and Reynolds et al. (2007). Aerosols were provided via a climatology. The UM-UKCA is a nonhydrostatic model which uses a regular longitude–latitude grid and a vertical hybrid height coordinate.

We have performed three sets of numerical experiments with two slightly different configurations of UKCA. The first

Table 5. Average wall-clock time in seconds (±2 standard error) across all processors used for various UM components comparing the CNTL and QN2–3 methods. All are from 1-year simulations performed on a Cray XC40.

Chemistry	StratTrop		StratTrop+GLOMAP		StratTrop	
Cores	432		432		216	
Simulation	CNTL	QN2–3	CNTL	QN2–3	CNTL	QN2–3
Dynamics	12 123 ± 22	12 099 ± 23	15 117 ± 28	15 297 ± 27	18 881 ± 27	18 743 ± 30
Chemistry	4228 ± 26	3678 ± 16	4725 ± 28	4123 ± 19	9102 ± 96	7875 ± 75
Diagnostics	2951 ± 1	2979 ± 1	3628 ± 1	3641 ± 1	3098 ± 1	3108 ± 1
Photolysis	3038 ± 7	3038 ± 7	3041 ± 7	3030 ± 7	6082 ± 43	6084 ± 43
Convection	1833 ± 51	1828 ± 51	2367 ± 62	2366 ± 62	3648 ± 148	3637 ± 148
Radiation	1184 ± 10	1184 ± 10	1140 ± 10	1136 ± 9	2487 ± 34	2485 ± 34
UM Total	48 871 ± 0	47 730 ± 0	71 900 ± 0	70 596 ± 0	82 561 ± 0	79 600 ± 0
Chemistry speed-up (%)	13.00		12.74		13.48	
UM speed-up (%)	2.33		1.81		3.59	

version (StratTrop) uses the stratosphere–troposphere chemistry where all radiative feedback from UKCA trace gases was turned off and aerosol climatologies were used. This setup allows for changing the chemical species whilst maintaining the same wind fields between the simulations. The UM-UKCA is parallelised by breaking the domain up into a chess-board pattern of subdomains, defined by the number of processes given for the east–west (EW) and north–south (NS) directions. The solver iterates across all grid cells in the subdomain until all have reached a stable solution. Thus, the computational speed is limited by the hardest-to-converge ("stiffest") grid cell in each subdomain. This configuration was run for 20 model years using 432 cores (24EW × 18NS) in both control (CNTL) and quasi-Newton configurations (QN2–3). Additionally, four 1-year simulations were performed with additional timer diagnostics included using the Dr Hook package (ECMWF, 2013), two using 432 cores and two using 216 cores (18EW × 12NS). In all these sets of simulations, the initial start file was the same and the wind fields bit compared at the end of the simulation.

A second set of simulations was performed using the stratosphere–troposphere chemistry combined with the GLOMAP-mode aerosol scheme (StratTrop+GLOMAP). This requires additional chemical species and reactions to be included on top of the standard StratTrop chemistry. In these simulations, both CNTL and QN2–3 simulations were performed on 432 cores (24EW × 18NS) for 20 model years (equivalent to the StratTrop simulations). However, here, both aerosols (via the direct and first and second indirect effects) and ozone, methane and nitrous oxide were coupled interactively to the Met Office UM dynamics via the model radiation scheme. This means that the wind fields in these simulations were not identical as the small concentration changes introduced by the QN method resulted in global changes to the dynamical fields. Additionally, two 1-year simulations with timer diagnostics were also completed for the CNTL and QN2–3 configurations.

3.2.1 Model performance

We begin our discussion with an overview of the timing for each simulation set. These total time measurements are complemented by a robustness assessment, checking the number of times that iteration steps of the main chemistry solver are halved in order to reach the prescribed accuracy (that is, where UKCA spends more CPU in regions of stiff chemistry). This initial analysis is then expanded to a more detailed analysis via time measurement maps of the simulations and iteration maps of the chemistry solver.

Table 4 gives the total wall-clock time measurement results for the four 20-year sets of simulations (jobs). A plot of the speed-up for absolute wall-clock time is also included in the Supplement (Fig. S7). Using our suggested modification of the current algorithm leads to a net savings of ∼ 2–3 % over the full UM simulation despite the fact that the chemistry routine takes a relatively small part (5–10 %) of the entire simulation (depending on the configuration). This suggests that using a (mixed) quasi-Newton method has the potential to reduce the computational costs of other non-spatial systems with more intensive chemistry or even spatial systems modelled by partial differential equations that involves construction of a Jacobian for the computation of solutions. For the comparison of core components of the UKCA routines, we conducted 1-year long timer diagnostics analysis with the Dr Hook package. The results are tabulated in Table 5. It is found that the QN scheme speeds up the chemistry component between 12.7 and 13.5 % depending on the configuration.

A legitimate question is to check how quasi-Newton methods, which are essentially based on approximations, change the robustness of the numerical scheme. This is particularly important since the modelled systems are generally under stiff conditions which are prone to instability. A poorly designed approximate method could wash out important information on the direction of the chemical evolution and cause

Table 6. Number of times that the solver needed to halve the time step in order to avoid divergences or wild oscillations over 1 year of integration.

Chemistry	Number of cores	Simulation	Number of halving steps	Fraction of total number of solver calls
StratTrop	216	CNTL	457 344	0.00288
		QN2–3	270 101	0.00170
	432	CNTL	436 048	0.00137
		QN2–3	256019	0.00081
StratTrop+GLOMAP	432	CNTL	544 532	0.00172
		QN2–3	328 836	0.00104

the program to crash after some number of steps. To demonstrate that the approximation scheme that we propose is safe, we show in Table 6 the number of times the UKCA model halves the time step (a sign that the chemical conditions at that particular location and time are such that the solution to fails to converge, oscillate or even diverge, and therefore the time step has to be reduced). According to Table 6, with the QN modification, the occurrence of halving the time step is nearly 2 times less frequent compared to the original algorithm, suggesting that the mixed QN method can be more robust in chemically stiff environments, saving more computational time overall as halving the time step significantly increases computational costs. The parallelisation of the UM-UKCA is such that the whole model can be held up by the few grid cells which fail to converge under the normal time step. So improving the robustness of the solver potentially has much greater benefits to net computational efficiency than just the direct reduction in cost to solve the individual grid cells.

