



Rapid Adaptive Optimization Model for Atmospheric Chemistry (ROMAC) v1.0

Jiangyong Li^{1,2}, Chunlin Zhang^{1,2}, Wenlong Zhao¹, Shijie Han^{1,2}, Yu Wang^{1,3}, Hao Wang^{1,2}, and Boguang Wang^{1,2}

¹Australia–China Centre for Air Quality Science and Management (Guangdong), Institute for Environmental and Climate Research, Jinan University, Guangzhou, 511443, China

²Guangdong Provincial Observation and Research Station for Atmospheric Environment and Carbon Neutrality in Nanling Forests, Guangzhou, 511443, China

³Air Quality Studies, Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China

Correspondence: Hao Wang (wanghao@jnu.edu.cn) and Boguang Wang (tbongue@jnu.edu.cn)

Received: 6 May 2023 – Discussion started: 5 July 2023

Revised: 21 September 2023 – Accepted: 22 September 2023 – Published: 30 October 2023

Abstract. The Rapid Adaptive Optimization Model for Atmospheric Chemistry (ROMAC) is a flexible and computationally efficient photochemical box model. Its unique adaptive dynamic optimization module allows for the dynamic and rapid estimation of the impact of chemical and physical processes on pollutant concentration. ROMAC outperforms traditional box models in evaluating the influence of physical processes on pollutant concentrations. Its ability to quantify the effects of chemical and physical processes on pollutant concentrations has been confirmed through chamber and field observation cases. Since the development of a variable-step and variable-order numerical solver that eliminates the need for Jacobian matrix processing, the computational efficiency of ROMAC has seen a marked improvement with only a marginal increase in error. Specifically, the computational efficiency has improved by 96 % when compared to several established box models, such as FOAM and AtChem. Moreover, the solver maintains a discrepancy of less than 0.1 % when its results are compared with those obtained from a high-precision solver in AtChem.

1 Introduction

Numerical models are effective tools of atmospheric chemistry studies. The 0-dimensional (0-D) box model has been widely used in previous studies to investigate the relationship between secondary pollutants and precursors (Decker et

al., 2021, 2019; Ling et al., 2017; Wang et al., 2017; He et al., 2019). The box model can be used as a ground Lagrangian trajectories model to study the influence of the regional transport of precursors on the formation of secondary pollutants (Cheng et al., 2010; Wang et al., 2019). In addition, the box model is also a powerful tool in environmental chamber studies (Chen et al., 2015; Novelli et al., 2018). Several box models have been developed and applied in previous studies, such as AtChem (Sommariva et al., 2020), Chemistry As A Box Model Application (CAABA/MECCA) (Sander et al., 2011, 2019), Framework for 0-D Atmospheric Modeling (FOAM) (Wolfe et al., 2016), PyCHAM (O’Meara et al., 2021), JIBox (Huang and Topping, 2021) and Photochemical Box Model incorporating the Master Chemical Mechanism (PBM-MCM) (Wang et al., 2018).

Since the processes of vertical and horizontal transmission are ignored, the simulation speed of the 0-D box model is higher than that of the 3-D air quality model. This allows box models to use more comprehensive chemical mechanisms and to focus on the analysis of chemical processes. For instance, the utilization of 0-D models in coupling with near-explicit chemical mechanisms can offer a comprehensive understanding of photochemical processes (Y. Wang et al., 2023). However, it is important to consider the impact of physical transport on long-lived species, such as its effect on O₃ concentration (Li et al., 2021; Liu et al., 2022). The 0-D model, which lacks a 3-D structure, is unable to directly estimate the impact of physical processes (e.g., vertical

and horizontal transport) on pollutant concentrations. Therefore, it is necessary to find a proper scheme to estimate the physical process for these models. Furthermore, as the atmospheric chemistry mechanism continues to develop, the number of chemical reactions involved gradually increases. Consequently, when using a chemical mechanism with massive reactions, the process of obtaining a chemical solution in a 0-D box model also becomes time-consuming. For example, the Master Chemical Mechanism (MCM v3.3.1) contains about 5900 species (Jenkin et al., 2015), and the size of the Jacobian matrix is close to 5900×5900 , which requires a large number of matrix calculations in the process of solving with the implicit solver. Therefore, it is necessary to develop a computationally efficient model for chemical mechanisms. Most of box models rely on third-party tools for differential equation solving. Several multistep or multistage approaches are commonly used by these chemical solvers, such as ROSENBROCK, BDF, LSODE, GEAR, SMVGEAR, etc. (Verwer et al., 1996; Aro, 1996a; Sandu et al., 1997a, b). Although these third-party solving tools have good accuracy and stability, the solving process requires a lot of computing resources, which significantly reduces the computational efficiency.

The simplified chemical mechanism can effectively improve the solution efficiency of chemical processes, such as SAPRC07 (Carter, 2010), CB6 (Yarwood, 2010), MOZART (Emmons et al., 2010) and the Mainz Organic Mechanism (MOM) (Sander et al., 2019). The MCM mechanism also has a simplified version (<https://mcm.york.ac.uk/CRI>, last access: 21 October 2023), which can improve the computational efficiency. General methods for reduction (Young and Boris (1977); Djouad and Sportisse, 2002) and their on-line implementations (Sander et al., 2019; Shen et al., 2022; Lin et al., 2023) have been developed. These simplified mechanisms generally have a focus on getting particular parts of chemistry. As a result, the simulation results for certain species may diverge from those obtained using near-explicit chemical mechanisms, particularly concerning radicals (e.g., OH, HO₂, RO₂) and the concentrations of secondary pollutants (Ying and Li, 2011; Jimenez, 2003). The adoption of near-explicit chemical mechanisms enables a more detailed representation of the intricate process of photochemical reactions. Consequently, the simplified mechanism cannot adequately replace the role of the near-explicit mechanism. To improve efficiency, another approach is to improve the computational efficiency of differential equation solver programs, such as using GPU acceleration (Alvanos and Christoudias, 2017) or using the quasi-Newton method (Esentürk et al., 2018). These methods can effectively shorten the running time of the program, but still need to consume a lot of memory and CPU (or GPU) resources when processing the Jacobian matrix. There are alternative solution methods that do not need to store and update the Jacobian matrix, such as quasi-steady-state approximation (QSSA), multistep explicit and semi-implicit methods (Mott et al., 2000; Young

and Boris, 1977). But these methods usually do not conserve mass (Cariolle et al., 2017). In addition, there are also fully implicit methods that do not need to deal with the Jacobian matrix, such as the Euler backward iterative (EBI) method (Hertel et al., 1993). However, the EBI solver has a large truncation error because it is only first-order accurate. Another stiff ordinary differential equations (ODEs) preconditioner method based on Newton linearization also simplifies the matrix operations during the solution (Aro, 1996a). But these algorithms may fail to converge when the Jacobian matrix is significantly off-diagonally dominant (Aro, 1996b). Hence, with the increasing complexity and scale of chemical mechanism systems, it is still a challenge to make these solving algorithms converge stably.

The Rapid Adaptive Optimization Model for Atmospheric Chemistry (ROMAC) is a computationally efficient photochemical box model. To enhance its computational efficiency, a variable-step and variable-order (VSVOR) solver without Jacobian matrix processing was developed for ROMAC. This distinctive solver offers superior computational efficiency in handling atmospheric chemical mechanisms by eliminating the need for third-party libraries for numerical solving. By utilizing the VSVOR solver, ROMAC provides users with the flexibility to dynamically optimize the influence of physical processes on pollutant concentration, which is difficult to achieve in the traditional box model with oversimplified physical modules. Therefore, ROMAC will be computationally efficient and outperform the traditional box models in evaluating the impact of physical processes on pollutant concentrations.

2 Description of ROMAC

ROMAC is a 0-D model focused on the simulation of atmospheric chemical kinetics problems. It was developed to provide users with a flexible and efficient computational tool. The core modules of ROMAC were developed in Fortran, and the data pre-processing and post-processing modules were developed in Python, which can keep the model running efficiently and provide users with flexible processing tools. In ROMAC, the changes in concentration of a species can mathematically be represented as Eq. (1):

$$\frac{dc}{dt} = \left[\frac{dc}{dt} \right]_{\text{chem}} + \left[\frac{dc}{dt} \right]_{\text{emis}} + \left[\frac{dc}{dt} \right]_{\text{dry}} + \left[\frac{dc}{dt} \right]_{\text{dilu}} + \left[\frac{dc}{dt} \right]_{\text{others}}, \quad (1)$$

where $\left[\frac{dc}{dt} \right]_{\text{chem}}$ represents the changes due to chemical reactions; $\left[\frac{dc}{dt} \right]_{\text{emis}}$ represents the emission rate for the species; $\left[\frac{dc}{dt} \right]_{\text{dry}}$ and $\left[\frac{dc}{dt} \right]_{\text{dilu}}$ represent the dry deposition and dilution, respectively. For dry deposition, ROMAC uses the maximum dry deposition velocity (cm s^{-1}) calculated by Zhang

et al. (2003) to estimate the dry deposition process of the species, and users can also customize this value. The dry deposition process is added to the model in the form of first-order kinetics, and the kinetic constant is calculated by the dry deposition velocity and the preset boundary layer height (cm). Similar to other models (Wolfe et al., 2016; Sommariva et al., 2020), ROMAC uses first-order kinetics to calculate the dilution process, and users can customize the constants of the dilution process. Note that the current version of ROMAC does not feature a dedicated input function for wet deposition. Instead, ROMAC allows users to set a custom rate term, $\left[\frac{dc}{dt}\right]_{\text{others}}$, which can be employed to account for wet deposition. If wet deposition is important for the simulation case, especially concerning the chemical mechanism of hydrophilic components like sulfate, it is suggested that the user incorporates it into $\left[\frac{dc}{dt}\right]_{\text{others}}$. Moreover, users have the flexibility to add additional change rates as needed, such as the gas-wall partitioning in the chamber studies or the external transport (e.g., vertical and horizontal transport) in field observations. It is not a difficult task to incorporate new rates of change into $\left[\frac{dc}{dt}\right]_{\text{others}}$ within the ROMAC framework.

The subsequent sections offer a comprehensive overview of the ROMAC features. Furthermore, to facilitate reference, all the parameters employed in this paper are catalogued in Appendix B.

2.1 High-efficiency solver for atmospheric chemical kinetic equations

Unlike many existing models, ROMAC distinguishes itself by not relying on third-party libraries for numerical solving. Instead, ROMAC employs its own computationally efficient variable-step and variable-order numerical solver named “VSVOR.” This solver is engineered to enhance computational efficiency while accommodating the universal attributes of atmospheric chemical mechanisms. It approaches all differential equations uniformly, eliminating the need for customized solution schemes tailored to specific chemical mechanisms. Therefore, the VSVOR solver offers a universal and versatile method for chemical solving. The VSVOR solver has a control on the truncation error of integration according to the relative tolerance (*rtol*) and the absolute tolerance (*atol*) specified by the user. The proposed solver offers an algorithm that strikes a balance between efficiency and accuracy. Most of the time, the accuracy of the VSVOR solver can be second order.

The chemical mechanism is the core of atmospheric chemical box models. Generally, chemical reaction equations can be described in Eq. (2):

$$\alpha_1 r_1 + \alpha_2 r_2 + \dots + \alpha_n r_n \rightarrow \beta_1 p_1 + \beta_2 p_2 + \dots + \beta_m p_m, \quad (2)$$

where α and β represent the stoichiometric number, and r and p represent the reactant and product, respectively.

