A Patankar Predictor-Corrector Approach for Positivity-Preserving Time Integration

Kamila Nurkhametova

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Adrian Sandu, Chair
Alexey Onufriev
Young Cao

August 13, 2025
Blacksburg, Virginia

Keywords: Positivity-preserving numerical methods, Production–destruction systems,
Ordinary differential equations

A Patankar Predictor-Corrector Approach for Positivity-Preserving Time Integration

Kamila Nurkhametova

ABSTRACT

In many physical, biological, and chemical systems, the underlying dynamics are modeled by systems of ordinary differential equations in which state variables such as species concentrations must remain non-negative and often satisfy conservation laws. Standard time integration methods, including classical Runge-Kutta schemes, can violate these structural properties, leading to non-physical solutions. This thesis presents a novel positivity-preserving correction strategy applicable to general time integration schemes, with a particular focus on Runge-Kutta methods. The proposed method operates as a predictor-corrector framework, using algebraic post-processing to clip negative stage values and apply diagonal scaling to enforce both positivity and conservation. A series of benchmark problems, including the stratospheric reaction system, the MAPK cascade, and the Robertson reaction, is used to evaluate the performance of the corrected integrators. Results show that the corrected schemes successfully preserve qualitative properties without compromising numerical stability. Efficiency tests demonstrate that while corrections introduce overhead in some stiff regimes, they may also improve performance. Order verification experiments prove that the correction mechanism does not change the formal order. Overall, the proposed method provides a practical and effective approach to enforcing structural constraints in the numerical integration of stiff production-destruction systems.

A Patankar Predictor-Corrector Approach for Positivity-Preserving Time Integration

Kamila Nurkhametova

GENERAL AUDIENCE ABSTRACT

Many natural processes, such as chemical reactions or biological systems, are described using mathematical equations that track how different quantities change over time. In these systems, certain values, like the amount of a chemical, can never be negative and must sometimes follow strict conservation rules, for example, the total mass conservation rule. However, the computer methods that are used to simulate these processes often produce results that violate these basic rules, giving unrealistic or "non-physical" outcomes, such as negative concentrations of a substance. This thesis introduces a new correction technique that ensures these simulations always follow the natural rules of the system. The method adjusts the simulation results whenever they become unrealistic, while still maintaining accuracy and efficiency. The approach is tested on several well-known examples from chemistry, biology, and physics. The results show that the corrected simulations remain realistic and stable without losing accuracy.

Acknowledgments

I would like to express my deepest gratitude to my academic advisor, Dr. Adrian Sandu, for giving me the opportunity to work with him in the Computational Science Laboratory. This work would not have been possible without his continuous support, insightful guidance, and profound expertise in ODE integrators. I am also thankful to all members of the Computational Science Laboratory for their help and encouragement, especially R. Gomillion, A. Novotny, A. Barry, and A. Bhattacharjee.

Finally, I would like to express my heartfelt gratitude to my mother, father, and dear sister. Without their support, this would not have been possible.

Contents

List of Figures

Chapter 1

Introduction

Many systems are described by processes that naturally involve strictly non-negative values such as concentrations, biological populations, or energy levels. These processes are often modeled using systems of ordinary differential equations (ODEs), especially ones that are structured as production-destruction systems, where components engage with each other through transfer, conversion, or decay. In such conditions, numerical integrators must preserve two properties of the true solution: positivity, which ensures that variables remain non-negative, and conservation, which maintains the total mass or other invariants of the system.

A fundamental requirement in the numerical simulation of chemical, biological, and physical systems is the preservation of positivity—the property that all concentrations and densities must remain non-negative throughout the evolution of the system. This is not only a natural physical requirement but also a mathematical necessity to avoid producing non-physical or unstable behaviors. As shown by Hundsdorfer and Verwer [6], even simple reaction systems such as $A + B \xrightarrow{k} C$ can exhibit severe numerical issues if positivity is violated. They show that negative initial concentrations can lead to unstable solutions or finite-time blow-up, even though the exact solution remains well-behaved for non-negative initial values. Furthermore, positivity violations may disrupt conservation laws and invalidate any meaningful physical interpretation of the simulation results [6]. From a computational standpoint, this implies that even a small negative value introduced by a numerical error can cascade into instability,

especially in stiff or nonlinear systems. Therefore, time integration methods for such systems must be carefully constructed to preserve positivity, unless we are willing to accept results that are physically meaningless or numerically divergent.

However, standard numerical integration methods, including Runge-Kutta schemes, may violate these properties when applied to stiff or nonlinear systems, leading to non-physical results such as negative concentrations or loss of mass. To address these limitations, a variety of positivity-preserving schemes have been developed. Modified Patankar methods, nonstandard finite difference schemes, and geometric conservative (GeCo) integrators are among the widely used approaches. These methods include algebraic modifications, scaling techniques, or stability-aware formulations to maintain positivity and invariants. While these approaches can guarantee positivity, they often suffer from order reduction, additional computational cost, or difficulty extending beyond low-order accuracy, particularly for stiff systems.

In this thesis, we propose a general positivity-preserving correction framework that can be applied as a predictor–corrector post-processing step to existing time integration methods. The key idea is to use the underlying integrator, such as singly diagonally implicit Runge–Kutta (SDIRK) schemes, to compute a predictor solution, and then apply algebraic corrections—clipping negative components and applying scaling matrices—to enforce positivity and conservation. This approach is simple and modular, requiring no modification of the base solver. It can be applied to both implicit and explicit integrators and is shown to preserve the formal order of accuracy for implicit SDIRK schemes.

The performance of the proposed method is evaluated on a diverse set of test problems: the Robertson reaction system, the mitogen-activated protein kinase (MAPK) cascade signaling network, and the stratospheric reaction. Through these experiments, we demonstrate that the correction mechanism preserves positivity and conservation even for highly stiff multiscale

problems, maintains the theoretical order of accuracy for implicit SDIRK methods, and introduces minimal computational overhead while improving solver robustness.

The remainder of this thesis is organized as follows. Chapter 2 reviews existing work on production–destruction systems, existing positivity-preserving schemes, and SDIRK methods. Chapter 3 describes the proposed correction framework and its theoretical properties. Chapter 4 describes the test problems and the numerical methods used in the experiments. Chapter 5 presents numerical experiments on a range of stiff problems, including positivity preservation, invariant preservation, order validation, and efficiency tests. Chapter 6 summarizes the main findings of this work.

Chapter 2

Review of literature on positive time integration

Numerical methods for ordinary differential equations must always preserve physical invariants such as the system's total mass, or energy levels, especially when modeling real-world processes and evolutions where these components cannot be negative. However, classical numerical methods such as Runge-Kutta can fail to preserve positivity and mass conservation when applied to stiff production-destruction systems, leading to non-physical solutions. This has led to the development of positivity preserving schemes, discussed next.

2.1 Patankar-type methods

Production–destruction systems (PDS) describe the time evolution of N interacting species through balanced production and destruction processes [2, 3]. They naturally arise in many applications, such as chemical kinetics, biochemical signaling networks, atmospheric chemistry, and ecological models. In these systems, the concentration of each species changes due to production from other species and destruction into others.

A general PDS can be written as

$$\frac{\mathrm{d}\mathbf{y}_i}{\mathrm{d}t} = P_i(\mathbf{y}) - D_i(\mathbf{y}), \quad i = 1, \ldots, N, \tag{2.1a}$$

where $P_i(\mathbf{y}) \geq 0$ and $D_i(\mathbf{y}) \geq 0$ denote the total production and destruction rates of species i.

A PDS is called *conservative* if production and destruction are pairwise balanced, i.e.,

$$P_i(\mathbf{y}) = \sum_{j=1}^{N} p_{i,j}(\mathbf{y}), \quad D_i(\mathbf{y}) = \sum_{j=1}^{N} d_{i,j}(\mathbf{y}), \quad p_{i,j}(\mathbf{y}) = d_{j,i}(\mathbf{y}), \tag{2.1b}$$

where $d_{i,j}(\mathbf{y}) \geq 0$ is the rate at which the ith constituent transforms into the jth component, while $p_{i,j}(\mathbf{y}) \geq 0$ is the corresponding production rate at which the ith component from the jth species [11].

Such pairwise balance directly implies mass conservation:

$$\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i=1}^{N} \mathbf{y}_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \mathbf{y}_i(t) = \text{const.}$$

Moreover, if the initial data $\mathbf{y}(0) \geq 0$, the exact solution of a PDS remains non-negative for all $t > 0$.

However, standard explicit and implicit time integrators may violate these structural properties, producing negative concentrations or artificial mass gain/loss when large time steps are used. To address this, Patankar-type methods were designed to guarantee unconditional positivity and mass conservation for any step size.

A significant early contribution came from Patankar-type schemes, initially developed for chemical kinetics. The Modified Patankar-Euler (MPE) and Modified Patankar-Runge-Kutta (MPRK) schemes introduced by [2] and later extended [3] guarantee unconditional positivity and conservation for stiff systems. These schemes modify destruction terms to depend linearly on the current solution, resulting in M-matrix structures that ensure non-negativity.

Based on this, Kopecz and Meister [11] derived order conditions for MPRK schemes, introducing the second-order MPRK22 family and showing that Patankar-weight denominators play an important role in improving accuracy. Their following work [13] demonstrated the non-existence of third-order three-stage MPRK schemes using standard Patankar weights. In [12], nevertheless, they introduced four-stage third order. Izgin, Thomas et al. [8] developed Lyapunov-based framework for analyzing the stability of such schemes, proving the stability of MPRK22 methods.

However, higher-order accuracy remains challenging for Patankar-type methods. Third-order accuracy cannot be achieved with only three stages under standard Patankar weights, and requires additional stages and modified denominators.

2.2 Nonstandard finite difference methods

In order to address higher order accuracy for positivity preserving schemes Martiradonna et al. [14] have developed Geometric Conservative (GeCo) and modified GeCo (mGeCo) methods, which are explicit, positive, and linear invariant conservative based on nonstandard finite difference schemes. These integrators show stability and their suitability for stiff biochemical systems. However, higher order variants such as GeCo2 show bounded stability, limiting their use in highly stiff problems [7].

2.3 Strong stability preserving methods

Recent works have focused on combining strong stability frameworks (SSPs) with the Patankar approach to handle problems that involve both convection and stiff reactive source terms. Such problems frequently arise in convection–reaction systems, chemically reacting flows,

and hyperbolic conservation laws with stiff kinetics, where maintaining positivity and mass conservation is critical even under large time-step restrictions.