Next, we make a grid point analysis of NR iterations to understand the origin of computational savings. In general, the time that it takes the solver to calculate final chemical concentrations on a grid point depends heavily on the ambient photochemical conditions at that point and time. So, the number of iterations in which the program exits the solver loop varies significantly across the domain.

Figure 5 shows maps of the mean number of iterations to convergence (averaged over column and time) for the 1-year simulations (one chemical time step is equal to 1 model hour) with the StratTrop (216- and 432-core) and GLOMAP (432-core) schemes. The CNTL simulations (left-hand column) clearly show regions where more iterations are required. The right-hand column shows the difference in mean number of iterations to convergence when using the QN2–3 method. Not only is the mean number of NR iterations reduced globally, but greater benefit is seen in the hot-spot regions noted in the CNTL simulations.

By summing the total number of points through the 1-year period according to number of iterations, a histogram of iteration numbers is produced which neatly summarises performance of both methods (the CNTL and the QN cases). Figure 6 shows the histogram of the iteration numbers over all grid points for the 1-year simulations with the same StratTrop (216- and 432-core) and GLOMAP (432-core) schemes. The QN method greatly reduces the peak at eight iterations, and allows the majority of solutions (approximately 70 %) to be found in four or less NR iterations.

3.2.2 Model evaluation

In this section, we evaluate the accuracy of our proposed method. Recall from Sect. 3.1 that the QN method produces physical values which are very close to what the original method calculates even for fast-changing species.

We test the accuracy of the two methods by comparing the model predictions for two different species which have very different lifetimes (O_3 and OH) and are key species that chemistry–climate models need to simulate accurately (Monks et al., 2015). If the 3-D model predictions for the two species which are on the opposite sides of the lifetime spectrum are very close, then it is very likely that physical values for all other species which have intermediate lifetimes will also be close.

For comparison of differences in values, we consider only the StratTrop scenario in which ozone and other chemical feedbacks are not included. This avoids intrinsic perturbations dominating the solutions over long periods of time and ensures that the dynamics are identical between both simulations.

From the last 10-year average of two 20-year experiments (StratTrop-CNTL and StratTrop-QN2–3), we see that O_3 concentrations (here plotted as a 10-year mean for the representative month of July) for the two experiments are very similar, as seen in Fig. 7 (for zonal-mean differences on the left column and for surface differences on the right column). The same figures also show that the relative percentage differences (bottom row) between the two runs are negligible, being of the order of 0.01 % or smaller.

For the comparison of OH concentrations in the 20-year StratTrop-CNTL and StratTrop-QN2–3 experiments, Fig. 8 shows the zonal-mean differences and surface value differ-

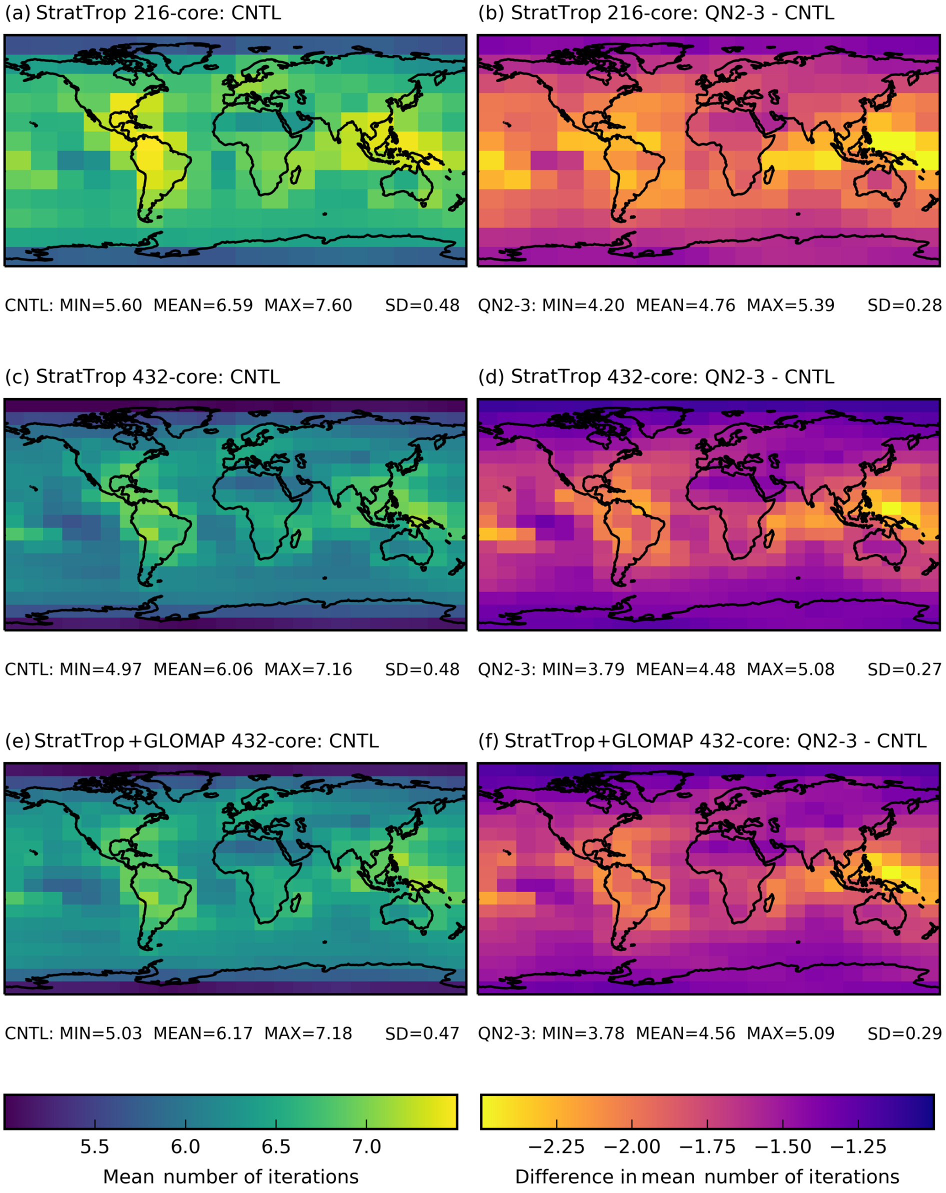
(a) StratTrop 216-core: CNTL
(b) StratTrop 216-core: QN2-3 - CNTL
CNTL: MIN=5.60 MEAN=6.59 MAX=7.60 SD=0.48
QN2-3: MIN=4.20 MEAN=4.76 MAX=5.39 SD=0.28
(c) StratTrop 432-core: CNTL
(d) StratTrop 432-core: QN2-3 - CNTL
CNTL: MIN=4.97 MEAN=6.06 MAX=7.16 SD=0.48
QN2-3: MIN=3.79 MEAN=4.48 MAX=5.08 SD=0.27
(e) StratTrop +GLOMAP 432-core: CNTL
(f) StratTrop+GLOMAP 432-core: QN2-3 - CNTL
CNTL: MIN=5.03 MEAN=6.17 MAX=7.18 SD=0.47
QN2-3: MIN=3.78 MEAN=4.56 MAX=5.09 SD=0.29
5.5
6.0
6.5
7.0
Mean number of iterations
−2.25
−2.00
−1.75
−1.50
−1.25
Difference in mean number of iterations