Hence, the derivative of species concentration with respect to time can be described as an ODEs system shown in Eq. (3). For species i , f_i can be calculated by Eq. (4). In Eq. (4), $P_{i,t}$ and $L_{i,t}$ denote the chemical generation rate and the loss rate of species i at time t , respectively. It is worth noting that the loss rate is related to the concentration of species i . Therefore, to facilitate the subsequent formula derivation, $L_{i,t}$ can be described as a multivariate higher-degree equation for the concentration of species i shown in Eq. (5), where R_{tot} represents the total number of the reactions related to the loss rate of species i ; α is the stoichiometric number, and $l_{i,t,R}$ is the part of the chemical reaction rate that is not directly related to the concentration of species i . The computation of the $f(C_t, t)$ follows the approach in the Fortran code provided on the MCM official website (<https://mcm.york.ac.uk/MCM/about/archive>, last access: 21 October 2023).

$$\left[\frac{dC_t}{dt}\right]_{\text{chem}} = f(C_t t) \quad (3)$$

$$f_i(C_{i,t}, t) = P_{i,t} - L_{i,t} \quad (4)$$

$$L_{i,t} = \sum_{R=1}^{R_{\text{tot}}} l_{i,t,R} C_{i,t}^{\alpha_{i,R}} \quad (5)$$

The lifetime of different species in atmospheric chemical mechanisms varies greatly. For example, OH has an atmospheric lifetime of only seconds but O₃ has a lifetime of several days. Therefore, the ODEs system of atmospheric chemical kinetics simulation is extremely stiff, and explicit methods (e.g., explicit Euler method, explicit Runge–Kutta method) cannot achieve a stable solution without using a time step shorter than all lifetimes in the system, which is computationally unfeasible.

In ROMAC, the implicit Euler method and the trapezoidal method were used to solve the ODEs; the iteration formula is given in Eqs. (6) and (7). The implicit Euler method, renowned for its exceptional numerical stability, has found extensive application in other atmospheric chemistry models (Esentürk et al., 2018). However, because the implicit Euler method only has first-order accuracy, it may introduce large truncation errors in the process of integration. Hence, the trapezoidal method iteration formula shown in Eq. (7) is used for integration in a specific situation. Both the implicit Euler method and the trapezoidal method have the term of $f(C_{t+1}, t+1)$ which is unknown at time t and needs to be solved. The Newton–Raphson (NR) scheme is a widely used method for solving implicit equations in both the implicit Euler method and the trapezoidal method. Equations (6) and (7) can be expressed in the form of Eqs. (8) and (9), respectively.

$$C_{t+1} = C_t + f(C_{t+1}, t+1) \Delta t \quad (6)$$

$$C_{t+1} = C_t + \frac{f(C_t, t) + f(C_{t+1}, t+1)}{2} \Delta t \quad (7)$$

$$g_1(C_{t+1}) = C_{t+1} - C_t - f(C_{t+1}, t+1) \Delta t = 0 \quad (8)$$

$$g_2(C_{t+1}) = C_{t+1} - C_t - \frac{f(C_t, t) + f(C_{t+1}, t + 1)}{2} \Delta t = 0 \quad (9)$$

Thus, the iteration formula of the NR scheme can be expressed in the form of Eq. (10). Where $\nabla g^{-1}(C_{t+1})$ is the inverse matrix of the Jacobian matrix of $g(C_{t+1})$. The Jacobian matrix for the implicit Euler method is given in Eq. (11), and the Jacobian matrix for the trapezoidal method is given in Eq. (12). It should be noted that the size of the Jacobian matrix and its inverse matrix will increase with the number of species in the chemical mechanisms increasing. In particular, dealing with near-explicit chemical mechanisms (e.g., MCM) would consume a lot of computer resources to store the Jacobian matrix and its inverse matrix. In addition, the inverse of a large-scale Jacobian matrix is quite time-consuming.

$$C_{t+1}^{k+1} = C_{t+1}^k - \nabla g^{-1}(C_{t+1})g(C_{t+1}) \quad (10)$$

$$\nabla g_1(C_{t+1}) = \begin{bmatrix} 1 - \frac{\partial f_1(C_{1,t+1})}{\partial C_{1,t+1}} \Delta t & \cdots & -\frac{\partial f_1(C_{1,t+1})}{\partial C_{n,t+1}} \Delta t \\ \vdots & \ddots & \vdots \\ -\frac{\partial f_n(C_{n,t+1})}{\partial C_{1,t+1}} \Delta t & \cdots & 1 - \frac{\partial f_n(C_{n,t+1})}{\partial C_{n,t+1}} \Delta t \end{bmatrix} \quad (11)$$

$$\nabla g_2(C_{t+1}) = \begin{bmatrix} 1 - \frac{\partial f_1(C_{1,t+1})}{\partial C_{1,t+1}} \Delta t & \cdots & -\frac{\partial f_1(C_{1,t+1})}{\partial C_{n,t+1}} \Delta t \\ -\frac{\partial L_{1,t+1}}{\partial C_{1,t+1}} \frac{\Delta t}{2} & \cdots & +\frac{\partial P_{1,t+1}}{\partial C_{n,t+1}} \frac{\Delta t}{2} - \frac{\partial L_{1,t+1}}{\partial C_{n,t+1}} \frac{\Delta t}{2} \\ \vdots & \ddots & \vdots \\ -\frac{\partial f_n(C_{n,t+1})}{\partial C_{1,t+1}} \Delta t & \cdots & 1 - \frac{\partial f_n(C_{n,t+1})}{\partial C_{n,t+1}} \Delta t \\ +\frac{\partial P_{n,t+1}}{\partial C_{1,t+1}} \frac{\Delta t}{2} - \frac{\partial L_{n,t+1}}{\partial C_{1,t+1}} \frac{\Delta t}{2} & \cdots & -\frac{\partial L_{n,t+1}}{\partial C_{n,t+1}} \frac{\Delta t}{2} \end{bmatrix} \quad (12)$$

A simplified Newton (SN) method can effectively reduce the computational complexity of the iterative process of the NR method. The traditional SN method substitutes the inverse Jacobian matrix obtained in the first iteration for the inverse matrix in the subsequent iterations. Although the traditional SN method can reduce the amount of computation, it still needs to calculate and store the inverse of the Jacobian matrix at each time step. To further improve the computational efficiency, ROMAC uses a diagonal simplified Newton (DSN) method to solve the implicit equations.

When the Δt in Eqs. (11) and (12) is small enough, the Jacobian matrix of $g(C_{t+1})$ will be a diagonally dominant matrix or a quasi-diagonally dominant matrix. The aforementioned characteristics are inherently present within the Jacobian matrix of chemical mechanisms and are impervious to variations in specific chemical mechanisms. As a result, this scheme proves to be universally applicable across different chemical mechanisms. Under these conditions, the inverse matrix of Jacobian in Eq. (11) can be approximated by Eq. (13). According to the equations associated with the implicit Euler method in Eqs. (1)–(13), the iteration formula for species i is shown in Eq. (14), where k represents the number of iterative solutions. A previous study has also shown that such approximations are reliable (Aro, 1996b). Similarly, the

approximate inverse of the Jacobian matrix for the trapezoidal method and the iterative formulas for the solution can be derived as shown in Eqs. (15) and (16), respectively. The solution process was iterated until the difference between the results of two iterations was less than one-tenth of the preset truncation error tolerance (etc., $0.1 \times \text{atol}$ or $0.1 \times \text{rtol}$) for ODEs solution.

$$\nabla g_1^{-1}(C_{t+1}) \approx \begin{bmatrix} \frac{1}{1 - \frac{\partial f_1(C_{1,t+1})}{\partial C_{1,t+1}} \Delta t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{1 - \frac{\partial f_n(C_{n,t+1})}{\partial C_{n,t+1}} \Delta t} \end{bmatrix} \quad (13)$$

$$C_{i,t+1}^{k+1} = \frac{\sum_{R=1}^{R_{\text{tot}}} (\alpha_{i,R} - 1) l_{t+1,R} C_{i,t+1}^{k\alpha_{i,R}} \Delta t + C_{i,t} + P_{t+1} \Delta t}{1 + \sum_{R=1}^{R_{\text{tot}}} \alpha_{i,R} l_{t+1,R} C_{i,t+1}^{k\alpha_{i,R}-1} \Delta t} \quad (14)$$

$$\nabla g_2^{-1}(C_{t+1}) \approx \begin{bmatrix} \frac{1}{1 - \frac{\partial f_1(C_{1,t+1})}{\partial C_{1,t+1}} \Delta t - \frac{\partial L_{1,t+1}}{\partial C_{1,t+1}} \frac{\Delta t}{2}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{1 - \frac{\partial f_n(C_{n,t+1})}{\partial C_{n,t+1}} \Delta t - \frac{\partial L_{n,t+1}}{\partial C_{n,t+1}} \frac{\Delta t}{2}} \end{bmatrix} \quad (15)$$

$$C_{i,t+1}^{k+1} = \frac{\sum_{R=1}^{R_{\text{tot}}} (\alpha_{i,R} - 1) l_{t+1,R} C_{i,t+1}^{k\alpha_{i,R}} \Delta t + 2C_{i,t} + P_{i,t} \Delta t - L_{i,t} \Delta t + P_{i,t+1} \Delta t}{2 + \sum_{R=1}^{R_{\text{tot}}} \alpha_{i,t+1,R} C_{i,t+1}^{k\alpha_{i,R}-1} \Delta t} \quad (16)$$

It is worth noting that if all of the stoichiometric number (α_R) is equal to 1, Eq. (14) is the same as the iteration formula of the EBI solver (Hertel et al., 1993) used in the CMAQ model. In this study, Eq. (14) provides a generalized form of the EBI iteration formula. Hertel et al.'s (1993) study shows that the EBI solver has the advantages of high computational efficiency and high accuracy. However, the convergence condition of this method has not been discussed, such as how to choose the optimal integration time step size to make the solution process stable and convergent. If the time step size is too short, the computational efficiency will decrease. However, if the time step size is too large, the Jacobian matrix will not be diagonally dominant, it will lead to an algorithm hard to converge or even not to converge. This problem also exists in the EBI algorithm. Especially for such a complex chemical mechanism as the MCM, directly using the EBI scheme involves a large risk of causing the algorithm not to converge. In ROMAC, the variable time step and variable order scheme help to balance the computational efficiency and accuracy, maintain the Jacobian matrix as a quasi-diagonally dominant matrix and reduce the risk of convergence failure. Hence, this scheme will enhance the applicability and stability of the ROMAC numerical solver compared to the EBI numerical solver.