Huang, Juntao et al. [5] introduced strong-stability-preserving modified Patankar–Runge–Kutta (SSPMPRK) schemes, which integrate the unconditional positivity and conservation properties of Patankar methods into the SSP Runge–Kutta framework. These schemes preserve positivity and conservation under the standard CFL conditions dictated by convection, while remaining independent of the stiffness of the source terms.

Stability is analyzed through a Lyapunov-based framework, offering parameter-dependent guarantees while maintaining high-order accuracy in convection-reaction systems.

2.4 Exponential and Magnus integrators

Other positivity preserving strategies include exponential and Magnus integrators [1]. These methods use graph Laplacian structures built-in in certain ODEs to maintain positivity and mass conservation, successfully bypassing the limitations on order typically encountered in classical methods. Their second- and third-order variants show robust performance on stiff problems while following the principles of geometric integration.

2.5 Summary

In summary, the field has evolved from low-order, robust schemes to a wide spectrum of methods ensuring accuracy, structure preservation, and efficiency. The ongoing challenge is to extend these properties to higher-order schemes applicable to complex, stiff systems without losing theoretical guarantees or practical usability.

This thesis introduces a Patankar predictor-corrector strategy that can be applied to Runge-Kutta (RK) and Singly-Diagonally Implicit Runge-Kutta (SDIRK) methods. The method consists of a two-phase correction, an initial predictor using standard RK or SDIRK steps, followed by a corrector that ensures positivity and conservation through a clip-and-scale mechanism. Stage values are clipped to eliminate negativity and then scaled using diagonal matrices that preserve the structure of systems. This method allows the application of classical time integration methods while preserving the properties required for the physical system and formal order of accuracy.

Chapter 3

Theoretical background

Many real-world dynamical systems, such as chemical reaction networks, biochemical pathways, and atmospheric models, exhibit two essential structural properties: positivity (state variables must remain non-negative for all time) and conservation laws (some linear invariants remain constant throughout the evolution).

These systems are often described as production–destruction systems, where each component can only be produced by interactions with other components or destroyed by internal decay. A characteristic of these systems is the fact that production and destruction processes balance each other out, ensuring invariant conservation. Moreover, components that are already zero cannot be further destroyed, ensuring non-negativity.

3.1 Graph Laplacian systems

To this end, we consider solving systems with a particular structure of the form [1]:

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}) = \mathbf{G}(t, \mathbf{y})\,\mathbf{y}, \quad \mathbf{y}(t_0) = \mathbf{y}_0 \succeq 0 \in \mathbb{R}^d, \tag{3.1}$$

where the symbol $\succeq 0$ denotes component-wise inequality. The matrix $\mathbf{G}(t, \mathbf{y}) \in \mathbb{R}^{d\times d}$ in (3.1) has the following properties.

Assumption 3.1 (Stability)**.** The matrix $\mathbf{G}(t, \mathbf{y})$ is stable and has eigenvalues with non-

positive real parts for any $t, \mathbf{y}$.

Assumption 3.2 (Sign entries assumption)**.** The matrix $\mathbf{G}(t, \mathbf{y})$ has non-positive diagonal entries, and non-negative off-diagonal entries:

$$\begin{cases} \mathbf{G}_{i,i}(t, \mathbf{y}) \leq 0, \quad \forall\, i = 1, \ldots, d, \\ \mathbf{G}_{i,j}(t, \mathbf{y}) \geq 0, \quad \forall\, i, j = 1, \ldots, d \;\text{ with }\; i \neq j, \end{cases} \qquad \forall\, t, \mathbf{y} \succeq 0. \tag{3.2}$$

Such systems are part of a broader class known as graph Laplacian systems [1].

The matrix $\mathbf{G}(t, \mathbf{y})$ is called a graph Laplacian if it satisfies two key structural properties:

1. **Sign structure:** Off-diagonal entries are non-negative and diagonal entries are non-positive, $\mathbf{G}_{i,j}(t, \mathbf{y}) \geq 0$ for $i \neq j$ and $\mathbf{G}_{i,i}(t, \mathbf{y}) \leq 0$. This property guarantees that any component which becomes zero cannot be further destroyed, thus ensuring non-negativity of the exact solution.

2. **Zero sum of each column:** $\sum_{i=1}^{d} \mathbf{G}_{i,j}(t, \mathbf{y}) = 0$ for all j. When this holds, the system preserves the total mass $\sum_{i=1}^{d} y_i(t) = \sum_{i=1}^{d} y_i(0)$.

Assumption 3.3 (Strong sign entries assumption)**.** The matrix $\mathbf{G}(t, \mathbf{y})$ has non-positive diagonal entries, and non-negative off-diagonal entries, for any argument:

$$\text{Entry sign property} \quad (3.2) \quad \text{holds} \quad \forall\, t, \mathbf{y}. \tag{3.3}$$

As a consequence of Assumption 3.2 (and of the stronger Assumption 3.3) the solution of

(3.1) is non-negative:

$$\begin{aligned} \mathbf{y}(t_0) \succeq 0 \;\text{ with }\; \mathbf{y}_i(t_0) = 0 \quad &\Rightarrow \quad \mathbf{y}_i'(t_0) = \sum_{j \neq i} \mathbf{G}_{i,j}\big((t_0, \mathbf{y}(t_0)\big)\, \mathbf{y}_j(t_0) \geq 0 \\ &\Rightarrow \quad \mathbf{y}(t) \succeq \mathbf{0}, \quad \forall\, t \geq t_0. \end{aligned} \tag{3.4}$$

Assumption 3.4 (Linear invariants)**.** There exist vectors $\mathbf{w}_1, \ldots, \mathbf{w}_M \in \mathbb{R}^d$ with $M < d$ such that:

$$\mathbf{w}_j^T\, \mathbf{G}(t, \mathbf{y}) = \mathbf{0}, \quad j = 1, \ldots, M, \quad \forall\, t, \mathbf{y}. \tag{3.5}$$

As a consequence of Assumption 3.4 the vectors $\mathbf{w}_j$ are linear invariants of the ODE system (3.1):

$$\mathbf{w}_j^T\, \mathbf{y}' = \mathbf{0} \quad \Rightarrow \quad \mathbf{w}_j^T\, \mathbf{y}(t) = \mathbf{w}_j^T\, \mathbf{y}_0 = \text{const}, \quad j = 1, \ldots, M, \quad \forall\, t \geq t_0. \tag{3.6}$$

Note that the zero sum of each column of graph Laplacian matrices is a linear invariant (3.5) with $\mathbf{w}_1 = \mathbf{1}_d$.

To design positivity-preserving integrators, we rely on some fundamental matrix properties.

Remark 3.5 (Positivity of the inverse)**.** For any step size $h \geq 0$, the matrix $\mathbf{I}_{d\times d} - h\,\mathbf{G}$ has positive diagonal entries, and non-positive off-diagonal entries. Moreover, the eigenvalues $\sigma(\mathbf{I}_{d\times d} - h\,\mathbf{G}) \subset \mathbb{C}^+$ due to Assumption 3.1. Consequently $\mathbf{I}_{d\times d} - h\,\mathbf{G}$ is an M-matrix [1] and

$$(\mathbf{I}_{d\times d} - h\,\mathbf{G})^{-1} \succeq \mathbf{0}.$$

Here, $\mathbf{I}_{d\times d} \in \mathbb{R}^{d\times d}$ denotes the identity matrix.

This property is proven in [1] for $M = 1$ and $\mathbf{w}_1 = \mathbf{1}_d$, where $\mathbf{1}_d \in \mathbb{R}^d$ is the column vector of all ones.

Remark 3.6 (Scaling the columns of graph Laplacian matrix)**.** Consider a diagonal matrix $\mathbf{\Sigma}$ with non-negative entries:

$$\mathbf{\Sigma} = \mathrm{diag}_{i=1,\dots,d}\, \sigma_{i,i}, \quad \sigma_{i,i} \geq 0 \quad \forall\, i. \tag{3.7a}$$

Multiplying the matrix $\mathbf{G}$ (3.1) from the right by $\mathbf{\Sigma}$ scales each column by the non-negative diagonal entry:

$$\overline{\mathbf{G}} \coloneqq \mathbf{G} \cdot \mathbf{\Sigma}, \quad \overline{\mathbf{G}}_{i,j} = \mathbf{G}_{i,j}\, \sigma_{j,j} \quad \forall\, i, j. \tag{3.7b}$$

We see immediately that the entries of the scaled matrix (3.7b) have the same signs as the entries of the original matrix (3.2),

$$\overline{\mathbf{G}}_{i,i}(t, \mathbf{y}) \leq 0, \quad \forall\, i; \quad \overline{\mathbf{G}}_{i,j}(t, \mathbf{y}) \geq 0, \quad \forall\, i \neq j. \tag{3.7c}$$

The scaled matrix (3.7b) admits the same left kernel vectors as the original matrix (3.5),

$$\mathbf{w}_j^T\, \overline{\mathbf{G}} = \mathbf{0} \quad \forall\, j. \tag{3.7d}$$

Remark 3.7 (Linear combinations of graph Laplacian matrices)**.** Let $\mathbf{G}_1, \dots, \mathbf{G}_s$ matrices whose entries have the same signs as the entries of the original matrix (3.2), and that have the same left kernel vectors (3.5). Let $b_1, \dots, b_s \geq 0$ be nonnegative numbers. Then the linear combination

$$\overline{\mathbf{G}} = b_1\, \mathbf{G}_1 + \cdots + b_s\, \mathbf{G}_s \tag{3.8}$$

has the entry sign property (3.2) and shares the same left kernel vectors (3.5). Consequently, a nonnegative linear combination of graph Laplacian matrices (3.8) is itself a graph Laplacian matrix.

Remarks 3.6 and 3.7 lead to the following.

Remark 3.8 (Linear combinations of scaled graph Laplacian matrices)**.** Let $\mathbf{G}_1, \ldots, \mathbf{G}_s$ be graph Laplacian matrices, with the same left kernel vectors (3.5). Let $b_1, \ldots, b_s \geq 0$ be nonnegative numbers. Let $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}s$ be diagonal scaling matrices with non-negative diagonal entries (3.7a). Then the nonnegative linear combination of the scaled matrices

$$\overline{\mathbf{G}} = b_1 \, \mathbf{G}_1 \, \mathbf{\Sigma}_1 + \cdots + b_s \, \mathbf{G}_s \, \mathbf{\Sigma}_s \tag{3.9}$$

is itself a graph Laplacian matrix with the same left kernel vectors (3.5).