Figure 5. Left column **(a, c, e)**: maps of average NR iteration numbers for the three different 1-year standard UM Newton–Raphson solver (CNTL) simulations from Table 5; right column **(b, d, f)**: differences between the quasi-Newton solver (QN2–3) and the equivalent control simulation. The top plots **(a, b)** are 216-core StratTrop (18EW × 12NS), the middle plots **(c, d)** are 432-core StratTrop (24EW × 18NS), and the bottom plots **(e, f)** are 432-core StratTrop+GLOMAP. The quoted statistics are for the simulations and not for the differences.

ences in the month of July. The difference values are slightly larger but still only of the order of 0.1 % or smaller. Note that the largest percentage differences are seen in the areas with the smallest absolute OH concentrations. Almost everywhere else the fractional difference in OH is less than the tolerance of the solver (10^{-4}). It is also clear from the surface OH plots (Fig. 8b, d, f) that the differences in OH are so small that they are approaching the limits of the numerical scheme, as the subdomains solved by each processor are clearly visible (being 24 in the X direction and 18 in the Y direction). This artefact appears because all grid cells in each subdomain are iterated in the solver until all have converged and thus can introduce small numerical differences.

3.2.3 Analysis of the differences between simulations with UM-UKCA

In this subsection, we give a quantitative analysis of the differences in the physical values obtained from the computations. In the strict sense of the word, there is actually no extra "error" associated with our proposed method of computation as both the classical NR and QN approaches give approxi-

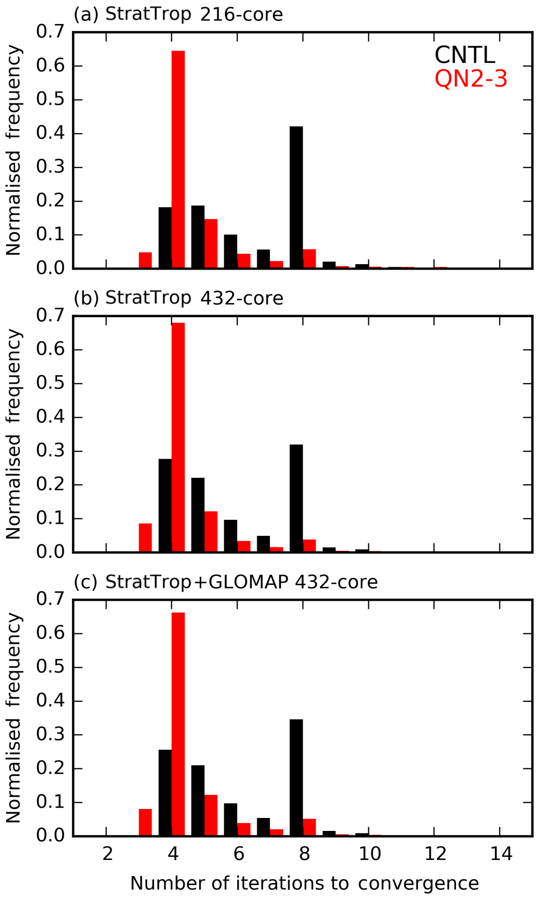

Figure 6. Histograms of the number of NR iterations to convergence for the 216-core StratTrop **(a)**, 432-core StratTrop **(b)**, and 432-core StratTrop+GLOMAP **(c)** 1-year long simulations.

mate solutions of the real DE within a chosen error tolerance (which is met by each method). Nevertheless, for completeness and comparison, we will regard the NR computations (CNTL runs) as the "true" values and measure the difference in OH and O_3 fields for the two runs using two different metrics defined below.