Actually, it is difficult to use a fixed time step to ensure that the Jacobian matrix is always quasi-diagonally dominant. In order to find the optimal time step, a variable time

step size scheme is used in our model. First, Δt_0 is defined as an extremely small positive value to ensure that this value is not less than the rounding error of the computer. According to IEEE Std 754–2008 (IEEE, 2008), Δt_0 is defined as 2.22×10^{-16} s in ROMAC. Secondly, Δt_1 is defined as atmospheric lifetime of the species with the shortest lifetime in the chemical mechanism, as shown in Eq. (17). Third, a strict diagonal dominance matrix requires that the diagonal elements are greater than the sum of the rest of the elements in the same row, as shown in Eq. (18). Hence, $\Delta t_{2,i}$ is calculated by Eq. (19) to ensure that Eq. (18) holds, and Δt_2 is the minimum in the set of $\Delta t_{2,i}$ shown in Eq. (20), where i represents the rows of the Jacobian matrix. Finally, the initial integration time step size is determined by Eq. (21).

$$\Delta t_1 = \left[\frac{1}{L_t} \right]_{\min} \quad (17)$$

$$|\nabla g(C_{t+1})_{i,i}| > \sum_{j=1}^n |\nabla g(C_{t+1})_{i,j}| \quad (18)$$

$$\Delta t_{2,i} = \frac{0.9}{\left(\sum_{j=1}^n \left| \frac{\partial f_i(C_{1,t+1})}{\partial C_{j,t+1}} \right| \right)} \quad (19)$$

$$\Delta t_2 = [\Delta t_{2,i}]_{\min} \quad (20)$$

$$\Delta t_{\text{init}} = \begin{cases} \Delta t_0 & (\Delta t_0 \geq \Delta t_1 \text{ and } \Delta t_0 \geq \Delta t_2) \\ \Delta t_1 & (\Delta t_0 < \Delta t_1 \text{ and } \Delta t_0 < \Delta t_2 \text{ and } \Delta t_2 \geq \Delta t_1) \\ \Delta t_2 & (\Delta t_0 < \Delta t_1 \text{ and } \Delta t_0 < \Delta t_2 \text{ and } \Delta t_2 < \Delta t_1) \end{cases} \quad (21)$$

In order to improve the computational efficiency, the integration time step size should grow while ensuring the accuracy of the solution. When the time step size grows, the local truncation error (LTE) should be controlled. In each step (Δt), ROMAC uses both single-step and double-step methods for integration, and the calculated results are recorded as $C_{\Delta t}$ and $C_{\frac{\Delta t}{2}}$, respectively. LTE is estimated by the difference between $C_{\Delta t}$ and $C_{\frac{\Delta t}{2}}$ ($\text{LTE} = |C_{\frac{\Delta t}{2}} - C_{\Delta t}|$), and the relative error is estimated by Eq. (22). This method has been successfully used in a previous study (Aro, 1996a).

$$\text{RERR} = \left[\frac{|C_{\frac{\Delta t}{2}} - C_{\Delta t}|}{1 + C_{\Delta t}} \right]_{\min} \quad (22)$$

The model needs to adjust the integration time step according to the tolerance preset by the user. This requires inferencing a maximum integration time step based on the preset tolerance. According to the Lagrange remainder of the Taylor formula, the RERR of the integration result can also be expressed as Eq. (23), where s is the order of integration accuracy, equal to 1 for the implicit Euler method and equal to 2 for the trapezoidal method. Similarly, the user-specified maximum integral relative error can be expressed as Eq. (24), where Δt_{max} is an estimate of the maximum step size allowed when the preset rtol condition is satisfied. In ROMAC, the values of ξ_1 and ξ_2 in Eqs. (23) and (24) are assumed to be approximate ($\xi_1 \approx \xi_2$). According to Eqs. (23) and (24), the maximum integration time step can be estimated by Eq. (25). Finally, the

integration time step is updated according to Eqs. (26) and (27) to make sure that the time step is not larger than the maximum time step. In order to avoid a too accurate result making the integration time step size grow too large, when Δt_{opt} is greater than Δt by a factor of 10, the time step is only increased by a factor of 10. In general, as the integration step size increases, the number of iterations (N) required by the solver in this study will also increase. Too many iterations will make the computation time-consuming, and thus the integration time step is not increased when the solver iteration time exceeds 50 ($N \geq 50$).

$$\text{RERR} = \frac{R_n(\Delta t)}{1 + C_{\Delta t}} = \frac{f^{(s+1)}(\xi_1)\Delta t^{s+1}}{(1 + C_{\Delta t}) \times (s + 1)!} \quad (23)$$

$$\text{rtol} = \frac{R_n(\Delta t_{\text{max}})}{1 + C_{\Delta t}} = \frac{f^{(s+1)}(\xi_2)\Delta t_{\text{max}}^{s+1}}{(1 + C_{\Delta t}) \times (s + 1)!} \quad (24)$$

$$\Delta t_{\text{max}} = \left(\frac{\text{rtol}}{\text{RERR}} \right)^{\frac{1}{s+1}} \Delta t \quad (25)$$

$$\Delta t_{\text{opt}} = 0.9 \Delta t_{\text{max}} \quad (26)$$

$$\Delta t_{t+1} = \begin{cases} \Delta t_{\text{opt}} & (\Delta t_{\text{opt}} < 10\Delta t_t \text{ and } N < 50) \\ 10\Delta t_t & (\Delta t_{\text{opt}} \geq 10\Delta t_t \text{ and } N < 50) \\ \Delta t_t & (N \geq 50) \end{cases} \quad (27)$$

If $\text{RERR} < \text{rtol}$ or $\text{LTE} < \text{atol}$, proceed to the next integration time, otherwise the integration time step is halved and re-integrated until the tolerance requirement is satisfied.

Another important question is whether to choose the implicit Euler method or the trapezoidal method for the integration process. Both the implicit Euler method and the trapezoidal method are stable for stiff ODEs. However, the solution method used in this study requires the Jacobian matrix to be diagonally dominant or quasi-diagonally dominant. Since the initial time step in Eqs. (18) and (19) are derived from the implicit Euler method, the implicit Euler method is used for integral starting. If the algorithm converges quickly ($N < 50$), then the trapezoidal method is used in the next integration time step. When N is greater than 50, the algorithm is switched to the implicit Euler method in the next integration time step to improve the computational efficiency.

2.2 Adaptive dynamic optimization module and variables constraints

ROMAC can be run under user-specified variable constraints, including but not limited to concentrations of chemical species, photolysis rate, temperature, humidity, pressure and other meteorological conditions. For concentrations of chemical species, ROMAC provides the user with three different constraint schemes.

Scheme 1. Different from previous models, ROMAC provides a novel constraint scheme to use the observed data to constrain the model run. Scheme 1 does not directly input the species concentration, but controls the $\left[\frac{dc}{dt} \right]_{\text{others}}$ term with adaptive dynamic optimization algorithm. The default value

for $[\frac{dc}{dt}]_{\text{others}}$ is 0, after integration, $\Delta[\frac{dc}{dt}]_{\text{others}}$ can be estimated by the gap between the observed and simulated values, as detailed in Eq. (28):

$$\Delta\left[\frac{dc}{dt}\right]_{\text{others}} = \begin{cases} \frac{C_{\text{obs},n+1} - C_{\text{model},n+1}}{t_{n+1} - t_n} \\ (|C_{\text{obs},n+1} - C_{\text{model},n+1}| \leq 0.1 \\ \times |C_{\text{model},n+1}|) \\ \frac{0.1 \times C_{\text{model},n+1}}{t_{n+1} - t_n} \times (-1)^u \\ (|C_{\text{obs},n+1} - C_{\text{model},n+1}| > 0.1 \\ \times |C_{\text{model},n+1}|) \end{cases}, \quad (28)$$

where C_{obs} represents the observations and C_{model} represents the simulations. In Eq. (28), $u = 1$ when $C_{\text{obs},n+1}$ is less than $C_{\text{model},n+1}$ and $u = 2$ when $C_{\text{obs},n+1}$ is greater than $C_{\text{model},n+1}$. In complex systems, changing $[\frac{dc}{dt}]_{\text{others}}$ may also affect other chemical process, and thus the relationship between $[\frac{dc}{dt}]_{\text{others}}$ and the simulation results may be nonlinear. Therefore, it is difficult to calculate $[\frac{dc}{dt}]_{\text{others}}$ in a single iteration. It is necessary to estimate this by loop iteration until the difference between observation and simulation reaches a preset tolerance. In this study, the difference between observation and simulation is characterized by the root mean square error (RMSE) shown in Eq. (29).

$$\text{RMSE} = \sqrt{(C_{\text{obs},n+1} - C_{\text{model},n+1})^2} \quad (29)$$

The cyclic dynamic optimization process of $[\frac{dc}{dt}]_{\text{others}}$ is shown in Fig. 1. The iterative updating formula of $[\frac{dc}{dt}]_{\text{others}}$ based on the Newton–Raphson method is given in Eq. (30).

$$\left[\frac{dc}{dt}\right]_{\text{others}_{m+1}} = \begin{cases} \left[\frac{dc}{dt}\right]_{\text{others}_m} + \Delta\left[\frac{dc}{dt}\right]_{\text{others}} & (m = 1) \\ \left[\frac{dc}{dt}\right]_{\text{others}_m} - \text{RMSE}_m \cdot \left[\frac{d\text{RMSE}}{d\left[\frac{dc}{dt}\right]_{\text{others}}}\right]_m^{-1} & (m > 1) \end{cases} \quad (30)$$

$$\left[\frac{d\text{RMSE}}{d\left[\frac{dc}{dt}\right]_{\text{others}}}\right]_m = \frac{\left[\frac{\Delta\text{RMSE}}{\Delta\left[\frac{dc}{dt}\right]_{\text{others}}}\right]_m}{\frac{\text{RMSE}_m - \text{RMSE}_{m-1}}{\left[\frac{dc}{dt}\right]_{\text{others}_m} - \left[\frac{dc}{dt}\right]_{\text{others}_{m-1}}}}, \quad (31)$$

where m is the number of iterations and $\Delta[\frac{dc}{dt}]_{\text{others}}$ at the first iteration ($m = 1$) is estimated by Eq. (28). When the number of iterations is greater than 1 ($m > 1$), the update equation of $\Delta[\frac{dc}{dt}]_{\text{others}}$ is developed based on the Newton–Raphson method. The RMSE is used as the objective function for optimization, and the derivative of RMSE with $[\frac{dc}{dt}]_{\text{others}}$ is estimated by the difference method shown in Eq. (31).

The $[\frac{dc}{dt}]_{\text{others}}$ can also be optimized in the mode of kinetic equations (e.g., $[\frac{dc}{dt}]_{\text{others}} = k_{\text{others}} \times C$), and then the kinetic constants (k_{others}) can be optimized using a similar process

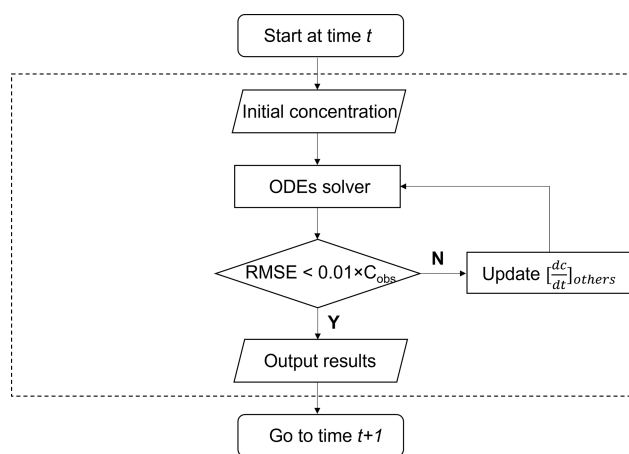


Figure 1. The cyclic dynamic optimization process of $[\frac{dc}{dt}]_{\text{others}}$.

shown in Fig. 1. ROMAC provides an option for the user to switch between these two modes. Furthermore, the user can also use other algorithms for dynamic optimization, such as ensemble Kalman filter (EnKF).

Scheme 2. In Scheme 2, the concentration of species can be initialized at the beginning of each simulation time step, which is mainly applied to the solution of initial value problems and more suitable for chamber simulation. This scheme has been widely used in previous models (e.g., PBM-MCM, AtChem, F0AM). However, if the regional transport process of pollutants is not considered, the simulation results of long-lived species in this scheme may have large deviations from the observed results.

Scheme 3. Scheme 3 constrains the change rate of species concentration ($\frac{dc}{dt} = 0$) while constraining the initial concentration, in a similar way as in F0AM. The advantage of this scheme is that the constrained variables can be kept at a user-specified level throughout the simulation. In this scheme, the long-lived species can maintain the observed concentration level. This constraining is appropriate if the temporal resolution of the observed data is high. The time interval of the model should be significantly smaller than the lifetime of constrained species. However, this approach also has its limitations. Since species concentrations are constrained as a constant, chemical imbalances may result.