3.2 Production-destruction systems

Many systems can be written explicitly as conservative production–destruction models (2.1), where the production rates $p_{i,j}$ and destruction rates $d_{i,j}$ have the following properties:

$$p_{i,j}(\mathbf{y}) \equiv d_{j,i}(\mathbf{y}), \qquad \forall i, j = 1, \ldots, N, \quad \forall \mathbf{y} \in \mathbb{R}^d, \tag{3.10a}$$

$$d_{i,j}(\mathbf{y}) = \ell_{i,j}(\mathbf{y}) \, \mathbf{y}_i, \quad \forall i, j = 1, \ldots, N, \quad \forall \mathbf{y} \in \mathbb{R}^d, \tag{3.10b}$$

$$\ell_{i,j}(\mathbf{y}) \geq 0, \qquad \forall i, j = 1, \ldots, N, \quad \forall \mathbf{y} \succeq 0. \tag{3.10c}$$

The coefficients $\ell_{i,j}(\mathbf{y}) \geq 0 \; \forall \mathbf{y} \succeq 0$ are non-negative transition rates, meaning that destruction terms can never generate mass or become negative. In particular, if $\mathbf{y}_i = 0$, then $d_{i,j}(\mathbf{y}) = 0 \; \forall \mathbf{y}_{j \neq i}$, so a vanishing species cannot be further destroyed.

Such systems have the detailed form [1]:

$$\begin{aligned}
\mathbf{y}_i' &= \sum_{j=1}^{N} p_{i,j}(\mathbf{y}) - \sum_{j=1}^{N} d_{i,j}(\mathbf{y}) \\
&\overset{(3.10a)}{=} \sum_{j=1}^{N} d_{j,i}(\mathbf{y}) - \sum_{j=1}^{N} d_{i,j}(\mathbf{y}) \\
&\overset{(3.10b)}{=} \sum_{j=1}^{N} \ell_{j,i}\,\mathbf{y}_j - \sum_{j=1}^{N} \ell_{i,j}\,\mathbf{y}_i.
\end{aligned} \tag{3.11}$$

Using the matrix of non-negative transition rates

$$\mathbf{L}(\mathbf{y}) := \big[\ell_{i,j}(\mathbf{y})\big]_{1\le i,j\le N}$$

the system (3.11) can be written in matrix form as

$$\begin{aligned}
\mathbf{y}' &= \mathbf{L}^T\,\mathbf{y} - \operatorname{diag}(\mathbf{L}\,\mathbf{1}_d)\,\mathbf{y} = \big(\mathbf{L}^T - \operatorname{diag}(\mathbf{L}\,\mathbf{1}_d)\big)\,\mathbf{y} \\
&= \mathbf{G}(\mathbf{y})\,\mathbf{y} \quad \text{with} \quad \mathbf{G}(\mathbf{y}) := \mathbf{L}^T - \operatorname{diag}(\mathbf{L}\,\mathbf{1}_d).
\end{aligned} \tag{3.12}$$

Note that

$$\mathbf{1}_d^T\,\mathbf{G}(\mathbf{y}) \equiv \mathbf{0} \quad \Rightarrow \quad \mathbf{1}_d^T\,\mathbf{y}(t) = \text{const}$$

From the definition (3.12) the matrix $\mathbf{G}$ has the following properties: ,

$$\begin{aligned}
\mathbf{G}_{i,i} &= \big(\mathbf{L}^T - \operatorname{diag}(\mathbf{L}\,\mathbf{1}_d)\big)_{i,i} = \mathbf{L}_{i,i} - \sum_{j=1}^{N} \mathbf{L}_{i,j} && \text{(3.13a)} \\
&= -\sum_{j\ne i} \mathbf{L}_{i,j} \le 0, \quad \forall\, i, \\
\mathbf{G}_{i,j} &= \mathbf{L}_{j,i} \ge 0 \quad \forall\, j \ne i. && \text{(3.13b)}
\end{aligned}$$

Thus, all diagonal entries are non-positive because they represent the total destruction rate

of species i taken with a negative sign, while all off-diagonal entries are non-negative because they correspond to production rates from other species into i. Hence, the sign structure in Equation (3.2) is satisfied.

Moreover,

$$\begin{aligned} \sum_{i=1}^{N} \mathbf{G}_{i,j} &= \mathbf{G}_{i,i} + \sum_{i \neq j} \mathbf{G}_{i,j} && \text{(3.13c)} \\ &= -\sum_{j \neq i} \mathbf{L}_{i,j} + \sum_{i \neq j} \mathbf{L}_{j,i} = 0, && \text{(3.13d)} \\ \Leftrightarrow \qquad & \mathbf{1}_d^T \, \mathbf{G} \equiv \mathbf{0}_d^T. && \text{(3.13e)} \end{aligned}$$

Therefore, $\mathbf{G}(\mathbf{y})$ satisfies the zero column sum property. So, by the argument in Equation (3.4), no negative values can appear in the exact solution since:

$$\mathbf{y}_i(t) = 0 \quad \Rightarrow \quad \mathbf{y}_i'(t) = \sum_{j \neq i} \mathbf{G}_{i,j} \, \mathbf{y}_j \geq 0.$$

By Assumptions 3.4 and (3.6), the total mass is conserved:

$$\frac{\mathrm{d}}{\mathrm{d}t}\big(\mathbf{1}_d^T \mathbf{y}(t)\big) = \mathbf{1}_d^T \mathbf{y}'(t) = \mathbf{1}_d^T \mathbf{G}(\mathbf{y}) \, \mathbf{y}(t) = 0.$$

Thus, production–destruction systems (3.11) are a special case of nonlinear graph Laplacian systems that satisfy the positivity and conservation properties required for Assumption 3.2.

3.3 The new positivity preserving predictor-corrector Runge-Kutta methods

We now seek to solve numerically the system (3.1) that satisfies Assumption 3.2 (or the stronger Assumption 3.3). Assume that we have computed a numerical approximation $\mathbf{y}_n \succeq 0$ for the exact solution $\mathbf{y}(t_n) \succeq 0$.

3.3.1 Clipping and scaling operations

The next step solution is computed by a singly diagonally implicit Runge-Kutta (SDIRK) method with only non-negative weights $b_i \geq 0$ as follows:

$$\mathbf{Y}_i = \mathbf{y}_n + h \sum_{j=1}^{i} a_{i,j}\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{Y}_j, \quad i = 1, \ldots, s; \tag{3.14a}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{s} b_j\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{Y}_j. \tag{3.14b}$$

Consider a *local* solution $u(t)$ of the system (3.1) started from the numerical value $u(t_n) = \mathbf{y}_n$. Since the SDIRK scheme (3.14) has stage order $q = 1$ we have

$$\mathbf{Y}_i + \mathcal{O}(h^{q+1}) = u(t_n + c_i h) \succeq 0,$$

and using the order p of scheme (3.14) yields

$$\mathbf{y}_{n+1} + \mathcal{O}(h^{p+1}) = u(t_{n+1}) \succeq 0.$$

Thus any possible negative values are of small size, i.e., the negative parts of the stage and solution values are:

$$\mathbf{Y}_i^- = \mathcal{O}(h^{q+1}), \qquad \mathbf{y}_{n+1}^- = \mathcal{O}(h^{p+1}). \tag{3.15}$$

We start by defining special clipping and scaling operations.

Definition 3.9 (Clipping)**.** Let $\mathbf{Y} \in \mathbb{R}^d$. The following clipping operation sets the negative entries of a vector to zero:

$$\mathring{\mathbf{Y}} = \text{clip}(\mathbf{Y}), \qquad \mathring{\mathbf{Y}}_\ell = \max(\mathbf{Y}_\ell, 0), \quad \ell = 1, \ldots, d. \tag{3.16}$$

Using (3.15) we see that:

$$\mathring{\mathbf{Y}}_i - \mathbf{Y}_i = \varepsilon + \mathcal{O}(h^{q+1}), \qquad \mathring{\mathbf{y}}_{n+1} - \mathbf{y}_{n+1} = \varepsilon + \mathcal{O}(h^{p+1}). \tag{3.17}$$

Figure 3.1 illustrates how clipping does *not* increase the local errors.

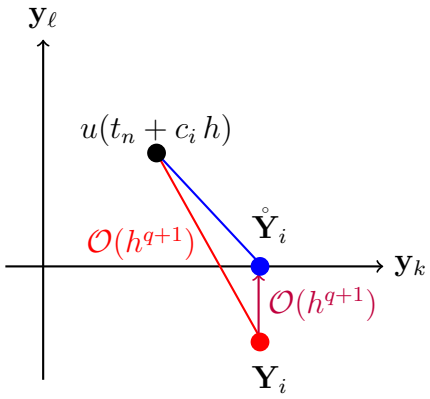


Figure 3.1: In this cartoon the ℓ-th component of the exact solution is close to zero, $u_\ell(t) \approx 0$, and the numerical stage solution has a negative ℓ-th component. Clipping changes only component ℓ of the solution. Since $u_\ell \geq 0$, $(\mathring{\mathbf{Y}}_i)_\ell = 0$, and $(\mathbf{Y}_i)_\ell \leq 0$ we see that clipping does not increase the local truncation error: $\|\mathring{\mathbf{Y}}_i - u(t_n + c_i\, h)\| \leq \|\mathbf{Y}_i - u(t_n + c_i\, h)\|$ and $\|\mathbf{Y}_i - \mathring{\mathbf{Y}}_i\| \leq \|\mathbf{Y}_i - u(t_n + c_i\, h)\|$.

Definition 3.10 (Ratio scaling)**.** Let $\mathbf{Y}, \mathbf{Z} \in \mathbb{R}^d$. The ratio scaling matrix is a positive semidefinite diagonal matrix, whose diagonal entries are the result of vector component-wise division operation:

$$\mathbf{\Sigma}_{\mathbf{Y}/\mathbf{Z}} \coloneqq \operatorname*{diag}_{\ell=1,\dots,d} \frac{\max(\mathbf{Y}_\ell, 0)}{\max(\mathbf{Z}_\ell, \varepsilon)} \in \mathbb{R}^{d\times d}. \tag{3.18}$$

The small fixed number $\varepsilon > 0$ in denominator is needed to avoid division by a number close to zero.

Remark 3.11 (Application of ratio scaling)**.** We will use Definition 3.10 with scaling vectors $\mathbf{Y}$ and $\mathbf{Z}$ that are two numerical solutions in the interval $[t_n, t_{n+1}]$, and consequently $\mathbf{Z} = \mathbf{Y} + \mathcal{O}(h)$. Let ℓ be a component that comes close to zero, and we assume that all stage and solution values have components ℓ close to zero, $\mathbf{Y}_\ell \sim 0$ and $\mathbf{Z}_\ell \sim 0$.