The figures in the previous sections provide maps of absolute and relative differences. Depending on the location of the point, these differences vary but always stay very small. In order to have a more quantitative measure of how different one particular run is from the other, we need a metric that will take into account all of the grid points and the corresponding errors. Considering the extreme low values of OH in certain regions, the most suitable metrics (Yu et al., 2006) are the normalised mean absolute difference (NMAD) and normalised root mean square difference (NRMSD) which are, respectively, defined by

$$\mathrm{NMAD_S} = \frac{\sum_i \left| c^i_{\mathrm{S,nr}}(T) - c^i_{S,qn}(T) \right|}{\sum_i \left| c^i_{\mathrm{S,nr}}(T) \right|}, \tag{12}$$

$$\mathrm{NRMSD_S} = \sqrt{\frac{\sum_i \left| c^i_{\mathrm{S,nr}}(T) - c^i_{\mathrm{S,qn}}(T) \right|^2}{\sum_i \left| c^i_{\mathrm{S,nr}}(T)) \right|^2}}, \tag{13}$$

where S denotes the species and T denotes the time at the end of the run. To measure the bias, we calculate normalised mean bias which is defined as

$$\mathrm{NMB_S} = \frac{\sum_i c^i_{\mathrm{S,nr}}(T) - c^i_{\mathrm{S,qn}}(T)}{\sum_i \left| c^i_{\mathrm{S,nr}}(T) \right|}. \tag{14}$$

Table 7 below shows the NMAD, NRMSD and NMB for the OH and O_3 species. A complete table showing NMAD, NRMSD and NMB for all species is provided in the Supplement.

We also plot the NMAD, NRMSD and NMB as a function of time (each month) in the last 10-year period for OH (Figs. S8, S9 and S10 of the Supplement, respectively) and for O_3 (Figs. S11, S12 and S13 of the Supplement, respectively). We observe that the differences are extremely small and stay bounded in time and do not grow, which indicates that the two methods reproduce essentially the same evolution. We remark that NMB values are smaller than NMAD in magnitude and do not grow in time, as expected. Similar conclusions can be drawn for the other species as shown in the Supplement.

4 Conclusions

Atmospheric chemistry simulations are at the heart of coupled chemistry–climate models. Solving the complex sets of equations that represent the evolution of species comes at a high computational cost. In this article, we introduced a version of the quasi-Newton method into the UKCA coupled climate model. The quasi-Newton method demonstrates improvements, in multiple ways, over the classical Newton–Raphson method used in the UKCA model chemistry solver.

The main benefit of the QN approach, as discussed in Sect. 3, is its ability to reduce the computational time for the simulations. The advantages, however, are not limited to reducing the costs of chemistry calculations. The computations are more robust against stiff chemical environments, thereby reducing the possibility of divergence and instability in computations. On parallel platforms, even when there is no danger of instability, robustness actually can translate into extra computational gain as the method saves further time by avoiding unnecessary wait times in the subdomains. Overall, we see a reduction in total computational costs of the whole UKCA model of approximately 3 %, corresponding to a reduction of approximately 15 % in the chemistry routines. Whilst this may not seem like a big reduction, it is significant given the high costs associated with the rest of the

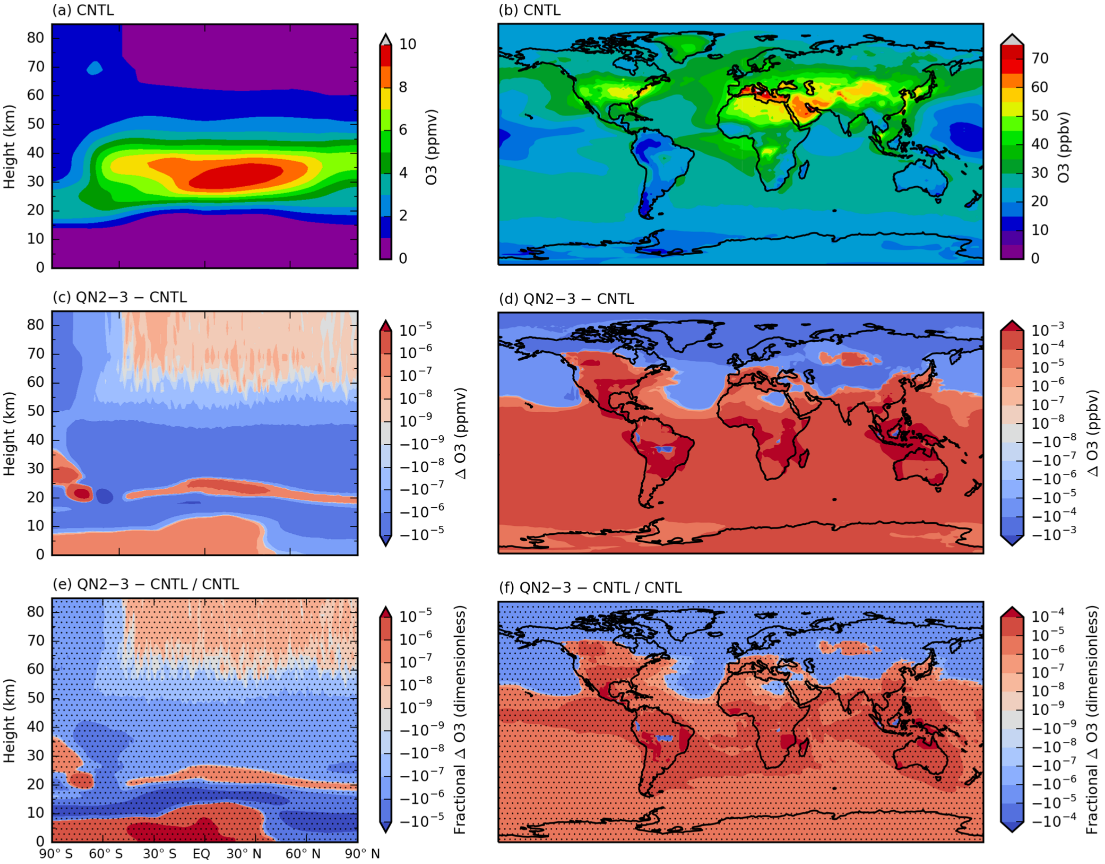

Figure 7. **(a, c, e)** Zonal-mean ozone from the last 10 years of the 20-year StratTrop 432-core simulations and **(b, d, f)** surface ozone from the last 10 years of the 20-year StratTrop 432-core simulations. **(a, b)** Ozone from the CNTL simulation. **(c, d)** Absolute differences between the QN2–3 simulation and the CNTL simulation. **(e, f)** Fractional differences between the QN2–3 simulation and the CNTL simulation. Stippling in the **(e)** and **(f)** plots indicates that the values are below the convergence criterion of the chemical solver (10^{-4}).