To better understand the distinctive attributes of various constraint schemes, we conducted a straightforward test case in which nitric oxide (NO) was constrained by three different schemes. Other input species (i.e., VOCs, NO₂, O₃, SO₂, CO) were constrained by Scheme 3. Figure 2 illustrates the results of this test, showcasing how different schemes affect the concentration of target species. The simulation utilized a time step of 120 s, while the input data interval from observations was set at 3600 s. Under Scheme 2, where emissions and regional transport were not considered, NO concentrations experienced a rapid decline before reaching a steady state (Fig. 2a). By contrast, both Scheme 1 and Scheme 3

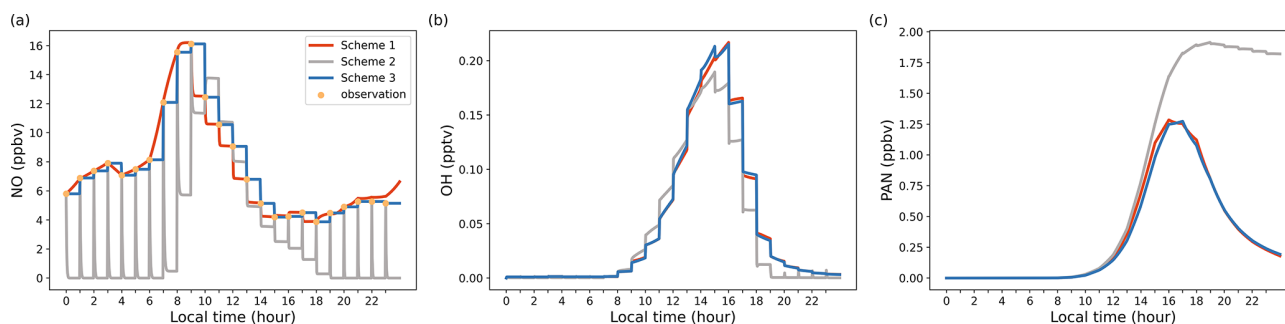


Figure 2. Model output results illustrating diurnal variations for selected species, highlighting the impact of different concentration constraint schemes. **(a)** NO concentrations; **(b)** OH concentrations; **(c)** PAN concentrations.

displayed NO concentrations in close agreement with the observed hourly averages, ensuring that the model faithfully replicated the real atmospheric conditions. It is worth noting that due to variations in constraint schemes, simulated concentrations of other unconstrained species, such as OH and PAN, can also diverge (Fig. 2b and c). This case study was primarily designed to elucidate the unique features of different constraint schemes, with no intent to definitively validate or invalidate any particular scheme. Users are encouraged to make their scheme selections judiciously, aligning them with their specific research needs and observational findings.

2.3 Photolysis

ROMAC provides two ways for the user to set the photolysis rate. First, the user can specify the photolysis rate at each integration time step in the form of an ASCII file. The input photolysis rate can be estimated by other models or the observations. In ROMAC, a Python script (*TUV2ROMAC.py*) is provided for coupling the output of the Tropospheric Ultraviolet and Visible Radiation model (TUVv5.2, available at <https://www2.acom.ucar.edu/modeling/tropospheric-ultraviolet-and-visible-tuv-radiation-model>, last access: 6 May 2023). Users can easily use this tool to convert the TUV model output results into ROMAC input files.

ROMAC provides users with an inline calculation module to calculate photolysis. In the current version, the inline calculation module of photolysis uses the algorithm provided by MCM, an algorithm based on the solar zenith angle (SZA). The trigonometric parameterization function is shown in Eq. (32). The parameters of l , m , n are provided by MCM (<http://mcm.york.ac.uk/>, last access: 6 May 2023).

$$J = l \times \cos(\text{SZA})^m \times e^{-n \times \sec(\text{SZA})} \quad (32)$$

If both the input photolysis rate and the inline calculated photolysis rate are present, ROMAC will use the input photolysis rate preferentially. In addition, ROMAC provides the user with a photolysis rate modification factor (J_{rate}), which can easily be used to adjust the photolysis rate in the model. The

default value of J_{rate} is 1.0, and the actual photolysis rate used in the model is the input rate or the inline calculated rate multiplied by J_{rate} .

2.4 Model accuracy and computational efficiency

The comparison of ROMAC with AtChem, FOAM and FACSIMILE, which is widely used for MCM, was performed on a PC with a CPU of 16-core AMD Ryzen 9 3950X at 3.5 GHz and 32 GB RAM. The operating system was 64 bit Ubuntu (version 20.04.1) and the software was compiled using Intel Fortran (ifort version 2021.2.0). The computational efficiency of the model is evaluated by CPU time. AtChem, ROMAC, and FACSIMILE are all run using a single core and the CPU time is recorded by the built-in function of the software. The CPU time used by FOAM is recorded by the function *cpitime* in MATLAB. The total integration time is 345 600 s, and the integration time step is 900 s. The settings of atol (10^{-4}) and rtol (10^{-3}) in the models are consistent. The temperature, pressure and humidity in the scenario simulation are 25 °C, 101.325 kPa and 35 %, respectively. The chemical mechanism used in this test is MCM v3.3.1, and the initial species concentrations are shown in Table A1. Since running the entire version of MCM v3.3.1 using AtChem is computationally excessive for our computing platform, we only selected the VOCs included in the EPA Photochemical Assessment Monitoring Stations (PAMS) Target List (<https://www.epa.gov/amtic/>, last access: 21 October 2023) and exported the mechanism file from the MCM website. In this test case, 3899 species and 11 814 chemical reactions were included.

In this study, we assumed that the solution results of AtChem based on the CVODE library are accurate. Therefore, the accuracy of the model is evaluated by calculating the relative difference between the solution results of ROMAC and AtChem (Eq. 33).

$$\text{RE}_t = \frac{|C_{\text{ROMAC},t} - C_{\text{AtChem},t}|}{|C_{\text{AtChem},t}|} \times 100 \% \quad (33)$$

Figure A1 shows a comparison of simulation results for nine species, including radicals and gaseous pollutants, which

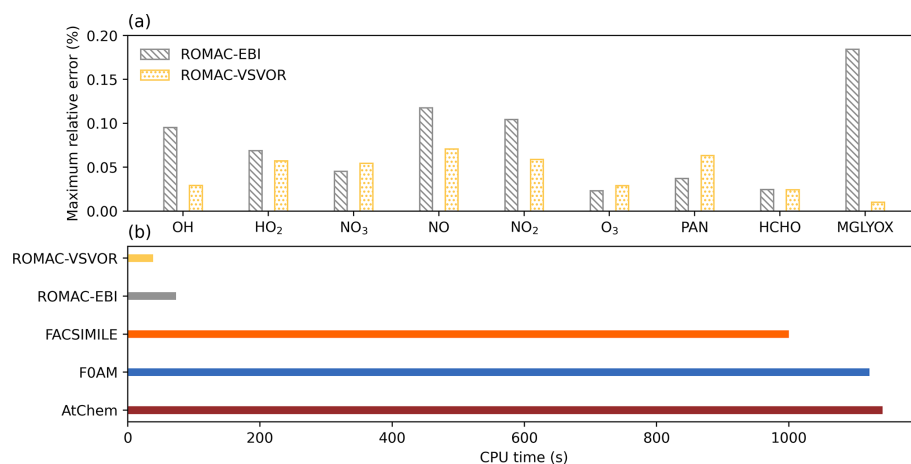


Figure 3. Accuracy evaluation and comparison of model computational efficiency. **(a)** Maximum relative error between the integration results of ROMAC and AtChem. **(b)** CPU time used to run compared with other models.

were commonly used in previous studies to evaluate solution results (Hertel et al., 1993; Esentürk et al., 2018; Aro, 1996b). The simulation results for ROMAC in Fig. A1 are processed by the VSVOR solver. As shown in Fig. A1, the solution results of ROMAC and AtChem are comparable, indicating that the solution results of ROMAC are comparable to the high-precision solution algorithm. The time series of relative error and its growth rates are depicted in Fig. A2. The relative difference between the solution results of ROMAC and that of AtChem is gradually stabilized, and the rate of change of the relative error after 270 000 s is extremely low, between $-1.0 \times 10^{-6} \% s^{-1}$ and $1.0 \times 10^{-6} \% s^{-1}$. Although the relative error of O₃ has a trend of continuous increase, the growth rate of the error remains stable and extremely low ($3.3 \times 10^{-8} \% s^{-1}$). Hence, the relative error remains within the preset rtol even if the simulation duration is extended by an additional 2.0×10^6 s at this growth rate. This suggests that the error of the ROMAC result can be stably controlled during long-term simulations. Figure 3 illustrates the maximum relative error in the scenario simulation and the CPU time used by each model. The maximum relative errors between the results of the VSVOR solver and the results of AtChem are all smaller than the preset rtol. The solution results obtained from the EBI solver were also evaluated in this study. The discrepancy of certain species in the results obtained by EBI exceeds the preset tolerance (0.1 %), such as NO, NO₂ and MGLYOX. Compared with the single-step EBI solver, the VSVOR solver with variable time step and variable order can better control the truncation error. In terms of CPU time required for execution, the VSVOR solver with higher solution accuracy is even more efficient than EBI. The CPU time consumed by EBI with different integration time steps is shown in Table A2. For the MCM chemical mechanism, the algorithm fails to converge when the integration time step is longer than 50 s. After a series of tests, we found that even with an integration time step of 10 s, the EBI solver was at

risk of failing to converge. However, reducing the integration time step too much diminishes the efficiency of the EBI solver when handling the MCM mechanism in comparison to the VSVOR solver. Hence, the VSVOR solver exhibits comparable computational efficiency to the EBI solver, while maintaining superior solution accuracy and stability.

Compared with other models, ROMAC has greatly improved the computational efficiency of solving large-scale chemical mechanisms. The computational efficiency of ROMAC is 97 % higher than that of FOAM and AtChem, and 96 % higher than that of FACSIMILE.

3 Model validation and application

3.1 Chamber simulation case

A chamber experiment for toluene degradation was used to evaluate the capabilities of ROMAC to dynamically optimize chemical and physical processes. In this case, the indoor smog chamber in JNU-VMDSC was used to simulate the degradation of toluene. The JNU-VMDSC provides a reliable experimental platform, and its structure and characterization (*e.g.*, wall loss, light intensity, airtightness test) have been described in a previous study (W. Wang et al., 2023). Toluene and isoprene were injected into the chamber before the UV light was turned on. The initial mixing ratios of toluene and isoprene were 2157 and 160 ppbv, respectively.