Let

$$\widehat{\mathbf{Y}} = \mathbf{\Sigma}_{\mathbf{Y}/\mathbf{Z}}\, \mathring{\mathbf{Z}} \in \mathbb{R}^d, \quad \widehat{\mathbf{Y}}_\ell = \begin{cases} \mathring{\mathbf{Y}}_\ell, & \mathbf{Z}_\ell \geq \varepsilon, \\ \mathring{\mathbf{Y}}_\ell \, \frac{\mathbf{Z}_\ell}{\varepsilon}, & 0 < \mathbf{Z}_\ell < \varepsilon, \\ 0, & \mathbf{Z}_\ell \leq 0. \end{cases} \tag{3.19}$$

The application of scaling on the same vector has the following effect.

1. For components that are not close to zero both $\mathbf{Z}_\ell$ and $\mathbf{Y}_\ell$ are $\mathcal{O}(1)$. Consequently $\mathbf{Z}_\ell > \varepsilon$ and $\widehat{\mathbf{Y}}_\ell = \mathring{\mathbf{Y}}_\ell = \mathbf{Y}_\ell$, such components are not changed.

2. For components ℓ that are close to zero, if either $\mathbf{Z}_\ell \leq 0$ or $\mathbf{Y}_\ell \leq 0$ then $\widehat{\mathbf{Y}}_\ell = 0$.

3. If both components are small and positive, $0 < \mathbf{Z}_\ell < \varepsilon$ and $\mathbf{Y}_\ell < \varepsilon + \mathcal{O}(h)$, then the output is a scaled value of the small $\mathbf{Y}_\ell$ component, with a multiplier smaller than one:

$$0 \leq \frac{\mathbf{Z}_\ell}{\varepsilon} < 1, \quad \widehat{\mathbf{Y}}_\ell = \frac{\mathbf{Z}_\ell}{\varepsilon}\, \mathbf{Y}_\ell,$$

and

$$\widehat{\mathbf{Y}}_\ell - \mathring{\mathbf{Y}}_\ell = (1 + \mathcal{O}(h^{q+1}))\,\mathbf{Y}_\ell = (1 + \mathcal{O}(h^{q+1}))\,(\mathbf{Z}_\ell + \mathcal{O}(h)) = \mathcal{O}(\varepsilon) + \mathcal{O}(h).$$

3.3.2 Predictor-corrector approach

Consider now the SDIRK method (3.14) with only non-negative weights $b_i \geq 0$ applied to solve (3.1) and give a "predicted" solution $\mathbf{y}_{n+1}^{\{p\}}$:

$$\mathbf{Y}_i = \mathbf{y}_n + h \sum_{j=1}^{i} a_{i,j}\,\mathbf{G}(\mathbf{Y}_j)\,\mathbf{Y}_j, \quad i = 1, \ldots, s; \tag{3.20a}$$

$$\mathbf{y}_{n+1}^{\{p\}} = \mathbf{y}_n + h \sum_{j=1}^{s} b_j\,\mathbf{G}(\mathbf{Y}_j)\,\mathbf{Y}_j. \tag{3.20b}$$

The proposed predictor-corrector algorithm proceeds as follows.

1. *Predictor step.* We apply the regular SDIRK step (3.20). The computed stage values $\mathbf{Y}_i$, including the new predicted solution $\mathbf{y}_{n+1}^{\{p\}}$, may have negative entries. Note that if in (3.20) all stages $\mathbf{Y}_j \succeq 0$ have non-negative entries, and the solution $\mathbf{y}_{n+1}^{\{p\}} \succeq \varepsilon$ has entries greater than the threshold, then the predictor solution (3.20b) reads:

$$\begin{aligned}
\mathbf{y}_{n+1}^{\{p\}} &= \mathbf{y}_n + h \left(\sum_{j=1}^{s-1} b_j\,\mathbf{G}(\mathbf{Y}_j)\,\boldsymbol{\Sigma}_{\mathbf{Y}_j/\mathbf{y}_{n+1}^{\{p\}}} + b_s\,\mathbf{G}(\mathbf{y}_{n+1}^{\{p\}}) \right) \mathbf{y}_{n+1}^{\{p\}} \\
&= \mathbf{y}_n + h\,\bar{\bar{\mathbf{G}}}\,\mathbf{y}_{n+1}^{\{p\}}, \\
\mathbf{y}_{n+1}^{\{p\}} &= \left(\mathbf{I}_{d\times d} - h\,\bar{\bar{\mathbf{G}}}\right)^{-1} \mathbf{y}_n.
\end{aligned} \tag{3.21}$$

   Under these assumptions $\boldsymbol{\Sigma}_{\mathbf{Y}_j/\mathbf{y}_{n+1}^{\{p\}}}$ are matrices of nonnegative weights, and $\bar{\bar{\mathbf{G}}}$ is a graph Laplacian matrix per Corollary 3.8. But this would be a circular argument.

The goal of the corrector is to construct a modified matrix $\overline{\mathbf{G}}$ that satisfies Assumptions even when some stage or solution entries are negative, and compute a corrected solution $\left(\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}}\right)^{-1}\,\mathbf{y}_n$.

2. *Corrector step.* We compute a new, positive solution as follows:

$$\begin{aligned}
\mathring{\mathbf{Y}}_j &= \mathrm{clip}(\mathbf{Y}_j) \quad \text{(clipped version of stage computed by (3.20))} \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + h\left(\sum_{j=1}^{s} b_j\,\mathbf{G}(\mathring{\mathbf{Y}}_j)\,\mathbf{\Sigma}_{\mathbf{Y}_j/\mathbf{y}_{n+1}^{\{p\}}}\right)\mathbf{y}_{n+1} \\
&= \mathbf{y}_n + h\,\overline{\mathbf{G}}\,\mathbf{y}_{n+1},
\end{aligned} \tag{3.22}$$

where

$$\overline{\mathbf{G}} = \sum_{j=1}^{s} b_j\,\mathbf{G}(\mathring{\mathbf{Y}}_j)\,\mathbf{\Sigma}_{\mathbf{Y}_j/\mathbf{y}_{n+1}^{\{p\}}}. \tag{3.23}$$

Note that the matrix $\overline{\mathbf{G}}$ satisfies the assumptions posed in eqs. (3.2) and (3.5) and (3.2) needed for positivity and conservation. Each $\mathbf{G}(\mathring{\mathbf{Y}}_j)$ is evaluated at clipped arguments with non-negative entries. Each $\mathbf{G}(\mathring{\mathbf{Y}}_j)$ is scaled by a diagonal matrix with non-negative entries $\mathbf{\Sigma}_{\mathring{\mathbf{Y}}_j/\mathring{\mathbf{y}}_{n+1}^{\{p\}}}$, applied from the right. The corrector solution is computed via (3.22):

$$\mathbf{y}_{n+1} = \left(\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}}\right)^{-1}\,\mathbf{y}_n,$$

and is both non-negative and conservative.

After the predictor step, the intermediate stage values $\mathbf{Y}_j$ may contain negative components. This happens because the original SDIRK update evaluates the nonlinear matrix $\mathbf{G}$ at $\mathbf{Y}_j$, which are not guaranteed to remain positive. Therefore, the matrix $\bar{\bar{\mathbf{G}}}$ may lose the structural properties in (3.2), breaking the proof of positivity.

The corrector step restores these properties in four stages:

1. Clipping restores the sign structure.

   Each stage is replaced by its clipped version:

$$\mathring{\mathbf{Y}}_j = \mathrm{clip}(\mathbf{Y}_j) \succeq 0.$$

   Since the arguments are now non-negative, each $\mathbf{G}(\mathring{\mathbf{Y}}_j)$ satisfies the required sign pattern:

$$\mathbf{G}_{i,i}(\mathring{\mathbf{Y}}_j) \leq 0, \qquad \mathbf{G}_{i,j}(\mathring{\mathbf{Y}}_j) \geq 0 \ (i \neq j).$$

   Note that if the strong sign entries Assumption 3.3 holds, then arguments need not be clipped to preserve the sign property, and (3.23) takes the simpler form:

$$\overline{\mathbf{G}} = \sum_{j=1}^{s} b_j \, \mathbf{G}(\mathbf{Y}_j) \, \mathbf{\Sigma}_{\mathbf{Y}_j / \mathbf{y}_{n+1}^{\{p\}}}. \tag{3.24}$$

2. Column scaling preserves invariants.

   Each matrix $\mathbf{G}(\mathring{\mathbf{Y}}_j)$ is right-multiplied by the non-negative diagonal scaling matrix $\mathbf{\Sigma}_{\mathbf{Y}_j / \mathbf{y}_{n+1}^{\{p\}}}$. By Remark 3.6, such column scaling preserves the sign pattern, and the same left kernel vectors $\mathbf{w}_j^\top \mathbf{G} = 0$.

3. Linear combinations also preserve invariants.

   The corrected matrix is a convex combination:

$$\overline{\mathbf{G}} = \sum_{j=1}^{s} b_j \, \mathbf{G}(\mathring{\mathbf{Y}}_j) \, \mathbf{\Sigma}_{\mathbf{Y}_j / \mathbf{y}_{n+1}^{\{p\}}},$$

   where all weights $b_j \geq 0$. By Remark 3.7, any non-negative linear combination of matrices that individually satisfy the sign structure and share the same left kernel

vectors also satisfies:

$$\overline{\mathbf{G}}_{i,i} \leq 0, \quad \overline{\mathbf{G}}_{i,j} \geq 0, \quad \mathbf{w}_j^\top \overline{\mathbf{G}} = 0.$$

4. Final implicit update preserves positivity and conserves linear invariants.

   The corrected solution is computed as

$$\mathbf{y}_{n+1} = \left(\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}}\right)^{-1} \mathbf{y}_n,$$

   Since $\overline{\mathbf{G}}$ satisfies Assumption 3.1, the matrix $\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}}$ is an M-matrix (positive diagonals, non-positive off-diagonals, eigenvalues in the open right half-plane). From classical M-matrix theory, its inverse is non-negative:

$$(\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}})^{-1} \succeq 0.$$

   Furthermore, since $\mathbf{w}_j^\top \overline{\mathbf{G}} = 0$, multiplying the corrector step by $\mathbf{w}_j^\top$ yields:

$$\mathbf{w}_j^\top \mathbf{y}_{n+1} = \mathbf{w}_j^\top (\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}})^{-1} \mathbf{y}_n = \mathbf{w}_j^\top \mathbf{y}_n,$$

   thus preserving all linear invariants exactly.

By enforcing non-negative clipping before recomputing the stage matrix, and by using only column scalings and convex combinations (which preserve the same invariant subspace), the corrected matrix $\overline{\mathbf{G}}$ satisfies all structural properties.