Table 7. Comparison of Newton–Raphson versus quasi-Newton methods by the metrics NMAD and NRMSD.

Chemistry	Species	Comparison	NMAD	NRMSD	NMB
StratTrop (432 cores)	OH	CNTL vs QN2–3	3.6986×10^{-8}	3.6019×10^{-8}	3.0382×10^{-9}
StratTrop (432 cores)	Ozone	CNTL vs QN2–3	8.8374×10^{-7}	8.9908×10^{-7}	7.3761×10^{-7}

coupled UKCA model. In practice, a 3 % reduction of costs for a large study involving 10 000 model years corresponds to 300 model years saved, roughly 100 real days of supercomputer time with the current setup.

We also demonstrated that the suggested method, while improving the performance, does not deteriorate the accuracy of physical predictions, which is an obvious requirement for any proposed method. From the cross comparisons under different computational environments (UKCA_BOX or parallel UM simulations), different chemical scenarios (interactive or noninteractive) for a large spectrum of chemical species (varying from very long lifetime or short lifetime), the method maintains the same level of accuracy as the original method.

Another feature of our approach is its flexibility to use with many existing chemistry solving systems. Whilst this work focussed specifically on the UKCA, the algorithm can be easily integrated to the existing codes of the other (unrelated) coupled chemical system solvers. If implemented in a chemical transport model, for example, one would expect the overall benefit to be greater, due to the greater proportion of computational expense of the chemical solver due to the lack of other online physical processes. As shown in Sect. 2, it is also simple to detach the algorithm from the modified program and revert back to the original algorithm if desired using options defined in the namelist. Furthermore, since the method is quite generic, it can be used beyond solving chemical systems. We think that it will be just as easy to implement

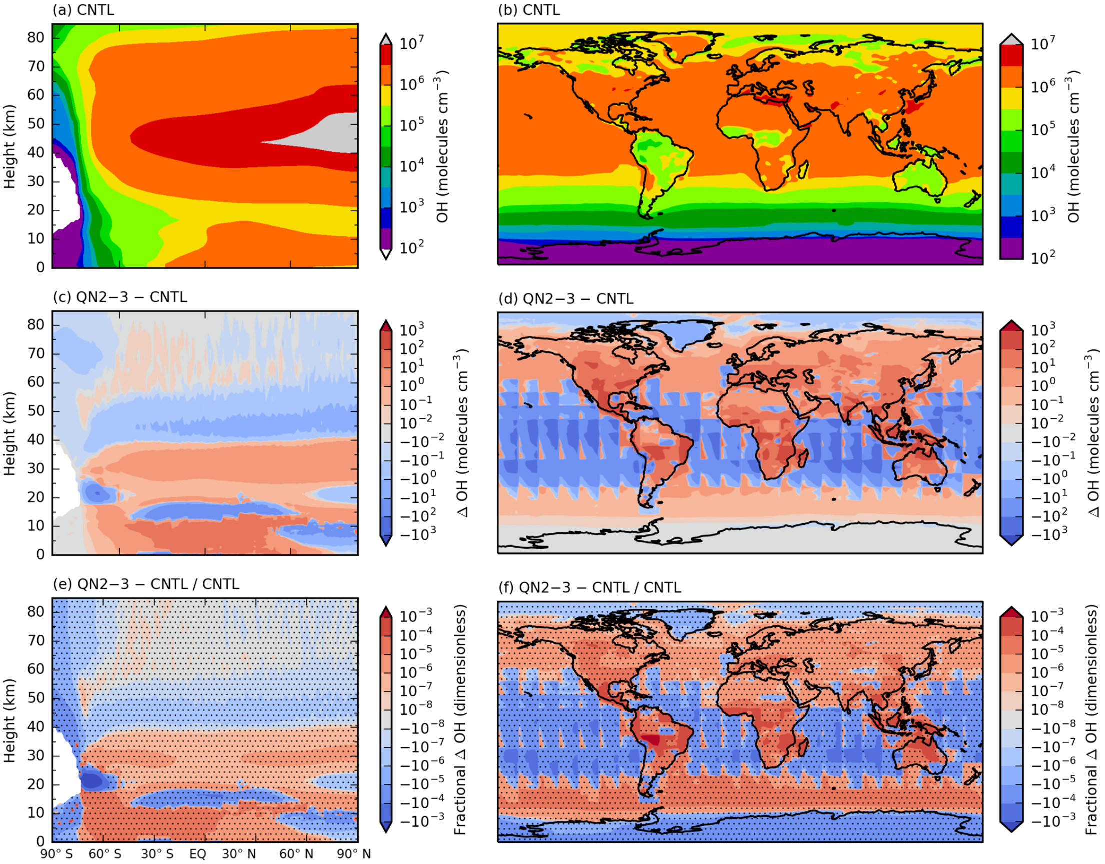

Figure 8. As Fig. 7 but for OH. Note the use of a log scale in the top (CNTL) plots. Note that the model domains are visible due to the extremely small differences in OH.

the method in other components of the climate model, for instance, solving systems of time-dependent nonlinear (partial) differential equations which can be cast into a problem of solving systems of nonlinear algebraic equations at each time step.

Finally, we remark that we have focused on one particular quasi-Newton approach which took advantage of available information and use it to replace costly Jacobian construction and linear system solving routines which proved to work robustly under fairly general conditions. There are also other Newton-type methods that avoid or reduce Jacobian construction (Brown and Saad, 1990). Although these methods pursue relatively different strategies (and hence require more substantial changes to a classical NR-type algorithm), it would be interesting to investigate their numerical capability.

Code availability. Due to intellectual property right restrictions, we cannot provide either the source code or documentation papers for the UM. However, we provide a pseudo-code for the NR and QN routine part of the DE system solver of the UKCA (see Appendix A below).