In order to simulate the effect of the dilution process on the toluene concentration, nitrogen was injected into the chamber at a flow of 7 L min^{-1} , while sampling was carried out at a flow of 7 L min^{-1} at the sampling port. Similar to previous studies (Dada et al., 2020; Jiang et al., 2020), the rate of dilution was calculated using Eqs. (34) and (35):

$$\left[\frac{dc}{dt} \right]_{\text{dilu}} = \frac{C \times dv}{V_{\text{chamber}} \times dt} = k_{\text{dilu}} \times C \quad (34)$$

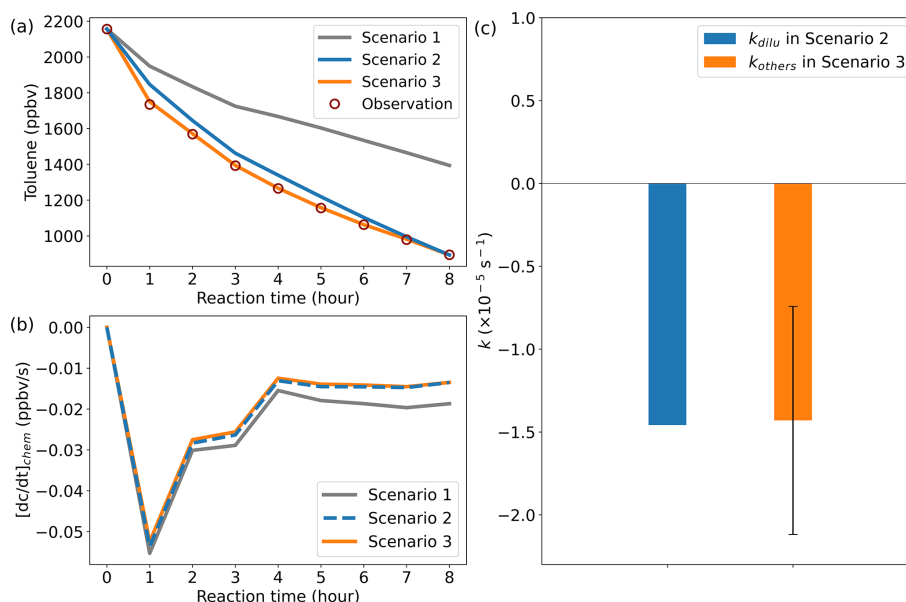


Figure 4. Model simulation results. **(a)** Comparison results between the simulated and observed toluene mixing ratios. **(b)** Chemical loss rate of toluene. **(c)** Comparison of kinetic constants in dilution process. Error bars indicate the standard deviation of k_{others} at different times in scenario 3.

$$k_{dilu} = \frac{1}{V_{chamber}} \times \frac{dv}{dt} = \frac{Flow}{V_{chamber}}, \quad (35)$$

where k_{dilu} is the rate constant of dilution, $V_{chamber}$ is the volume of the chamber (8000 L), and “Flow” is the flow of nitrogen injection. Therefore, the theoretical estimation result of k_{dilu} in this case is $1.458 \times 10^{-5} \text{ s}^{-1}$. Wall loss was not considered in this simple experiment with gaseous pollutants.

The version of the chemical mechanism used in the model simulations is MCM v3.3.1, all species and mechanisms in MCM are included. Three case scenarios were set up to evaluate the simulation capabilities of ROMAC. In scenario 1, only chemical processes were considered (Eq. 36). In scenario 2, chemical processes and dilution processes were considered (Eq. 37). In scenario 3, we assume that the results of the experiment are influenced by an unknown process, and this process is assumed to be a first-order kinetic process (Eq. 38). The k_{others} in scenario 3 was dynamically optimized with Scheme 1 as described in Sect. 2.2. Theoretically, the value of k_{others} obtained by the dynamic optimization in scenario 3 should be close to k_{dilu} in scenario 2.

$$\frac{dc_{Tolu}}{dt} = \left[\frac{dc_{Tolu}}{dt} \right]_{chem} \quad (36)$$

$$\begin{aligned} \frac{dc_{Tolu}}{dt} &= \left[\frac{dc_{Tolu}}{dt} \right]_{chem} + \left[\frac{dc_{Tolu}}{dt} \right]_{dilu} \\ &= \left[\frac{dc_{Tolu}}{dt} \right]_{chem} + k_{dilu} \times C_{Tolu} \end{aligned} \quad (37)$$

$$\begin{aligned} \frac{dc_{Tolu}}{dt} &= \left[\frac{dc_{Tolu}}{dt} \right]_{chem} + \left[\frac{dc_{Tolu}}{dt} \right]_{others} \\ &= \left[\frac{dc_{Tolu}}{dt} \right]_{chem} + k_{others} \times C_{Tolu} \end{aligned} \quad (38)$$

The total duration of the chamber experiment was 8 h, and the CPU time consumed by a single simulation of ROMAC was about 13 s. Figure 4 illustrates the comparison results between the simulated and observed toluene mixing ratios for different scenario cases. Due to the lack of dilution process in scenario 1, there is a large gap between simulation results and observations. After considering the dilution process, the simulation of scenario 2 was improved, which indicates that the setting of scenario 2 is reasonable. The absence of a physical process in the model is identified as the primary cause for failure to reproduce observations in scenario 1. The simulation results of scenario 3 agree well with the observations, which indicates that the dynamic optimization algorithm successfully captures the process that cannot be explained by the MCM chemical mechanism. Figure 4b illustrates the chemical loss rate of toluene under different simulation scenarios. The results of scenario 3 and scenario 2 are consistent and significantly different from the results of scenario 1. This indicates that the dynamic optimization algorithm can improve the chemical process while optimizing the physical process. Ignoring physical processes in the traditional box model may introduce large uncertainty to the simulation results. The rate of the physical process is subject to uncertainty in practical applications, but its average value is expected to closely approximate the theoretical value. The optimized value of k_{others} in scenario 3, as shown in Fig. 4c, exhibits a certain

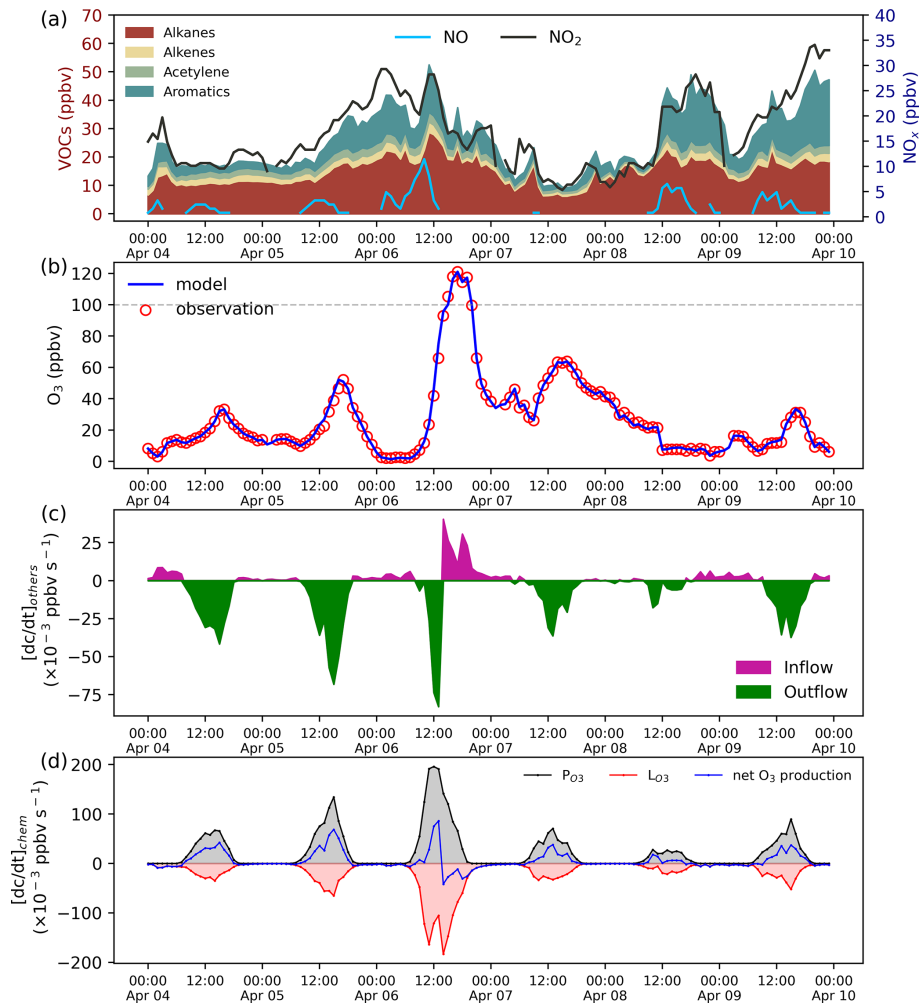


Figure 5. Species mixing ratio and the rate of O₃ change. **(a)** VOCs and NO_x mixing ratios. **(b)** Model and observation O₃ mixing ratios. **(c)** The effect of the physical process on the O₃ mixing ratios calculated by the adaptive dynamic optimization module. **(d)** The rate of O₃ chemical production.

range of fluctuations rather than a fixed value. However, its average values (1.430×10^{-5}) are comparable to k_{dilu} in scenario 2 (Fig. 4c), which indicates that the dynamically optimized algorithm is reliable.

3.2 Field observation case

This case demonstrates the application of ROMAC to the analysis of the photochemical process of O₃ formation and the dynamic optimization of physical processes. The observation data were obtained at the Heshan Atmospheric Super-site (22.728° N, 112.929° E) in Guangdong Province, China. A detailed description of the Heshan site can be found in previous publications (He et al., 2019; Yang et al., 2017). The observation period was from 4 to 10 April 2021. Meteorological parameters and the mixing ratios of NO_x, VOCs, SO₂, CO were constrained by Scheme 3. The concentrations

of NO_x and VOCs were shown in Fig. 5a, with the meteorological observations in Fig. A3.

The simulation of O₃ was constrained by Scheme 1. In this case, all physical processes of O₃ (e.g., dry deposition, dilution, transport) were merged into $[\frac{dc_{\text{O}_3}}{dt}]_{\text{others}}$. The rate of change of O₃ is shown in Eq. (39). The optimal estimate of $[\frac{dc_{\text{O}_3}}{dt}]_{\text{others}}$ uses Scheme 1 shown in Fig. 1.

$$\frac{dc_{\text{O}_3}}{dt} = \left[\frac{dc_{\text{O}_3}}{dt} \right]_{\text{chem}} + \left[\frac{dc_{\text{O}_3}}{dt} \right]_{\text{others}} \quad (39)$$

The comparison between the optimized simulation results and the observations of O₃ mixing ratios is shown in Fig. 5b. As expected, the model outputs are consistent with the observations due to the dynamic optimization. The estimated value of $[\frac{dc_{\text{O}_3}}{dt}]_{\text{others}}$ for the physical process is shown in Fig. 5c. Positive values of $[\frac{dc_{\text{O}_3}}{dt}]_{\text{others}}$ indicate that phys-

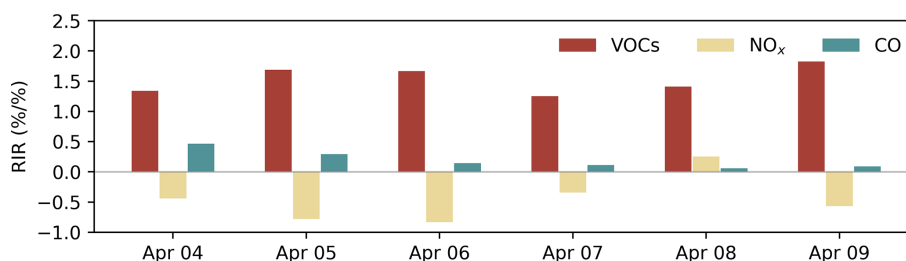


Figure 6. RIR values of O₃ precursors, i.e., VOCs, NO_x and CO.

ical processes increase local O₃ concentration (e.g., external transport), while negative values indicate a decrease in O₃ concentration (e.g., dilution, deposition). As displayed in Fig. 5c, $\left[\frac{dc_{O_3}}{dt}\right]_{\text{others}}$ is usually negative during the daytime, indicating that O₃ was transported out of the region after formation by photochemical processes. However, positive values of $\left[\frac{dc_{O_3}}{dt}\right]_{\text{others}}$ can also occur during the daytime. On 6 April, the surface ozone mixing ratio increased rapidly, and the maximum hourly mixing ratio exceeded China II Emission Standard (> 100 ppbv). The value of $\left[\frac{dc_{O_3}}{dt}\right]_{\text{others}}$ on the afternoon of 6 April is positive, indicating that physical processes were one of the reasons for the occurrence of O₃ pollution.