Therefore, the final update

$$\mathbf{y}_{n+1} = \left(\mathbf{I}_{d\times d} - h\,\overline{\mathbf{G}}\right)^{-1} \mathbf{y}_n,$$

is guaranteed to be non-negative (due to the M-matrix inverse being non-negative) and

conservative (due to invariants $\mathbf{w}_j^\top \overline{\mathbf{G}} = 0$ being preserved).

3.4 Patankar stage predictor-corrector DIRK methods

We now apply the same approach to make all stage vectors non-negative.

We consider a stiffly accurate (S)DIRK method with non-negative coefficients $a_{i,j} \geq 0$ and weights $b_i \geq 0$ applied to solve (3.1):

$$\begin{aligned} \mathbf{Y}_i &= \mathbf{y}_n + h \sum_{j=1}^{i} a_{i,j}\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{Y}_j, \quad i = 1, \dots, s; \\ \mathbf{y}_{n+1} = \mathbf{Y}_s &= \mathbf{y}_n + h \sum_{j=1}^{s} a_{s,j}\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{Y}_j \\ &\equiv \mathbf{y}_n + h \sum_{j=1}^{s} b_j\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{Y}_j. \end{aligned} \tag{3.25}$$

Stiff accuracy means that the last stage is the next step solution, i.e., $b_j = a_{s,j} \geq 0$ for all j.

1. *Predictor step.* We apply the regular DIRK stage (3.25):

$$\mathbf{Y}_i^{\{p\}} = \mathbf{y}_n + h \sum_{j=1}^{i-1} a_{i,j}\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{Y}_j + h\, a_{i,i}\, \mathbf{G}(\mathbf{Y}_i^{\{p\}})\, \mathbf{Y}_i^{\{p\}}.$$

   We note that, if all predicted entries are positive $\mathbf{Y}_i^{\{p\}} \succeq \varepsilon \mathbf{1}_d$, then:

$$\mathbf{Y}_i^{\{p\}} = \mathbf{y}_n + h \left(\sum_{j=1}^{i-1} a_{i,j}\, \mathbf{G}(\mathbf{Y}_j)\, \mathbf{\Sigma}_{\mathbf{Y}_j / \mathbf{Y}_i^{\{p\}}} + h\, a_{i,i}\, \mathbf{G}(\mathbf{Y}_i^{\{p\}}) \right) \mathbf{Y}_i^{\{p\}}.$$

   However, the computed predictor stage values $\mathbf{Y}_i^{\{p\}}$ may have negative entries.

2. *Corrector step.* We compute a new, positive stage vector as follows.

$$\mathbf{Y}_i = \mathbf{y}_n + h \left(\sum_{j=1}^{i-1} a_{i,j}\, \mathbf{G}(\mathbf{Y}_j)\, \boldsymbol{\Sigma}_{\mathbf{Y}_j / \mathring{\mathbf{Y}}_i^{\{p\}}} + h\, a_{i,i}\, \mathbf{G}(\mathring{\mathbf{Y}}_i^{\{p\}}) \right) \mathbf{Y}_i. \tag{3.26}$$

By the same reasoning as before, each corrected stage $\mathbf{Y}_i \succeq \mathbf{0}$ is a conservative and positive vector.

3. Using the stiff accuracy property, the new solution is the last corrected stage $\mathbf{y}_{n+1} = \mathbf{Y}_s$.

Example. Consider the order 2 SDIRK2 method:

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{y}_n + h\,\gamma\,\mathbf{G}(\mathbf{Y}_1)\,\mathbf{Y}_1 \\ \mathbf{Y}_2 &= \mathbf{y}_n + h\,(1-\gamma)\,\mathbf{G}(\mathbf{Y}_1)\,\mathbf{Y}_1 + h\,\gamma\,\mathbf{G}(\mathbf{Y}_2)\,\mathbf{Y}_2 \\ \mathbf{y}_{n+1} &= \mathbf{Y}_2, \end{aligned} \tag{3.27}$$

with $\gamma = 1 - \frac{\sqrt{2}}{2}$ and where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are the stages of SDIRK2 method.

1. *Predictor step.* The regular SDIRK2 stages are computed as follows (3.25):

$$\begin{aligned} \mathbf{Y}_1^{\{p\}} &= \mathbf{y}_n + h\,\gamma\,\mathbf{G}(\mathbf{Y}_1^{\{p\}})\,\mathbf{Y}_1^{\{p\}} \\ \mathbf{Y}_2^{\{p\}} &= \mathbf{y}_n + h\,(1-\gamma)\,\mathbf{G}(\mathbf{Y}_1^{\{p\}})\,\mathbf{Y}_1^{\{p\}} + h\,\gamma\,\mathbf{G}(\mathbf{Y}_2^{\{p\}})\,\mathbf{Y}_2^{\{p\}}, \end{aligned} \tag{3.28}$$

where the predicted stages $\mathbf{Y}_1^{\{p\}}$ and $\mathbf{Y}_2^{\{p\}}$ might contain negative components.

2. *Corrector step.* To enforce positivity, we apply clipping (3.16) and scaling operations

(3.26):

$$\begin{aligned}
\mathring{\mathbf{Y}}_1^{\{p\}} &= \mathrm{clip}(\mathbf{Y}_1^{\{p\}}) \\
\mathbf{Y}_1 &= \mathbf{y}_n + h\,\gamma\,\mathbf{G}(\mathring{\mathbf{Y}}_1^{\{p\}})\,\mathbf{Y}_1 \\
\mathring{\mathbf{Y}}_2^{\{p\}} &= \mathrm{clip}(\mathbf{Y}_2^{\{p\}}) \\
\mathbf{Y}_2 &= \mathbf{y}_n + h\,(1-\gamma)\,\mathbf{G}(\mathbf{Y}_1)\,\boldsymbol{\Sigma}_{\mathbf{Y}_1/\mathring{\mathbf{Y}}_2^{\{p\}}}\,\mathbf{Y}_2 + h\,\gamma\,\mathbf{G}(\mathring{\mathbf{Y}}_2^{\{p\}})\,\mathbf{Y}_2.
\end{aligned} \tag{3.29}$$

Each corrected stage $\mathbf{Y}_1,\ \mathbf{Y}_2 \succeq \mathbf{0}$ is now guaranteed to have non-negative entries.

3. *Final solution.* The corrected solution is obtained directly from the last stage:

$$\mathbf{y}_{n+1} = \mathbf{Y}_2. \tag{3.30}$$

This ensures both positivity and conservation properties of the numerical method.

3.5 Accuracy considerations

Consider the predicted solution (3.20b) and the Patankar corrector (3.22) started from the exact solution $\mathbf{y}_n = \mathbf{y}(t_n)$:

$$\begin{aligned}
\mathbf{y}_{n+1}^{\{p\}} &= \mathbf{y}_n + h \sum_{j=1}^{s} b_j\,\mathbf{G}(\mathbf{Y}_j)\,\mathbf{Y}_j && \text{(3.31a)} \\
&= \mathbf{y}_n + h \sum_{j=1}^{s} b_j\,\mathbf{G}(\mathbf{Y}_j)\,\mathrm{diag}(\mathbf{Y}_j)\,\mathbf{1}_d, \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + h \Big(\sum_{j=1}^{s} b_j\,\mathbf{G}(\mathring{\mathbf{Y}}_j)\,\boldsymbol{\Sigma}_{\mathbf{Y}_j/\mathbf{y}_{n+1}^{\{p\}}} \Big)\,\mathbf{y}_{n+1} && \text{(3.31b)} \\
&= \mathbf{y}_n + h \sum_{j=1}^{s} b_j\,\mathbf{G}(\mathring{\mathbf{Y}}_j)\,\mathrm{diag}(\mathring{\mathbf{Y}}_j)\,\boldsymbol{\Sigma}_{\mathring{\mathbf{Y}}_j/\mathring{\mathbf{y}}_{n+1}^{\{p\}}}\,\mathbf{y}_{n+1},
\end{aligned}$$

We apply Remark 3.11 to study the accuracy of the Patankar corrector.

Let $\delta_j = \|\mathbf{Y}_j^-\|$ be the size of the negative part of the stage vector, and $\delta = \max_j \delta_j$. From our previous discussion we know that $\delta = \mathcal{O}(h^{q+1})$, where q is the stage order of the scheme.

From (3.31a) we have:

$$\begin{aligned}
\max(\mathbf{y}_{n+1}^{\{p\}}, \varepsilon) &= \mathbf{y}_n + h \sum_{j=1}^{s} b_j \, \mathbf{G}(\mathbf{Y}_j) \operatorname*{diag}_{\ell=1,\dots,d} ([\mathbf{Y}_j]_\ell) \; \mathbf{1}_d + \mathcal{O}(\varepsilon) \\
&= \mathbf{y}_n + h \sum_{j=1}^{s} b_j \, \mathbf{G}(\mathring{\mathbf{Y}}_j) \operatorname*{diag}_{\ell=1,\dots,d} \left([\mathring{\mathbf{Y}}_j]_\ell \right) \; \mathbf{1}_d + \mathcal{O}(\varepsilon) + \mathcal{O}(h\,\delta) \\
&= \mathbf{y}_n + h \sum_{j=1}^{s} b_j \, \mathbf{G}(\mathring{\mathbf{Y}}_j) \operatorname*{diag}_{\ell=1,\dots,d} \left([\mathring{\mathbf{Y}}_j]_\ell \, \frac{\max\left([\mathbf{y}_{n+1}^{\{p\}}]_\ell, \varepsilon\right)}{\max\left[(\mathbf{y}_{n+1}^{\{p\}}]_\ell, \varepsilon\right)} \right) \; \mathbf{1}_d \\
&\quad + \mathcal{O}(\varepsilon) + \mathcal{O}(h\,\delta) \\
&= \mathbf{y}_n + h \left(\sum_{j=1}^{s} b_j \, \mathbf{G}(\mathring{\mathbf{Y}}_j) \, \mathbf{\Sigma}_{\mathbf{Y}_j / \mathbf{y}_{n+1}^{\{p\}}} \right) \max(\mathbf{y}_{n+1}^{\{p\}}, \varepsilon) \\
&\quad + \mathcal{O}(\varepsilon) + \mathcal{O}(h\,\delta) \\
&= \left(\mathbf{I} - h\,\overline{\mathbf{G}} \right) \mathbf{y}_n + \mathcal{O}(\varepsilon) + \mathcal{O}(h\,\delta) \\
&= \mathbf{y}_{n+1} + \mathcal{O}(\varepsilon) + \mathcal{O}(h\,\delta).
\end{aligned}$$

Since $\mathbf{y}_{n+1}^{\{p\}} - \mathbf{y}(t_{n+1}) = \mathcal{O}(h^{p+1})$, we have that $\mathbf{y}_{n+1} - \mathbf{y}(t_{n+1}) = \mathcal{O}(h^{\min(q+2,p+1)}) + \mathcal{O}(\varepsilon)$. This worst-case analysis shows that the order of the Patankar corrected method is $\min(p, q+1)$.