Obtaining the UM. The Met Office Unified Model is available for use under licence. The code is available in the UM trunk from version 10.8. Branches are also available in vn10.7 and vn10.6.1. A number of research organisations and national meteorological services use the UM in collaboration with the Met Office to undertake basic atmospheric process research, produce forecasts, develop the UM code and build and evaluate Earth system models. For further information on how to apply for a licence, see http://www.metoffice.gov.uk/research/modelling-systems/unified-model (Banzon et al., 2018; Cullen, 1993; Reynolds et al., 2007).

Appendix A: (Pseudo-code for NR+QN routine)

```
! Pseudo-code for solving the equation $F(c)=0$
! Inside the new chemistry step: determine the concentrations
for the next step...

...
...
$err = 10^{-4}$
...

Update tendencies ($\boldsymbol{f}\,(\boldsymbol{c}_*)$) at the time of the current chemistry step ($t_*$)

Make an initial guess for the algebraic system as an input to
the iterative solver $\boldsymbol{c} = \boldsymbol{c}_* + \boldsymbol{f}\,(\boldsymbol{c}_*)\; del\_t$

! Main NR Iteration loop starts

! Iteration counter: k, maximum iteration counter: max_iter

Do k=1,max_iter
  ! Update the $\boldsymbol{F}$ vector and store it
  $\mathbf{F} = \frac{\mathbf{c}-\boldsymbol{c}_*}{del\_t} - \boldsymbol{f}$
  $\boldsymbol{Fold} = \boldsymbol{F}$

  ! Jacobian construction and linear system solving
  Compute exact Jacobian $\mathbf{J}(\mathbf{c})$ of the $\boldsymbol{F}$ vector()

  Solve for the new increment del_c in the equation
  $\mathbf{J}\,(\mathbf{del\_}\boldsymbol{c}) = \mathbf{F}$
  $err\_c = \quad maxval(abs(\mathbf{del\_c})/\boldsymbol{c})$

  ! Updating the $\boldsymbol{c}$ values
  Perform treatments for troublesome convergence (e.g. $\beta$
  dampening factor) or
  Filtering of possible negative values in components of
  $\mathbf{c} = \mathbf{c} + \beta\,\mathbf{del\_c}$

  ! Test and decide if QN step will be taken
  ! This can be done on iterations $2 \le k \le 50$, and
  recommended on steps 2 & 3
  ! This step will not be done if the $\boldsymbol{c}$ vector converged
  and the routine is about to exit

  If ($err\_c \ge err$ .AND. $choice\_qn$) Then

    Update the tendencies
    Update the $\boldsymbol{F}$ vector
    $\mathbf{F} = \frac{\mathbf{c}-\boldsymbol{c}_*}{\mathrm{del\_t}} - \boldsymbol{f}$
    $\boldsymbol{delF} = \boldsymbol{F} - \boldsymbol{Fold}$

    ! QN approximation below ...

    Compute the Jacobian modification factor
    $a = \frac{DOT\_product(\mathbf{delF},\mathbf{F})}{DOT\_product(\mathbf{del\_c},\mathbf{del\_c})}$

    Re-solve for the newer increment $\boldsymbol{del\_c}$
    $\mathbf{J}\,(\mathbf{del\_c}) = (1-a)\mathbf{F}$

    Update $\boldsymbol{c}$ values
    $\mathbf{c} = \mathbf{c} + \beta\,\mathbf{del\_c}$

  End If

End Do
```

The Supplement related to this article is available online at https://doi.org/10.5194/gmd-11-3089-2018-supplement.

Author contributions. EE developed and implemented the method. NLA modernised the implementation and performed the global simulations in the article. SAN developed the modern BOX_MODEL with support from PTG, NLA, and ATA and performed the UKCA_BOX simulations in the article. CM developed the earlier versions of the box model. ATA and JAP oversaw the work. EE, NLA, and SAN wrote the paper with contributions from all authors.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank Oliver Wild for many useful discussions. The first author thanks Nigel Wood and Olaf Morgenstern for helpful comments. We also thank Alan Hewitt and Stuart Whitehouse for reviewing the code.

Model integrations have been performed using the ARCHER UK National Supercomputing Service and the MONSooN system, a collaborative facility supplied by the Joint Weather and Climate Research Programme, which is a strategic partnership between the UK Met Office and the Natural Environment Research Council. This work used the NEXCS HPC facility provided by the Natural Environment Research Council. We thank NCAS for providing support for the UKCA model development.

The first and last authors were supported under the ERC (ACCI) grant (project number 267760). Alex T. Archibald and Scott Nicholls thank the Isaac Newton Trust under whose auspices this work was funded.

Edited by: Jason Williams
Reviewed by: three anonymous referees

References

Banzon, V., Reynolds, R., and National Center for Atmospheric Research Staff (Eds.): The Climate Data Guide: SST data: NOAA High-resolution (0.25 × 0.25) Blended Analysis of Daily SST and Ice, OISSTv2, available at: https://climatedataguide.ucar.edu/climate-data/sst-data-noaa-high-resolution-025x025-blended-analysis-daily-sst-and-ice-oisstv2, last access: 23 June 2018.

Abraham, N. L., Archibald, A. T., Bellouin, N., Boucher, O., Braesicke, P., Bushell, A., Carslaw, K., Collins, B., Dalvi, M., Emmerson, K., Folberth, G., Haywood, J., Hewitt, A., Johnson, C., Kipling, Z., Macintyre, H., Mann, G., Telford, P., Merikanto, J., Morgenstern, O., O'Connor, F., Ordonez, C., Osprey, S., Pringle, K., Pyle, J., Rae, J., Reddington, C., Savage, N., Spracklen, D., Stier, P., West, R., Mulcahy, J., Woodward, S., Boutle, I., and Woodhouse, M. T.: Unified Model Documentation Paper 084: United Kingdom Chemistry and Aerosol (UKCA) Technical Description Met UM Version 10.6, available at: https://code.metoffice.gov.uk/doc/um/vn10.6/papers/umdp084.pdf (last access: 26 October 2016) (for the version of the model used here), 2012.