The rate of O₃ chemical production and precursor sensitivities were calculated using a method described in previous studies (Liu et al., 2022; Wang et al., 2018). As displayed in Fig. 5d, the net O₃ production rate on 6 April 2021 was significantly higher than that on other days, indicating that chemical processes were also an important cause of O₃ pollution. The sensitivity of the O₃ formation to its precursors can be represented by relative incremental reactivity (RIR). Figure 6 shows the daily average RIR values of VOCs, NO_x and CO. The RIR values of VOCs and CO were positive, which indicates that reducing the concentration of VOCs and CO can effectively reduce the chemical formation of O₃. Except for 8 April, the RIR values of NO_x were negative, indicating that decreasing the NO_x concentration leads to an increase in O₃ concentration. The negative values of RIR for NO_x and higher positive values of RIR for VOC indicate that the ozone formation at the Heshan Atmospheric Supersite was mostly likely under a VOC limited regime. The result was consistent with a previous study (He et al., 2019), indicating that the application of ROMAC in chemical process diagnosis is reliable.

The application of this case demonstrates the ability of ROMAC to quantify the contribution of physical and chemical processes to air pollutant concentrations. Compared with the traditional observation-based box model (OBM), ROMAC is superior in evaluating the impact of physical processes on pollutant concentrations. Compared with emission-based 3-D air quality models (e.g., CMAQ, WRF-Chem, NAQPMS), the observation-based dynamic optimization al-

gorithm in ROMAC reduces the uncertainty introduced by emission inventory and meteorological simulation.

4 Future developments

ROMAC will be continuously updated and developed. Functionality for future improvements and upgrades includes the following:

- A multiphase chemical reaction module and a module for gas–particle partitioning and sectional simulation are being developed.
- Adjoint sensitivity analysis will be added in a future version, and users can use ROMAC to analyze the relationship between precursors and secondary pollutants.
- The ensemble Kalman filter (EnKF) will be added to dynamically optimize the physical process in future versions.
- In a future development roadmap, we have plans to introduce a modeling framework version of ROMAC known as “ROMAC plug-in.” This ROMAC plug-in will support calls from Python or Fortran, ensuring compatibility and flexibility for users. Importantly, the efficient design of ROMAC will be maintained, allowing for an optimized performance. The kernel of the ROMAC plug-in will be specifically engineered to provide users with flexibility to effortlessly construct their own models or integrate ROMAC with existing frameworks, such as CTMs.

Appendix A

Table A1. Initial species concentration used for model comparisons (unit: molecules cm⁻³).

Species	Concentration	Species	Concentration	Species	Concentration
O ₃	5.20 × 10 ¹⁰	NC ₁₁ H ₂₄	4.90 × 10 ⁸	TM123B	1.20 × 10 ⁹
NO ₂	9.80 × 10 ¹¹	NC ₁₂ H ₂₆	9.80 × 10 ⁸	STYRENE	5.90 × 10 ⁹
NO	9.80 × 10 ¹¹	C ₂ H ₄	4.70 × 10 ¹⁰	C ₄ H ₆	4.90 × 10 ⁸
CO	1.50 × 10 ¹³	C ₃ H ₆	5.70 × 10 ⁹	BENZAL	1.90 × 10 ¹⁰
SO ₂	7.50 × 10 ¹⁰	BUT1ENE	7.40 × 10 ⁸	CH ₃ COCH ₃	9.10 × 10 ⁹
NO ₃	1.40 × 10 ⁸	TBUT2ENE	2.50 × 10 ⁸	MEK	4.90 × 10 ¹⁰
C ₂ H ₂	6.20 × 10 ¹⁰	C ₅ H ₈	2.50 × 10 ⁸		
C ₂ H ₆	9.70 × 10 ¹⁰	PENT1ENE	1.80 × 10 ⁹		
C ₃ H ₈	1.40 × 10 ¹¹	TPENT2ENE	2.50 × 10 ⁸		
IC ₄ H ₁₀	5.00 × 10 ¹⁰	CPENT2ENE	1.50 × 10 ⁹		
NC ₄ H ₁₀	9.90 × 10 ¹⁰	HEX1ENE	2.50 × 10 ⁸		
IC ₅ H ₁₂	1.30 × 10 ¹¹	TOLUENE	1.00 × 10 ¹¹		
NC ₅ H ₁₂	1.70 × 10 ¹¹	BENZENE	1.40 × 10 ¹⁰		
CHEX	1.70 × 10 ⁹	EBENZ	4.20 × 10 ¹⁰		
M22C4	1.20 × 10 ⁹	OXYL	5.80 × 10 ¹⁰		
M23C4	7.20 × 10 ⁹	IPBENZ	1.50 × 10 ⁹		
M3PE	7.40 × 10 ⁹	PBENZ	1.20 × 10 ⁹		
NC ₆ H ₁₄	8.40 × 10 ⁹	OETHTOL	1.50 × 10 ⁹		
M2HEX	6.60 × 10 ⁹	METHTOL	2.00 × 10 ⁹		
M3HEX	6.40 × 10 ⁹	TM135B	2.50 × 10 ⁹		
NC ₇ H ₁₆	4.40 × 10 ⁹	HCHO	1.20 × 10 ¹¹		
NC ₈ H ₁₈	4.40 × 10 ⁹	CH ₃ CHO	3.90 × 10 ¹⁰		
NC ₉ H ₂₀	4.40 × 10 ⁹	C ₂ H ₅ CHO	3.40 × 10 ⁹		
NC ₁₀ H ₂₂	1.50 × 10 ⁹	C ₃ H ₇ CHO	1.70 × 10 ⁹		
PETHTOL	2.00 × 10 ⁹	MIBK	3.80 × 10 ¹¹		
TM124B	2.20 × 10 ⁹	HEX2ONE	6.30 × 10 ¹¹		

Table A2. CPU time used by the EBI solver at different integration time step sizes (unit: seconds). “Nonconvergence” represents that the EBI solver fails to converge.

Time step	1	10	50	120	900
CPU time	256	73	Nonconvergence	Nonconvergence	Nonconvergence

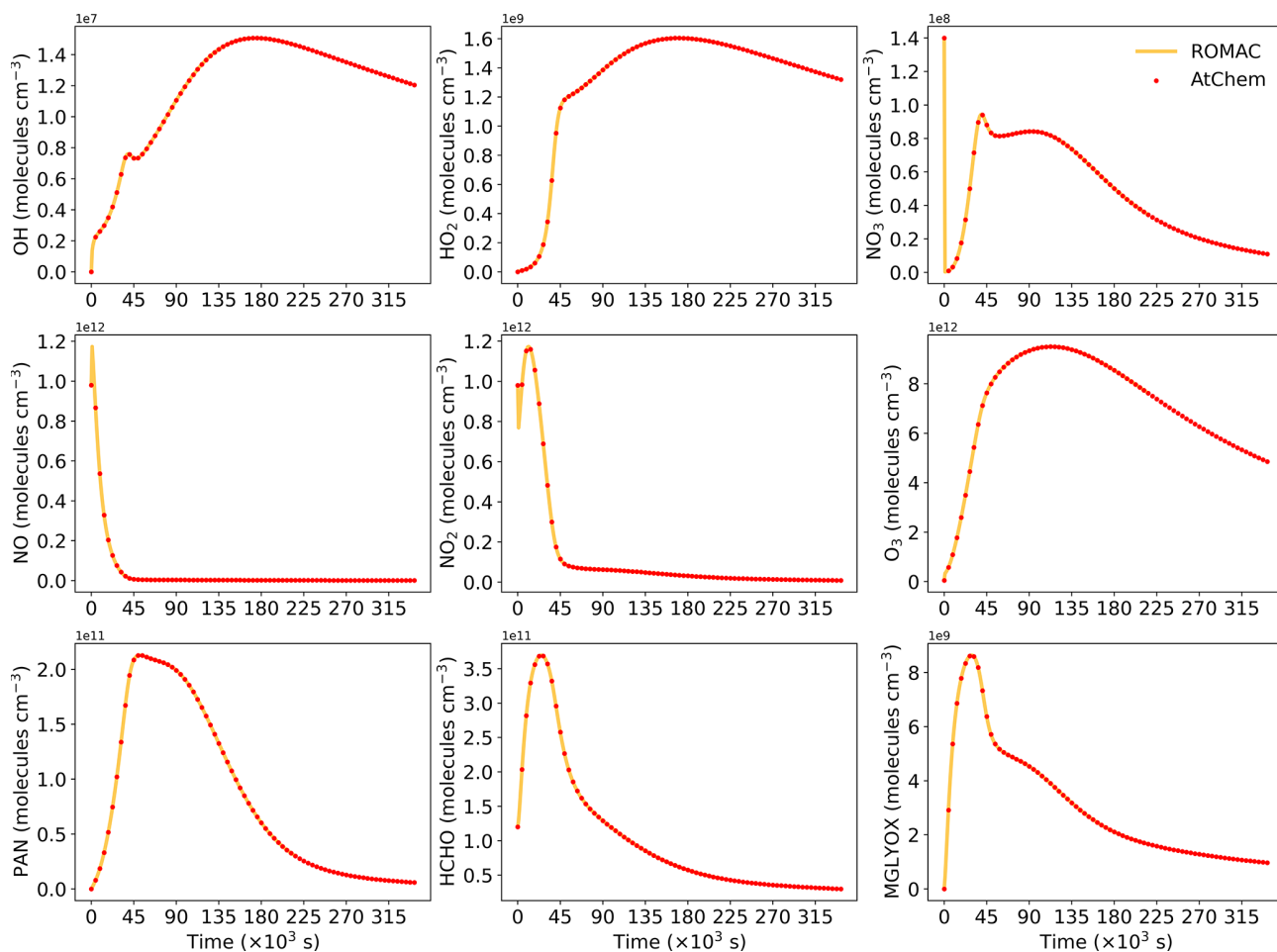


Figure A1. Comparison of the simulation results between ROMAC and AtChem for nine substances. ROMAC used the VSVOR solver in this test.

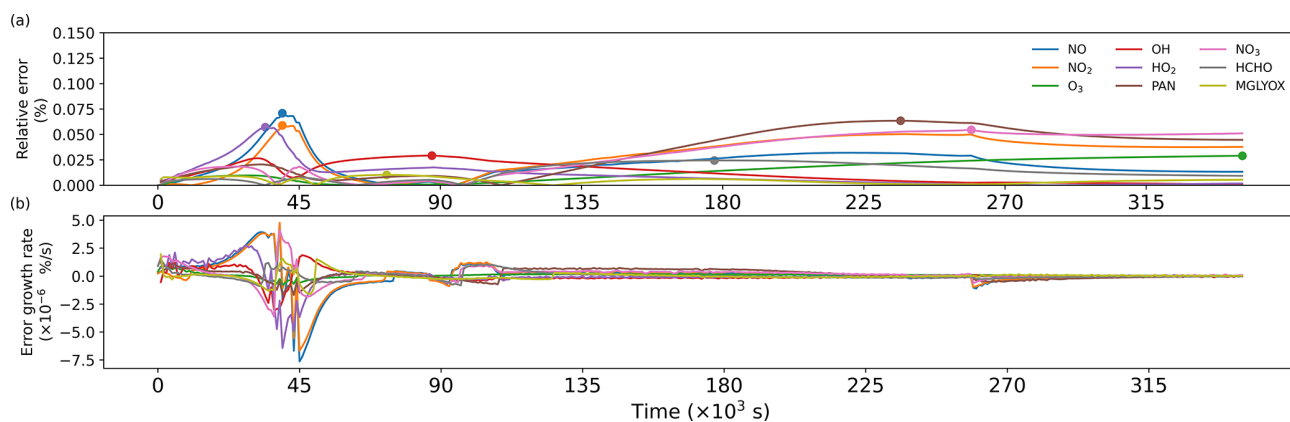


Figure A2. (a) Time series of relative errors, with dots marking the maximum values. (b) Growth rate of relative errors.