Remark 3.12. However, in many practical situations a better empirical order can be expected, if the stage negative parts are of size $\delta = \mathcal{O}(h^{q+1})$ with a very small leading constant.

Remark 3.13. For SDIRK schemes considered in the numerical experiment we have $q = 1$, and therefore the Patankar correction reduces to second order in the worst case. One possible solution is to consider base schemes with higher stage order, e.g., ESDIRK schemes with $q = 2$ or FIRK schemes with larger q.

Chapter 4

Test problems and methods

To assess the proposed positivity-preserving correction strategy, we employed four representative stiff problems: the Robertson reaction, the MAPK cascade, and the stratospheric reaction system. Each of these models can be expressed in the general graph Laplacian form:

$$\mathbf{y}' = \mathbf{G}(\mathbf{y})\mathbf{y}$$

where $\mathbf{G}(\mathbf{y})$ satisfies the sign-pattern of a graph Laplacian [1], i.e., off-diagonal elements are nonnegative, diagonal elements are nonpositive, and (in some cases) columns sum to zero, ensuring positivity and possibly mass preservation.

4.1 Test Problems

4.1.1 Robertson Reaction System

The classical Robertson reaction models a stiff three-species chemical network [1]:

$$A \to B, \quad B + B \to B + C \to A + C,$$

which yields the following system of ODEs for the concentrations $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$:

$$\mathbf{y}_1' = -0.04\,\mathbf{y}_1 + 10^4\,\mathbf{y}_2\,\mathbf{y}_3, \tag{4.1}$$

$$\mathbf{y}_2' = 0.04\,\mathbf{y}_1 - 10^4\,\mathbf{y}_2\,\mathbf{y}_3 - 3\cdot 10^7\,\mathbf{y}_2^2, \tag{4.2}$$

$$\mathbf{y}_3' = 3\cdot 10^7\,\mathbf{y}_2^2. \tag{4.3}$$

This can be written compactly in the Laplacian form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\end{bmatrix} = \underbrace{\begin{bmatrix}-0.04 & 10^4\,\mathbf{y}_3 & 0\\ 0.04 & -3\cdot 10^7\,\mathbf{y}_2 - 10^4\,\mathbf{y}_3 & 0\\ 0 & 3\cdot 10^7\,\mathbf{y}_2 & 0\end{bmatrix}}_{\mathbf{G}(\mathbf{y})}\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\end{bmatrix}.$$

The initial conditions are:

$$\mathbf{y}(0) = (1, 0, 0)^\top.$$

Here, $\mathbf{G}(\mathbf{y})$ has the graph Laplacian structure, which guarantees that the exact solution remains non-negative and satisfies a linear invariant corresponding to mass conservation [1].

In particular, the invariant is

$$\mathbf{w}^\top\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2 + \mathbf{y}_3 = \text{constant}.$$

This means that the total mass of all species is preserved by the continuous system, even though the individual species concentrations evolve over multiple time scales.

4.1.2 MAPK Cascade Model

The mitogen-activated protein kinase (MAPK) cascade is a fundamental biochemical signaling network exhibiting multistability and oscillations. In the reduced six-dimensional form (following [1]), the system reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\\ \mathbf{y}_4\\ \mathbf{y}_5\\ \mathbf{y}_6\end{bmatrix} = \underbrace{\begin{bmatrix} -(k_7+k_1\mathbf{y}_2) & 0 & 0 & k_2 & 0 & 0\\ 0 & -k_1\mathbf{y}_1 & k_5 & 0 & 0 & 0\\ 0 & 0 & -(k_3\mathbf{y}_1+k_5) & k_2 & k_4 & 0\\ (1-\alpha)k_1\mathbf{y}_2 & \alpha k_1\mathbf{y}_1 & 0 & -k_2 & 0 & 0\\ 0 & 0 & k_3\mathbf{y}_1 & 0 & -k_4 & 0\\ k_7 & 0 & 0 & 0 & 0 & -k_6 \end{bmatrix}}_{\mathbf{G}(\mathbf{y})}\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\\ \mathbf{y}_4\\ \mathbf{y}_5\\ \mathbf{y}_6\end{bmatrix}. \tag{4.4}$$

Here, the parameter $\alpha \in [0,1]$ controls how the coupling between $\mathbf{y}_1$ and $\mathbf{y}_2$ is distributed between different reaction pathways. While $\mathbf{G}(\mathbf{y})$ does not always have column sums equal to zero, it retains the sign pattern of a Laplacian, thus preserving positivity of the exact solution [1].

The typical rate parameters used are: $k_1 = \frac{100}{3}$, $k_2 = \frac{1}{3}$, $k_3 = 50$, $k_4 = \frac{1}{2}$, $k_5 = \frac{10}{3}$, $k_6 = \frac{1}{10}$, $k_7 = \frac{7}{10}$.

The initial condition for this model is given by:$\mathbf{y}(0) = [0.1,\ 0.175,\ 0.15,\ 1.15,\ 0.81,\ 0.5]^\top$.

This MAPK cascade possesses two independent linear conservation laws (mass-like invariants):

$$\mathbf{y}_1 + \mathbf{y}_4 + \mathbf{y}_6 = \text{const}, \tag{4.5}$$

$$\mathbf{y}_2 + \mathbf{y}_3 + \mathbf{y}_4 + \mathbf{y}_5 = \text{const}. \tag{4.6}$$

These invariants correspond to the left kernel vectors

$$\mathbf{w_1}^\top = [1, 0, 0, 1, 0, 1], \quad \mathbf{w_2}^\top = [0, 1, 1, 1, 1, 0].$$

Indeed, it holds that:

$$\mathbf{w_1}^\top \mathbf{G}(\mathbf{y}) = \mathbf{0}^\top, \quad \mathbf{w_2}^\top \mathbf{G}(\mathbf{y}) = \mathbf{0}^\top.$$

While this property holds for $\alpha \in (0, 1)$, special cases behave differently. For $\alpha = 0$, $\mathbf{w_1}^\top \mathbf{G}(\mathbf{y}) = 0$ but $\mathbf{w_2}^\top \mathbf{G}(\mathbf{y}) \neq 0$. For $\alpha = 1$, $\mathbf{w_2}^\top \mathbf{G}(\mathbf{y}) = 0$ but $\mathbf{w_1}^\top \mathbf{G}(\mathbf{y}) \neq 0$.

Therefore, it is impossible to simultaneously enforce both invariants with methods that rely solely on matrix exponentials—this reflects a typical limitation in the geometric integration of nonlinear systems with multiple invariants [1]. We chose $\alpha = 1$ to ensure exact preservation of the conservation law corresponding to $\mathbf{w_2}$, which is typically more relevant for the signaling interpretation of the cascade.

4.1.3 Stratospheric Reaction System

The stratospheric reaction model represents photochemical interactions in atmospheric chemistry. The stratospheric reaction problem involves six chemical species:

$$\mathbf{y} = \left[[O^{1D}],\, [O],\, [O_3],\, [O_2],\, [NO],\, [NO_2]\right]^\top = [\mathbf{y}_1, \ldots, \mathbf{y}_6]^\top,$$

where each component represents the concentration of a specific species. The time evolution of the concentrations is governed by the following stiff system of ODEs [1]:

$$\begin{aligned}
\mathbf{y}_1' &= k_5\mathbf{y}_3 - k_6\mathbf{y}_1 - k_7\mathbf{y}_1\mathbf{y}_3,\\
\mathbf{y}_2' &= 2k_1\mathbf{y}_4 - k_2\mathbf{y}_2\mathbf{y}_4 + k_3\mathbf{y}_3 - k_4\mathbf{y}_2\mathbf{y}_3 + k_6\mathbf{y}_1 - k_9\mathbf{y}_2\mathbf{y}_6 + k_{10}\mathbf{y}_6,\\
\mathbf{y}_3' &= k_2\mathbf{y}_2\mathbf{y}_4 - k_3\mathbf{y}_3 - k_4\mathbf{y}_2\mathbf{y}_3 - k_7\mathbf{y}_1\mathbf{y}_3 - k_8\mathbf{y}_3\mathbf{y}_5,\\
\mathbf{y}_4' &= -k_1\mathbf{y}_4 - k_2\mathbf{y}_2\mathbf{y}_4 + k_3\mathbf{y}_3 + 2k_4\mathbf{y}_2\mathbf{y}_3 + k_5\mathbf{y}_3 + 2k_7\mathbf{y}_1\mathbf{y}_3 + k_8\mathbf{y}_3\mathbf{y}_5 + k_9\mathbf{y}_2\mathbf{y}_6,\\
\mathbf{y}_5' &= -k_8\mathbf{y}_3\mathbf{y}_5 + k_9\mathbf{y}_2\mathbf{y}_6 + k_{10}\mathbf{y}_6,\\
\mathbf{y}_6' &= k_8\mathbf{y}_3\mathbf{y}_5 - k_9\mathbf{y}_2\mathbf{y}_6 - k_{10}\mathbf{y}_6.
\end{aligned} \tag{4.7}$$

The rate coefficients are given by:

$$\begin{aligned}
k_1 &= 2.643 \cdot 10^{-10}\,\sigma^3(t), & k_2 &= 8.018 \cdot 10^{-17}, & k_3 &= 6.120 \cdot 10^{-4}\,\sigma(t),\\
k_4 &= 1.576 \cdot 10^{-15}, & k_5 &= 1.070 \cdot 10^{-3}\,\sigma^2(t), & k_6 &= 7.110 \cdot 10^{-11},\\
k_7 &= 1.200 \cdot 10^{-10}, & k_8 &= 6.062 \cdot 10^{-15}, & k_9 &= 1.069 \cdot 10^{-11},\\
k_{10} &= 1.289 \cdot 10^{-2}\,\sigma(t).
\end{aligned}$$

Here, $\sigma(t)$ is a periodic function modeling the diurnal cycle:

$$\sigma(t) = \begin{cases} \dfrac{1}{2} + \dfrac{1}{2}\cos\left(\pi \left|\dfrac{2T_L - T_R - T_S}{T_S - T_R}\right| \dfrac{2T_L - T_R - T_S}{T_S - T_R}\right), & \text{if } T_R \le T_L \le T_S,\\ 0, & \text{otherwise,} \end{cases}$$

where:

$$T_L = \left(\frac{t}{3600}\right) \bmod 24, \quad T_R = 4.5, \quad T_S = 19.5.$$

The time is measured in seconds. The initial time is taken at noon, $t_0 = 12 \cdot 3600$, and the

integration is typically performed over a three-day period $t_f = t_0 + 72 \cdot 3600$.