Atkinson, K.: Introduction to Numerical Analysis, John Wiley & Sons Inc., 1989.

Banerjee, A., Maycock, A. C., Archibald, A. T., Abraham, N. L., Telford, P., Braesicke, P., and Pyle, J. A.: Drivers of changes in stratospheric and tropospheric ozone between year 2000 and 2100, Atmos. Chem. Phys., 16, 2727–2746, https://doi.org/10.5194/acp-16-2727-2016, 2016.

Brandt, A.: Multilevel adaptive solutions to boundary value problems, Math. Comp., 31, 333–390, 1977.

Brown, P. N. and Saad, Y.: Hybrid Krylov methods for nonlinear system of equations, SIAM J. Sci. Stat. Comp., 11, 450–481, 1990.

Broyden, C. G.: A Class of Methods for Solving Nonlinear Simultaneous Equations, Mathematics of Computation, American Mathematical Society, 19, 577–593, 1965.

Cariolle, D., Moinat, P., Teyssèdre, H., Giraud, L., Josse, B., and Lefèvre, F.: ASIS v1.0: an adaptive solver for the simulation of atmospheric chemistry, Geosci. Model Dev., 10, 1467–1485, https://doi.org/10.5194/gmd-10-1467-2017, 2017.

Carver, G. D., Brown, P. D., and Wild, O.: The ASAD atmospheric chemistry integration package and chemical reaction database, Comput. Phys. Commun., 105, 197–215, 1997.

Chan, T. F. and Jackson, K. R.: Nonlinearly preconditioned Kylov subspace methods for discrete Newton algortihms, SIAM J. Sci. Stat. Comp., 5, 533–542, 1984.

Collins, W. J., Lamarque, J.-F., Schulz, M., Boucher, O., Eyring, V., Hegglin, M. I., Maycock, A., Myhre, G., Prather, M., Shindell, D., and Smith, S. J.: AerChemMIP: quantifying the effects of chemistry and aerosols in CMIP6, Geosci. Model Dev., 10, 585–607, https://doi.org/10.5194/gmd-10-585-2017, 2017.

Cullen, M. J. P.: The unified forecast/climate model, Meteorol. Mag., 122, 81–94, https://digital.nmla.metoffice.gov.uk/file/sdb%3AdigitalFile%7C33af6d4a-7c25-4bbc-819a-be64312e783d/, 1993.

Damian, V., Sandu, A., Damian, M., Potra, F., and Carmichael, G. R.: The kinetic preprocessor KPP – a software environment for solving chemical kinetics, Comput. Chem. Eng., 26, 1567–1579, 2002.

Davidon, W. C.: Variable metric method for minimization, SIAM J. Optimiz., 1, 1–17, 1991.

ECMWF: Documentation, available at: https://software.ecmwf.int/wiki/display/OIFS/Documentation, direct link: https://software.ecmwf.int/wiki/download/attachments/19661682/drhook.pdf (last access: 19 January 2018), 2013.

Fletcher, R.: A New Approach to Variable Metric Algorithms, Comput. J., 13, 317–322, 1970.

Glotfelty, T., He, J., and Zhang, Y.: Impact of future climate policy scenarios on air quality and aerosol-cloud interactions using an advanced version of CESM/CAM5: Part I. model evaluation for the current decadal simulations, Atmos. Environ., 152, 222–239, 2017.

Goldfarb, D.: A Family of Variable Metric Updates Derived by Variational Means, Math. Comput., 24, 23–26, https://doi.org/10.1090/S0025-5718-1970-0258249-6, 1970.

Heal, M. R., Heaviside, C., Doherty, R. M., Vieno, M., Stevenson, D. S., and Vardoulakis, S.: Health burdens of surface ozone in the UK for a range of future scenarios, Environ. Int., 61, 36–44, 2013.

Hewitt, H. T., Copsey, D., Culverwell, I. D., Harris, C. M., Hill, R. S. R., Keen, A. B., McLaren, A. J., and Hunke, E. C.: Design and implementation of the infrastructure of HadGEM3: the next-generation Met Office climate modelling system, Geosci. Model Dev., 4, 223–253, https://doi.org/10.5194/gmd-4-223-2011, 2011.

IPCC: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P. M., Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 1535 pp., 2013.

Kelley, C. T.: Iterative Methods for Linear and Nonliner Equations, SIAM, Philadelphia, 1995.

Knoll, D. A. and Keyes, D. E.: Newton-Krylov methods: A survey of approaches and applications, J. Comput. Phys., 193, 357–397, 2004.

Kvaalen, E.: A faster Broyden method, BIT Numerical Mathematics, SIAM, 31, 369–372, 1991.

Lamarque, J.-F., Shindell, D. T., Josse, B., Young, P. J., Cionni, I., Eyring, V., Bergmann, D., Cameron-Smith, P., Collins, W. J., Doherty, R., Dalsoren, S., Faluvegi, G., Folberth, G., Ghan, S. J., Horowitz, L. W., Lee, Y. H., MacKenzie, I. A., Nagashima, T., Naik, V., Plummer, D., Righi, M., Rumbold, S. T., Schulz, M., Skeie, R. B., Stevenson, D. S., Strode, S., Sudo, K., Szopa, S., Voulgarakis, A., and Zeng, G.: The Atmospheric Chemistry and Climate Model Intercomparison Project (ACCMIP): overview and description of models, simulations and climate diagnostics, Geosci. Model Dev., 6, 179–206, https://doi.org/10.5194/gmd-6-179-2013, 2013.

Lauritzen, P. H., Nair, R. D., and Ullrich, P.: A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid, J. Comput. Phys., 229, 1401–1424, 2009.

Lu, T., Chung, L., Yoo, C. S., and Chen, J. H.: Dynamic stiffness for direct numerical simulations, Combust. Flame, 156, 1542–1551, 2009.