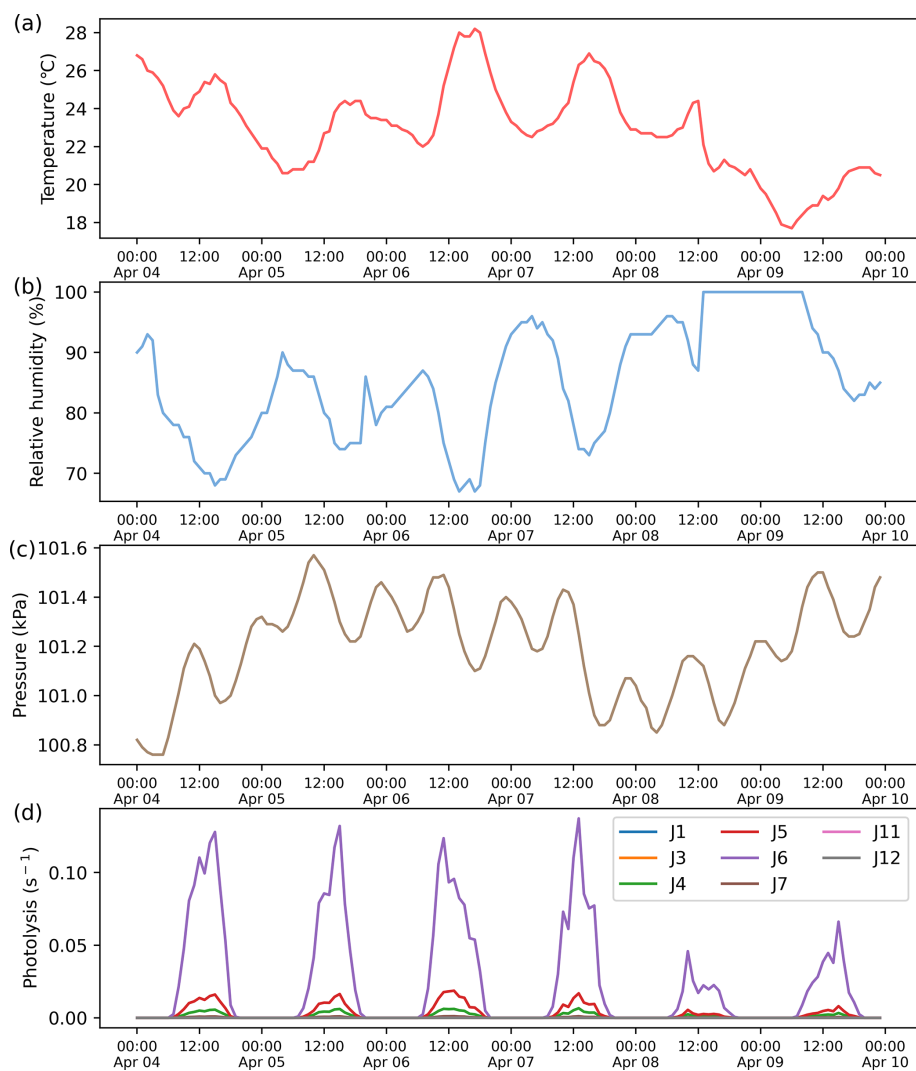


Figure A3. Meteorological data input to the model. (a) Temperature. (b) Relative humidity. (c) Atmospheric pressure. (d) Photolysis rate.

Appendix B: Nomenclature

ODEs	Ordinary differential equations
VSVOR	Variable-step and variable-order solver
atol	absolute tolerance
rtol	relative tolerance
r	The reactant in a chemical reaction
p	The product in a chemical reaction
α, β	Stoichiometric number
C_t	Concentration of species at time t
$f_i(C_{i,t}, t)$	Rate of change of species i at time t
$P_{i,t}$	Product rate of species i at time t
$L_{i,t}$	Loss rate of species i at time t
$l_{i,t,R}$	The part of the chemical reaction rate that is not directly related to the concentration of species i in reaction R at time t
Δt	Integration time step size
$g_1(C_{t+1})$	The objective function when Newton's method solves the implicit Euler method
$g_2(C_{t+1})$	The objective function when Newton's method solves the implicit trapezoidal method
C_{t+1}^k	Species concentration at iteration k of Newton's method
$\nabla g_1(C_{t+1})$	The Jacobian matrix of $g_1(C_{t+1})$
$\nabla g_2(C_{t+1})$	The Jacobian matrix of $g_2(C_{t+1})$
$\nabla g^{-1}(C_{t+1})$	The inverse of the Jacobian matrix
Δt_0	Integration time step size equal to 2.22×10^{-16} s
Δt_1	Minimum species atmospheric lifetime in chemical mechanisms
Δt_2	The maximum time step size necessary to achieve diagonal dominance of the Jacobian matrix.
Δt_{init}	Initial integration time step size
Δt_{max}	The maximum integration time step to ensure the result does not exceed the preset tolerance
Δt_{opt}	Optimal integration step size
RERR	Relative error calculated by doubled-step method
LTE	Local truncation error
atol	Absolute tolerance
rtol	Relative tolerance
Rn	Lagrangian remainder in the Taylor expansion
ξ	Real number in the Lagrangian remainder in the Taylor expansion
RMSE	Root mean square error

Code and data availability. The current version of ROMAC coupled MCM v3.3.1 is archived on Zenodo: <https://doi.org/10.5281/zenodo.7900781> (Li, 2023a) under the Attribution 4.0 International licence.

The input data used to produce the results used in this paper is archived on Zenodo (<https://doi.org/10.5281/zenodo.7900710>, Li, 2023b).

Author contributions. JL: The developer of all model source code and algorithms for ROMAC; Conceptualization; Formal analysis; Writing – Original Draft.

CZ: Formal analysis; Writing – Review & Editing.

WZ: Formal analysis; Software testing.

SH: The principal investigator of chamber study case; Data curation.

YW: Model Comparison and Evaluation.

HW: Funding acquisition; Writing – review & editing.

BW: Funding acquisition; Writing – review & editing.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Financial support. This work was supported by the National Natural Science Foundation of China (grant nos. 42121004 and 42077190), and Science and Technology Project of Guangdong Province of China (grant no. 2019B121202002).

Review statement. This paper was edited by Yilong Wang and reviewed by two anonymous referees.

References

- Alvanos, M. and Christoudias, T.: GPU-accelerated atmospheric chemical kinetics in the ECHAM/MESSy (EMAC) Earth system model (version 2.52), *Geosci. Model Dev.*, 10, 3679–3693, <https://doi.org/10.5194/gmd-10-3679-2017>, 2017.
- Aro, C. J.: CHEMSODE: a stiff ODE solver for the equations of chemical kinetics, *Comput. Phys. Commun.*, 97, 304–314, 1996a.
- Aro, C. J.: A stiff ODE preconditioner based on Newton linearization, *Appl. Numer. Math.*, 21, 335–352, 1996b.
- Cariolle, D., Moinat, P., Teyssède, 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.
- Carter, W. P. L.: Development of the SAPRC-07 chemical mechanism, *Atmos. Environ.*, 44, 5324–5335, <https://doi.org/10.1016/j.atmosenv.2010.01.026>, 2010.
- Chen, Y. Z., Sexton, K. G., Jerry, R. E., Surratt, J. D., and Vizuete, W.: Assessment of SAPRC07 with updated isoprene chemistry against outdoor chamber experiments, *Atmos. Environ.*, 105, 109–120, <https://doi.org/10.1016/j.atmosenv.2015.01.042>, 2015.
- Cheng, H. R., Guo, H., Saunders, S. M., Lam, S. H. M., Jiang, F., Wang, X. M., Simpson, I. J., Blake, D. R., Louie, P. K. K., and Wang, T. J.: Assessing photochemical ozone formation in the Pearl River Delta with a photochemical trajectory model, *Atmos. Environ.*, 44, 4199–4208, <https://doi.org/10.1016/j.atmosenv.2010.07.019>, 2010.
- IEEE: IEEE Standard for Floating-Point Arithmetic, *IEEE Std 754-2019* (Revision of IEEE 754-2008), 1–84, 22 July 2019, <https://doi.org/10.1109/IEEESTD.2019.8766229>, 2008.
- Dada, L., Lehtipalo, K., Kontkanen, J., Nieminen, T., Baalbaki, R., Ahonen, L., Duplissy, J., Yan, C., Chu, B., Petaja, T., Lehtinen, K., Kerminen, V. M., Kulmala, M., and Kangasluoma, J.: Formation and growth of sub-3-nm aerosol particles in experimental chambers, *Nat. Protoc.*, 15, 1013–1040, <https://doi.org/10.1038/s41596-019-0274-z>, 2020.
- Decker, Z. C. J., Zarzana, K. J., Coggon, M., Min, K. E., Pollack, I., Ryerson, T. B., Peischl, J., Edwards, P., Dube, W. P., Markovic, M. Z., Roberts, J. M., Veres, P. R., Graus, M., Warneke, C., de Gouw, J., Hatch, L. E., Barsanti, K. C., and Brown, S. S.: Nighttime Chemical Transformation in Biomass Burning Plumes: A Box Model Analysis Initialized with Aircraft Observations, *Environ. Sci. Technol.*, 53, 2529–2538, <https://doi.org/10.1021/acs.est.8b05359>, 2019.
- Decker, Z. C. J., Robinson, M. A., Barsanti, K. C., Bourgeois, I., Coggon, M. M., DiGangi, J. P., Diskin, G. S., Flocke, F. M., Franchin, A., Fredrickson, C. D., Gkatzelis, G. I., Hall, S. R., Halliday, H., Holmes, C. D., Huey, L. G., Lee, Y. R., Lindaas, J., Middlebrook, A. M., Montzka, D. D., Moore, R., Neuman, J. A., Nowak, J. B., Palm, B. B., Peischl, J., Piel, F., Rickly, P. S., Rollins, A. W., Ryerson, T. B., Schwantes, R. H., Sekimoto, K., Thornhill, L., Thornton, J. A., Tyndall, G. S., Ullmann, K., Van Rooy, P., Veres, P. R., Warneke, C., Washenfelder, R. A., Weinheimer, A. J., Wiggins, E., Winstead, E., Wisthaler, A., Womack, C., and Brown, S. S.: Nighttime and daytime dark oxidation chemistry in wildfire plumes: an observation and model analysis of FIREX-AQ aircraft data, *Atmos. Chem. Phys.*, 21, 16293–16317, <https://doi.org/10.5194/acp-21-16293-2021>, 2021.
- Djouad, R. and Sportisse, B.: Partitioning techniques and lumping computation for reducing chemical kinetics. APLA: An automatic partitioning and lumping algorithm, *Appl. Numer. Math.*, 43, 383–398, [https://doi.org/10.1016/S0168-9274\(02\)00111-3](https://doi.org/10.1016/S0168-9274(02)00111-3), 2002.
- Emmons, L. K., Walters, S., Hess, P. G., Lamarque, J.-F., Pfister, G. G., Fillmore, D., Granier, C., Guenther, A., Kinnison, D., Laepple, T., Orlando, J., Tie, X., Tyndall, G., Wiedinmyer, C., Baughcum, S. L., and Kloster, S.: Description and evaluation of the Model for Ozone and Related chemical Tracers, version 4 (MOZART-4), *Geosci. Model Dev.*, 3, 43–67, <https://doi.org/10.5194/gmd-3-43-2010>, 2010.
- Esentürk, E., Abraham, N. L., Archer-Nicholls, S., Mitsakou, C., Griffiths, P., Archibald, A., and Pyle, J.: Quasi-Newton methods for atmospheric chemistry simulations: implementation