The initial concentrations are:

$$y_0 = \left[9.906 \cdot 10,\ 6.624 \cdot 10^8,\ 5.326 \cdot 10^{11},\ 1.697 \cdot 10^{16},\ 8.725 \cdot 10^8,\ 2.240 \cdot 10^8\right]^\top.$$

This is a non-autonomous system that can be written in the compact form:

$$\mathbf{y}'(t) = \mathbf{G}(t, \mathbf{y})\mathbf{y},$$

where $\mathbf{G}(t, \mathbf{y})$ is an explicitly time-dependent graph Laplacian matrix.

A possible choice for the graph Laplacian $\mathbf{G}$ is:

$$\begin{bmatrix}
-(k_6 + k_7\mathbf{y}_3) & 0 & k_5 & 0 & 0 & 0 \\
k_6 & -(k_2\mathbf{y}_4 + k_4\mathbf{y}_3 + k_9\mathbf{y}_6) & k_3 & 2k_1 & 0 & k_{10} \\
0 & \frac{1}{3}k_2\mathbf{y}_4 & -\gamma & \frac{2}{3}k_2\mathbf{y}_2 & 0 & 0 \\
\frac{1}{2}k_7\mathbf{y}_3 & k_4\mathbf{y}_3 + \frac{1}{2}k_9 y_6 & \gamma + \frac{1}{2}k_7\mathbf{y}_1 & -(k_1 + k_2\mathbf{y}_2) & 0 & \frac{1}{2}k_9\mathbf{y}_2 \\
0 & 0 & 0 & 0 & -k_8\mathbf{y}_3 & k_{10} + k_9\mathbf{y}_2 \\
0 & 0 & 0 & 0 & k_8\mathbf{y}_3 & -(k_{10} + k_9\mathbf{y}_2)
\end{bmatrix}$$

with $\gamma = k_3 + k_5 + k_4\mathbf{y}_2 + k_7\mathbf{y}_1 + k_8\mathbf{y}_5$.

This problem has two linear invariants corresponding to the conservation of the total number of oxygen and nitrogen atoms, respectively:

$$\mathbf{w_1} = [1,\ 1,\ 3,\ 2,\ 1,\ 2]^\top, \quad \mathbf{w_2} = [0,\ 0,\ 0,\ 0,\ 1,\ 1]^\top.$$

However, due to the structure of $\mathbf{G}$, it is impossible to find a single $\mathbf{G}$ that simultaneously

preserves both invariants exactly:

$$\mathbf{w_1}^\top \mathbf{G}(t, \mathbf{y}) \neq 0, \quad \mathbf{w_2}^\top \mathbf{G}(t, \mathbf{y}) = 0.$$

Thus, one invariant can be preserved exactly (here, nitrogen), while the other (oxygen) is only preserved to high accuracy numerically [1].

In summary, all three problems—Robertson reaction, MAPK cascade, and stratospheric chemistry—fit the $\mathbf{G}(\mathbf{y})\mathbf{y}$ framework of production–destruction systems, enabling the application of positivity-preserving time integrators.

4.2 Base SDIRK methods

To integrate stiff production–destruction systems, we use a family of singly diagonally implicit Runge–Kutta (SDIRK) methods with embedded error estimators.

An s-stage SDIRK method for the autonomous system

$$\mathbf{y}' = f(\mathbf{y}), \quad \mathbf{y}(t_n) = \mathbf{y}_n,$$

takes the form

$$\mathbf{Y}_i = \mathbf{y}_n + h \sum_{j=1}^{i} a_{ij}\, f(\mathbf{Y}_j), \quad i = 1, \ldots, s, \tag{4.8}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{s} b_j\, f(\mathbf{Y}_j), \tag{4.9}$$

where the Butcher matrix $A = (a_{ij})$ is lower triangular with identical diagonal entries $a_{ii} = \gamma$ [4].

We use three classical SDIRK methods of increasing order, each with an embedded scheme for adaptive step-size control.

4.2.1 SDIRK21 scheme

The SDIRK21 scheme is a two-stage, second-order, L-stable method with an embedded first-order solution for local error estimation.

Its general Butcher tableau is:

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline b_i & 1-\gamma & \gamma \\ \hline \hat{b}_i & 1-\hat{b}_2 & \hat{b}_2 \end{array} \quad \text{with } \gamma = 1 \pm \frac{1}{\sqrt{2}}.$$

where $\hat{b}_2$ is a free parameter defining the embedded lower-order method.

The specific numerical values for coefficients used in this work are:

$$\begin{array}{c|cc} 1-\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & 0 \\ 1 & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\ \hline b & \frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\ \hline \hat{b} & \frac{2}{3} & \frac{1}{3} \end{array}$$

This method is second-order accurate, L-stable, making it robust for stiff problems, and has an embedded Euler-like first-order scheme for adaptive step-size control [9].

4.2.2 SDIRK32 scheme

The SDIRK32 scheme is a four-stage, third-order, L-stable method with an embedded second-order solution.

The Butcher tableau is given by:

γ	γ	0	0	0
c_2	$(c_2-\gamma)$	γ	0	0
c_3	$(c_3-a_{32}-\gamma)$	a_{32}	γ	0
1	$(1-b_2-b_3-\gamma)$	b_2	b_3	γ
b_i	$(1-b_2-b_3-\gamma)$	b_2	b_3	γ
$\hat{b}_i$	$(1-\hat{b}_2-\hat{b}_3-\hat{b}_4)$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$

The coefficients used in this work are:

$\frac{9}{40}$	$\frac{9}{40}$	0	0	0
$\frac{7}{13}$	$\frac{163}{520}$	$\frac{9}{40}$	0	0
$\frac{11}{15}$	$-\frac{6481433}{8838675}$	$\frac{87795409}{70709400}$	$\frac{9}{40}$	0
1	$\frac{4032}{9943}$	$\frac{6929}{15485}$	$-\frac{723}{9272}$	$\frac{9}{40}$
b	$\frac{4032}{9943}$	$\frac{6929}{15485}$	$-\frac{723}{9272}$	$\frac{9}{40}$
$\hat{b}$	$\frac{20}{51}$	$-\frac{410011317313}{22571052756900}$	$-\frac{2075562131}{281561189810}$	$\frac{29595333}{2429345900}$

This method is third-order accurate, L-stable, suitable for moderately stiff problems, and has an embedded second-order method for error estimation [9].

4.2.3 SDIRK43 scheme

The SDIRK43 scheme is a five-stage, fourth-order method with an embedded third-order solution.

The Butcher tableau is:

γ	γ	0	0	0	0
c_2	$(c_2 - \gamma)$	γ	0	0	0
c_3	$(c_3 - a_{32} - \gamma)$	a_{32}	γ	0	0
c_4	$(c_4 - a_{42} - a_{43} - \gamma)$	a_{42}	a_{43}	γ	0
1	$(1 - b_2 - b_3 - b_4 - \gamma)$	b_2	b_3	b_4	γ
b_i	$(1 - b_2 - b_3 - b_4 - \gamma)$	b_2	b_3	b_4	γ
$\hat{b}_i$	$(1 - \hat{b}_2 - \hat{b}_3 - \hat{b}_4 - \hat{b}_5)$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$

With numerical values:

$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{9}{10}$	$\frac{13}{20}$	$\frac{1}{4}$	0	0	0
$\frac{2}{3}$	$\frac{580}{1287}$	$-\frac{175}{5148}$	$\frac{1}{4}$	0	0
$\frac{3}{5}$	$\frac{12698}{37375}$	$-\frac{201}{2990}$	$\frac{891}{11500}$	$\frac{1}{4}$	0
1	$\frac{944}{1365}$	$-\frac{400}{819}$	$\frac{99}{35}$	$-\frac{575}{252}$	$\frac{1}{4}$
b	$\frac{944}{1365}$	$-\frac{400}{819}$	$\frac{99}{35}$	$-\frac{575}{252}$	$\frac{1}{4}$
$\hat{b}$	$\frac{41911}{60060}$	$-\frac{83975}{144144}$	$\frac{3393}{1120}$	$-\frac{27025}{11088}$	$\frac{103}{352}$

This method is fourth-order accurate, L-stable but the embedded lower-order method may be A-stable, and provides an embedded third-order estimate for adaptive control [9].

4.2.4 Summary and usage

These methods provide a hierarchy of accuracy and stiffness robustness. SDIRK21 is the simplest, fully L-stable, and ideal for highly stiff problems. SDIRK32 offers a balance between accuracy and stability. SDIRK43 gives high accuracy while retaining good A-stability properties.

Before applying the positivity-preserving predictor–corrector correction (Section 3.3), we verify that these baseline SDIRK methods achieve their theoretical order on standard test problems. The second step is apply the correction from Sec. 3.3, i.e., correct only $\mathbf{y}_{n+1} = \mathbf{Y}_2$. Check positivity and order. The third step is apply the correction from Sec. 3.4, i.e., correct both $\mathbf{Y}_1$ and $\mathbf{y}_{n+1} = \mathbf{Y}_2$. Check positivity and order.

This framework generalizes to SSP SDIRK methods and higher-order schemes [10], ensuring positivity and conservation without degrading the formal order of accuracy.

Chapter 5

Numerical results

This chapter evaluates the proposed positivity-preserving predictor-corrector schemes for three main aspects:

1. Positivity enforcement. Verifying that physically meaningful constraints are maintained (see Section 5.1).

2. Invariant preservation. Verifying that intrinsic conservation laws are respected by the numerical scheme, even after corrections are applied (see Section 5.2).

3. Order. Ensuring that the formal convergence order of SDIRK methods is preserved (see Section 5.3).

4. Efficiency. Quantifying the computational overhead (or potential benefits) of applying corrections (see Section 5.4).

5.1 Positivity preservation

5.1.1 MAPK Cascade

The MAPK cascade did not exhibit any negative concentrations when simulated using the uncorrected SDIRK methods. This behavior suggests that the intrinsic dynamics of the

MAPK system—while moderately stiff—do not drive the solution outside the positive region, at least under the integration settings used in our tests.

Therefore, positivity-preserving corrections had no visible effect on the MAPK results in terms of enforcing non-negativity. However, corrections were still applied in subsequent sections to test their impact on invariants, convergence order, and computational performance.

5.1.2 Robertson Reaction

To assess the effectiveness of the positivity-preserving corrections, we tested the SDIRK21 method on the classical stiff Robertson reaction problem. This system is known to exhibit significant numerical challenges due to rapid changes and the potential for negative concentrations in standard integrators, particularly in the intermediate species $\mathbf{y}_2$.