Mann, G. W., Carslaw, K. S., Spracklen, D. V., Ridley, D. A., Manktelow, P. T., Chipperfield, M. P., Pickering, S. J., and Johnson, C. E.: Description and evaluation of GLOMAP-mode: a modal global aerosol microphysics model for the UKCA composition-climate model, Geosci. Model Dev., 3, 519–551, https://doi.org/10.5194/gmd-3-519-2010, 2010.

Mitsakou, C., Helmis, C., and Housiadas, C.: Eulerian modelling of lung deposition with sectional representation of aerosol dynamics, J. Aerosol Sci., 36, 75–94, 2005.

Monks, P. S., Archibald, A. T., Colette, A., Cooper, O., Coyle, M., Derwent, R., Fowler, D., Granier, C., Law, K. S., Mills, G. E., Stevenson, D. S., Tarasova, O., Thouret, V., von Schneidemesser, E., Sommariva, R., Wild, O., and Williams, M. L.: Tropospheric ozone and its precursors from the urban to the global scale from air quality to short-lived climate forcer, Atmos. Chem. Phys., 15, 8889–8973, https://doi.org/10.5194/acp-15-8889-2015, 2015.

Morgenstern, O., Braesicke, P., O'Connor, F. M., Bushell, A. C., Johnson, C. E., Osprey, S. M., and Pyle, J. A.: Evaluation of the new UKCA climate-composition model – Part 1: The stratosphere, Geosci. Model Dev., 2, 43–57, https://doi.org/10.5194/gmd-2-43-2009, 2009.

O'Connor, F. M., Johnson, C. E., Morgenstern, O., Abraham, N. L., Braesicke, P., Dalvi, M., Folberth, G. A., Sanderson, M. G., Telford, P. J., Voulgarakis, A., Young, P. J., Zeng, G., Collins, W. J., and Pyle, J. A.: Evaluation of the new UKCA climate-composition model – Part 2: The Troposphere, Geosci. Model Dev., 7, 41-91, https://doi.org/10.5194/gmd-7-41-2014, 2014.

Ortega, J. and Rheinboldt, W.: Iterative solutions of Nonlinear Equations in several variables, Acadmic Press, Boston, 1970.

Reynolds, R. W., Smith, T. M., Liu, C., Chelton, D. B., Casey, K. S., and Schlax, M. G.: Daily high-resolution blended analyses for sea surface temperature, J. Climate, 20, 5473–5496, https://doi.org/10.1175/2007JCLI1824.1, 2007.

Sandu, A., Verwer, J. G., Blom, J. G., Spee, E. J., Carmichael, G. R., and Potra, F. A., Benchmarking stiff ode solvers for atmospheric chemistry problems II: Rosenbrock solvers, Atmos. Environ., 31, 3469–3472, 1997.

Shanno, D. F.: Conditioning of quasi-Newton methods for function minimization, Math. Comput., 24, 647–656, 1970.

Telford, P. J., Abraham, N. L., Archibald, A. T., Braesicke, P., Dalvi, M., Morgenstern, O., O'Connor, F. M., Richards, N. A. D., and Pyle, J. A.: Implementation of the Fast-JX Photolysis scheme (v6.4) into the UKCA component of the MetUM chemistry-climate model (v7.3), Geosci. Model Dev., 6, 161–177, https://doi.org/10.5194/gmd-6-161-2013, 2013.

Tilmes, S., Lamarque, J.-F., Emmons, L. K., Kinnison, D. E., Ma, P.-L., Liu, X., Ghan, S., Bardeen, C., Arnold, S., Deeter, M., Vitt, F., Ryerson, T., Elkins, J. W., Moore, F., Spackman, J. R., and Val Martin, M.: Description and evaluation of tropospheric chemistry and aerosols in the Community Earth System Model (CESM1.2), Geosci. Model Dev., 8, 1395–1426, https://doi.org/10.5194/gmd-8-1395-2015, 2015.

Viallet, M., Goffrey, T., Baraffe, I., Folini, D., Geroux, C., Popov, M. V., Pratt, J., and Walder, R.: A Jacobian-free Newton-Krylov method for time-implicit multidimensional hydrodynamics-Physics-based preconditioning for sound waves and thermal diffusion, Astron. Astrophys., 586, 1–17, https://doi.org/10.1051/0004-6361/201527339, 2016.

Walters, D., Boutle, I., Brooks, M., Melvin, T., Stratton, R., Vosper, S., Wells, H., Williams, K., Wood, N., Allen, T., Bushell, A., Copsey, D., Earnshaw, P., Edwards, J., Gross, M., Hardiman, S., Harris, C., Heming, J., Klingaman, N., Levine, R., Manners, J., Martin, G., Milton, S., Mittermaier, M., Morcrette, C., Riddick, T., Roberts, M., Sanchez, C., Selwood, P., Stirling, A., Smith, C., Suri, D., Tennant, W., Vidale, P. L., Wilkinson, J., Willett, M., Woolnough, S., and Xavier, P.: The Met Office Unified Model Global Atmosphere 6.0/6.1 and JULES Global Land 6.0/6.1 configurations, Geosci. Model Dev., 10, 1487–1520, https://doi.org/10.5194/gmd-10-1487-2017, 2017.

Whitehouse, L. E., Tomlin, A. S., and Pilling, M. J.: Systematic reduction of complex tropospheric chemical mechanisms, Part I: sensitivity and time-scale analyses, Atmos. Chem. Phys., 4, 2025–2056, https://doi.org/10.5194/acp-4-2025-2004, 2004.

Wild, O. and Prather, M. J.: Excitation of the primary tropospheric chemical mode in a global three dimensional model, J. Geophys. Res.-Atmos., 105, 24647–24660, 2000.

Wild, O., Zhu, X. I. N., and Prather, M. J.: Fast-J: Accurate Simulation of In- and Below-Cloud Photolysis in Tropospheric Chemical Models, J. Atmos. Chem., 37, 245–282, 2000.

Yu, S., Eder, B., Dennis, R., Chu, S., and Schwartz, S. E.: New unbiased symmetric metrics for evaluation of air quality models, Atmos. Sci. Lett., 7, 26–34, 2006.