- in UKCA UM vn10.8, *Geosci. Model Dev.*, 11, 3089–3108, <https://doi.org/10.5194/gmd-11-3089-2018>, 2018.
- He, Z., Wang, X., Ling, Z., Zhao, J., Guo, H., Shao, M., and Wang, Z.: Contributions of different anthropogenic volatile organic compound sources to ozone formation at a receptor site in the Pearl River Delta region and its policy implications, *Atmos. Chem. Phys.*, 19, 8801–8816, <https://doi.org/10.5194/acp-19-8801-2019>, 2019.
- Hertel, O., Berkowicz, R., and Christensen, J.: Test of two numerical schemes for use in atmosphere, *Atmos. Environ.*, 27, 2591–2611, 1993.
- Huang, L. and Topping, D.: JIBox v1.1: a Julia-based multi-phase atmospheric chemistry box model, *Geosci. Model Dev.*, 14, 2187–2203, <https://doi.org/10.5194/gmd-14-2187-2021>, 2021.
- Jenkin, M. E., Young, J. C., and Rickard, A. R.: The MCM v3.3.1 degradation scheme for isoprene, *Atmos. Chem. Phys.*, 15, 11433–11459, <https://doi.org/10.5194/acp-15-11433-2015>, 2015.
- Jiang, X., Lv, C., You, B., Liu, Z., Wang, X., and Du, L.: Joint impact of atmospheric SO₂ and NH₃ on the formation of nanoparticles from photo-oxidation of a typical biomass burning compound, *Environmental Science: Nano*, 7, 2532–2545, <https://doi.org/10.1039/d0en00520g>, 2020.
- Jimenez, P.: Comparison of photochemical mechanisms for air quality modeling, *Atmos. Environ.*, 37, 4179–4194, [https://doi.org/10.1016/s1352-2310\(03\)00567-3](https://doi.org/10.1016/s1352-2310(03)00567-3), 2003.
- Li, J.: ROMAC v1.0 (1.0), Zenodo [code], <https://doi.org/10.5281/zenodo.7900781>, 2023a.
- Li, J.: Supplementary data, Zenodo [data set], <https://doi.org/10.5281/zenodo.7900710>, 2023b.
- Li, X. B., Fan, G., Lou, S., Yuan, B., Wang, X., and Shao, M.: Transport and boundary layer interaction contribution to extremely high surface ozone levels in eastern China, *Environ. Pollut.*, 268, 115804, <https://doi.org/10.1016/j.envpol.2020.115804>, 2021.
- Lin, H., Long, M. S., Sander, R., Sandu, A., Yantosca, R. M., Estrada, L. A., Shen, L., and Jacob, D. J.: An adaptive auto-reduction solver for speeding up integration of chemical kinetics in atmospheric chemistry models: Implementation and evaluation in the Kinetic Pre-Processor (KPP) version 3.0.0, *J. Adv. Model. Earth Sy.*, 15, e2022MS003293, <https://doi.org/10.1029/2022MS003293>, 2023.
- Ling, Z. H., Zhao, J., Fan, S. J., and Wang, X. M.: Sources of formaldehyde and their contributions to photochemical O(3) formation at an urban site in the Pearl River Delta, southern China, *Chemosphere*, 168, 1293–1301, <https://doi.org/10.1016/j.chemosphere.2016.11.140>, 2017.
- Liu, T., Hong, Y., Li, M., Xu, L., Chen, J., Bian, Y., Yang, C., Dan, Y., Zhang, Y., Xue, L., Zhao, M., Huang, Z., and Wang, H.: Atmospheric oxidation capacity and ozone pollution mechanism in a coastal city of southeastern China: analysis of a typical photochemical episode by an observation-based model, *Atmos. Chem. Phys.*, 22, 2173–2190, <https://doi.org/10.5194/acp-22-2173-2022>, 2022.
- Mott, D. R., Oran, E. S., and van Leer, B.: A Quasi-Steady-State Solver for the Stiff Ordinary Differential Equations of Reaction Kinetics, *J. Comput. Phys.*, 164, 407–428, <https://doi.org/10.1006/jcph.2000.6605>, 2000.
- Novelli, A., Kaminski, M., Rolletter, M., Acir, I.-H., Bohn, B., Dorn, H.-P., Li, X., Lutz, A., Nehr, S., Rohrer, F., Tillmann, R., Wegener, R., Holland, F., Hofzumahaus, A., Kiendler-Scharr, A., Wahner, A., and Fuchs, H.: Evaluation of OH and HO₂ concentrations and their budgets during photooxidation of 2-methyl-3-butene-2-ol (MBO) in the atmospheric simulation chamber SAPHIR, *Atmos. Chem. Phys.*, 18, 11409–11422, <https://doi.org/10.5194/acp-18-11409-2018>, 2018.
- O’Meara, S. P., Xu, S., Topping, D., Alfara, M. R., Capes, G., Lowe, D., Shao, Y., and McFiggans, G.: PyCHAM (v2.1.1): a Python box model for simulating aerosol chambers, *Geosci. Model Dev.*, 14, 675–702, <https://doi.org/10.5194/gmd-14-675-2021>, 2021.
- Sander, R., Baumgaertner, A., Gromov, S., Harder, H., Jöckel, P., Kerkweg, A., Kubistin, D., Regelin, E., Riede, H., Sandu, A., Taraborrelli, D., Tost, H., and Xie, Z.-Q.: The atmospheric chemistry box model CAABA/MECCA-3.0, *Geosci. Model Dev.*, 4, 373–380, <https://doi.org/10.5194/gmd-4-373-2011>, 2011.
- Sander, R., Baumgaertner, A., Cabrera-Perez, D., Frank, F., Gromov, S., Grooß, J.-U., Harder, H., Huijnen, V., Jöckel, P., Karydis, V. A., Niemeyer, K. E., Pozzer, A., Riede, H., Schultz, M. G., Taraborrelli, D., and Tauer, S.: The community atmospheric chemistry box model CAABA/MECCA-4.0, *Geosci. Model Dev.*, 12, 1365–1385, <https://doi.org/10.5194/gmd-12-1365-2019>, 2019.
- 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, 3459–3472, 1997a.
- Sandu, A., Verwer, J. G., Loon, M. V., Carmichael, G. R., Potra, F. A., Dabdub, D., and Seinfeld, J. H.: Benchmarking stiff ode solvers for atmospheric chemistry problems-I. implicit vs explicit, *Atmos. Environ.*, 31, 3151–3166, 1997b.
- Shen, L., Jacob, D. J., Santillana, M., Bates, K., Zhuang, J., and Chen, W.: A machine-learning-guided adaptive algorithm to reduce the computational cost of integrating kinetics in global atmospheric chemistry models: application to GEOS-Chem versions 12.0.0 and 12.9.1, *Geosci. Model Dev.*, 15, 1677–1687, <https://doi.org/10.5194/gmd-15-1677-2022>, 2022.
- Sommariva, R., Cox, S., Martin, C., Boroniska, K., Young, J., Jimack, P. K., Pilling, M. J., Matthaios, V. N., Nelson, B. S., Newland, M. J., Panagi, M., Bloss, W. J., Monks, P. S., and Rickard, A. R.: AtChem (version 1), an open-source box model for the Master Chemical Mechanism, *Geosci. Model Dev.*, 13, 169–183, <https://doi.org/10.5194/gmd-13-169-2020>, 2020.
- Verwer, J. G., Blom, J. G., Loon, M. V., and Spee, E. J.: A comparison of stiff ODE solvers for atmospheric chemistry problems, *Atmos. Environ.*, 30, 49–58, 1996.
- Wang, W., Xiao, Y., Han, S., Zhang, Y., Gong, D., Wang, H., and Wang, B.: A vehicle-mounted dual-smog chamber: Characterization and its preliminary application to evolutionary simulation of photochemical processes in a quasi-realistic atmosphere, *J. Environ. Sci.*, 132, 98–108, <https://doi.org/10.1016/j.jes.2022.07.034>, 2023.
- Wang, Y., Wang, H., Guo, H., Lyu, X., Cheng, H., Ling, Z., Louie, P. K. K., Simpson, I. J., Meinardi, S., and Blake, D. R.: Long-term O₃-precursor relationships in Hong Kong: field observation and model simulation, *Atmos. Chem. Phys.*, 17, 10919–10935, <https://doi.org/10.5194/acp-17-10919-2017>, 2017.
- Wang, Y., Guo, H., Zou, S., Lyu, X., Ling, Z., Cheng, H., and Zeren, Y.: Surface O₃ photochemistry over the

- South China Sea: Application of a near-explicit chemical mechanism box model, *Environ. Pollut.*, 234, 155–166, <https://doi.org/10.1016/j.envpol.2017.11.001>, 2018.
- Wang, Y., Guo, H., Lyu, X., Zhang, L., Zeren, Y., Zou, S., and Ling, Z.: Photochemical evolution of continental air masses and their influence on ozone formation over the South China Sea, *Sci. Total Environ.*, 673, 424–434, <https://doi.org/10.1016/j.scitotenv.2019.04.075>, 2019.
- Wang, Y., Liu, T., Gong, D., Wang, H., Guo, H., Liao, M., Deng, S., Cai, H., and Wang, B.: Anthropogenic Pollutants Induce Changes in Peroxyacetyl Nitrate Formation Intensity and Pathways in a Mountainous Background Atmosphere in Southern China, *Environ. Sci. Technol.*, 57, 6253–6262, <https://doi.org/10.1021/acs.est.2c02845>, 2023.
- Wolfe, G. M., Marvin, M. R., Roberts, S. J., Travis, K. R., and Liao, J.: The Framework for 0-D Atmospheric Modeling (F0AM) v3.1, *Geosci. Model Dev.*, 9, 3309–3319, <https://doi.org/10.5194/gmd-9-3309-2016>, 2016.
- Yang, Y., Shao, M., Keßel, S., Li, Y., Lu, K., Lu, S., Williams, J., Zhang, Y., Zeng, L., Nölscher, A. C., Wu, Y., Wang, X., and Zheng, J.: How the OH reactivity affects the ozone production efficiency: case studies in Beijing and Heshan, China, *Atmos. Chem. Phys.*, 17, 7127–7142, <https://doi.org/10.5194/acp-17-7127-2017>, 2017.
- Yarwood, G.: Development, Evaluation and Testing of Version 6 of the Carbon Bond Chemical Mechanism (CB6), Texas Commission on Environmental Quality, https://www.tceq.texas.gov/airquality/airmod/project/pj_report_pm.html (last access: 21 October 2023), 2010.
- Ying, Q. and Li, J.: Implementation and initial application of the near-explicit Master Chemical Mechanism in the 3D Community Multiscale Air Quality (CMAQ) model, *Atmos. Environ.*, 45, 3244–3256, <https://doi.org/10.1016/j.atmosenv.2011.03.043>, 2011.
- Young, T. R. and Boris, J. P.: A numerical technique for solving stiff ordinary differential equations associated with the chemical kinetics of reactive-flow problems, *J. Phys. Chem.*, 81, 2424–2427, <https://doi.org/10.1021/j100540a018>, 1977.
- Zhang, L., Brook, J. R., and Vet, R.: A revised parameterization for gaseous dry deposition in air-quality models, *Atmos. Chem. Phys.*, 3, 2067–2082, <https://doi.org/10.5194/acp-3-2067-2003>, 2003.