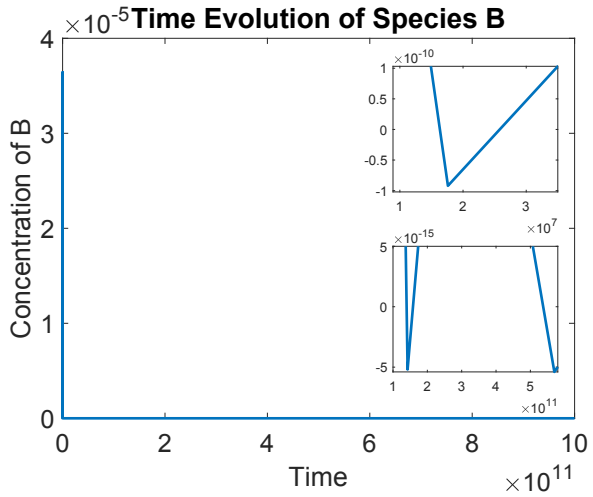


Figure 5.1: Time evolution of $\mathbf{y}_2$ base version of SDIRK21. Negative values are observed, indicating a violation of physical constraints.

Figure 5.1 shows that when the SDIRK21 scheme is applied without correction, the concentration of $\mathbf{y}_2$ becomes negative during the integration, which is physically unrealistic.

After applying our proposed positivity correction strategies—either at the final stage or

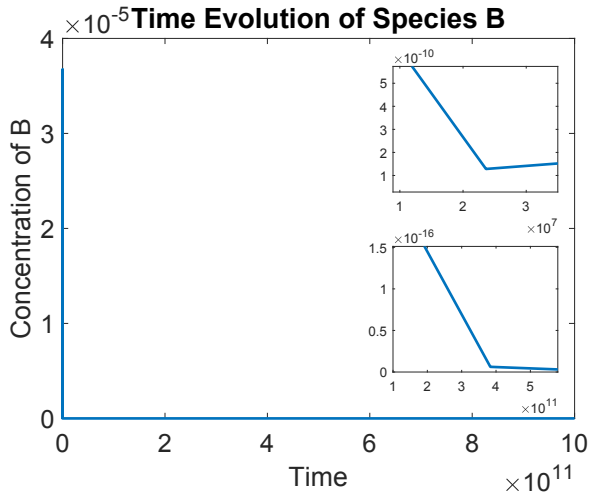


Figure 5.2: Time evolution of $\mathbf{y}_2$ using SDIRK21 with positivity correction applied to $\mathbf{y}_{n+1}$. All values remain strictly non-negative.

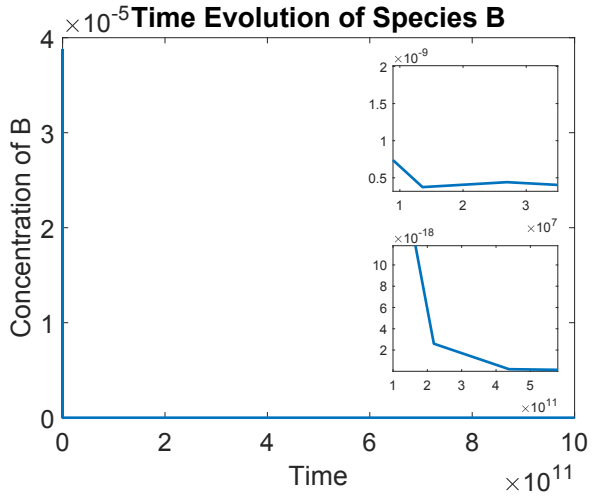


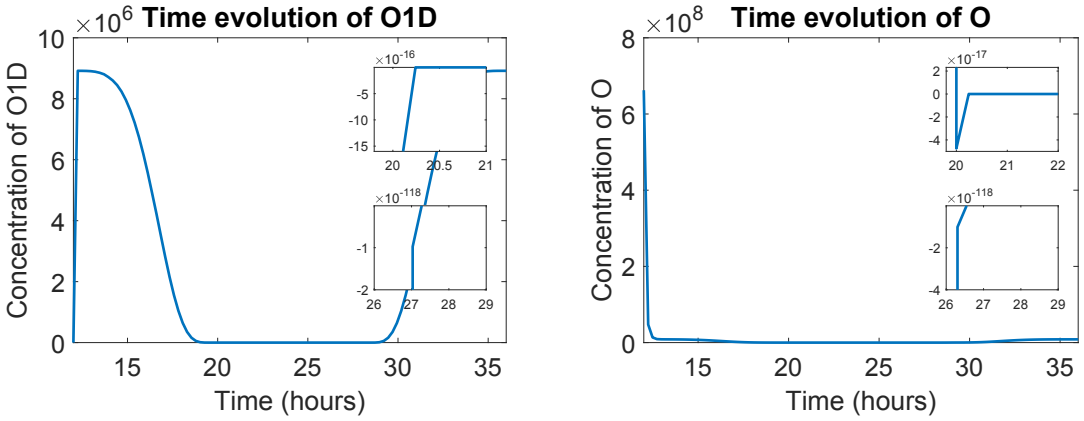
Figure 5.3: Time evolution of $\mathbf{y}_2$ using SDIRK21 with positivity correction applied to each stage. The solution remains strictly non-negative.

at all stages—we obtained fully non-negative trajectories as shown in Figures 5.2 and 5.3, respectively.

These results demonstrate that our correction scheme is essential for preserving the physical integrity of stiff reaction models like Robertson's. Importantly, the corrected methods do not introduce instability or noticeable error in the solution profile over the long simulation interval.

5.1.3 Stratospheric Reaction

To verify the effectiveness of the proposed correction scheme, we conducted a one-day simulation of the stratospheric reaction system using SDIRK21 method, focusing on the concentrations of species O and O^{1D}. These two concentrations are particularly tend to numerical negativity in standard time integration.



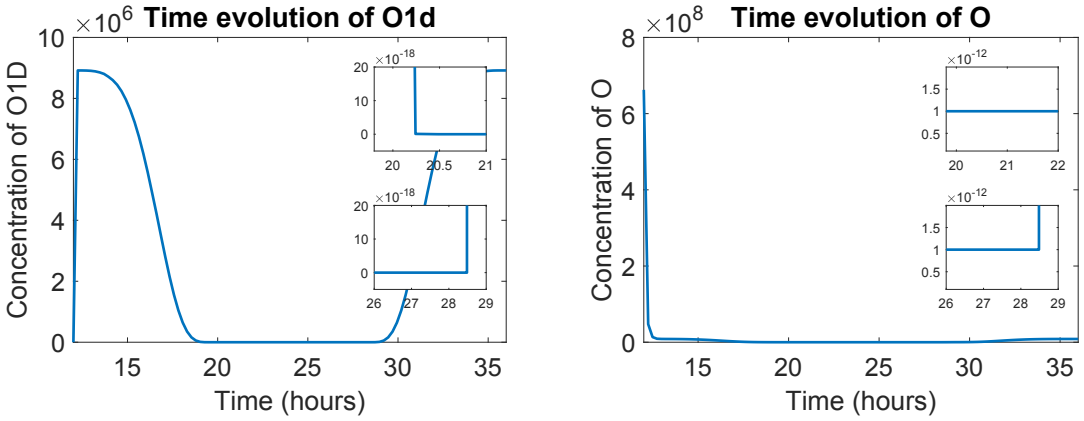
(a) Concentration of O^{1D} (b) Concentration of O

Figure 5.4: Concentrations of O^{1D} and O over time using base version of SDIRK21. Negative values are observed, indicating a violation of physical constraints.

Figure 5.4 illustrates the results of integrating the stratospheric reaction system using the SDIRK21 method without applying positivity corrections. In both cases concentrations become negative during the integration interval, which is physically meaningless and violates

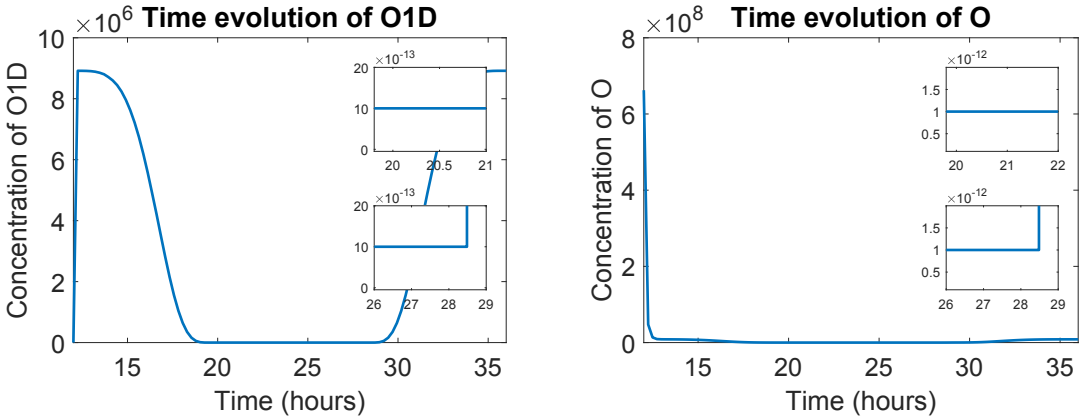
the positivity of chemical concentrations.

After applying the proposed positivity-preserving correction mechanism, the new results in Figure 5.5 and 5.6 show fully non-negative solutions throughout the simulation interval.



(a) Concentration of O^{1D} (b) Concentration of O

Figure 5.5: Concentrations of O^{1D} and O over time using SDIRK21 with positivity correction applied to $\mathbf{y}_{n+1}$. All values remain strictly non-negative.



(a) Concentration of O^{1D} (b) Concentration of O

Figure 5.6: Concentrations of O^{1D} and O over time using SDIRK21 with positivity correction applied to each stage. All values remain strictly non-negative.

This experiment highlights the necessity and effectiveness of the correction strategy. By enforcing positivity at each stage of the integration, the corrected method maintains the

physical realism of the solution without compromising the integration scheme's accuracy or efficiency.

A natural question is whether positivity could be enforced simply by rejecting steps that produce negative components and retrying with smaller step sizes. We tested this approach by halving h whenever a negative value appeared. While this strategy initially prevents negativity, in practice, the integrator quickly drives the step size below useful limits. In stiff regimes, the solver spends many consecutive attempts with extremely small steps without making appreciable progress in time, eventually hitting the "step size too small" termination condition. This experiment shows that simply decreasing the step size is not a viable solution and underscores the necessity of dedicated positivity-preserving corrections.

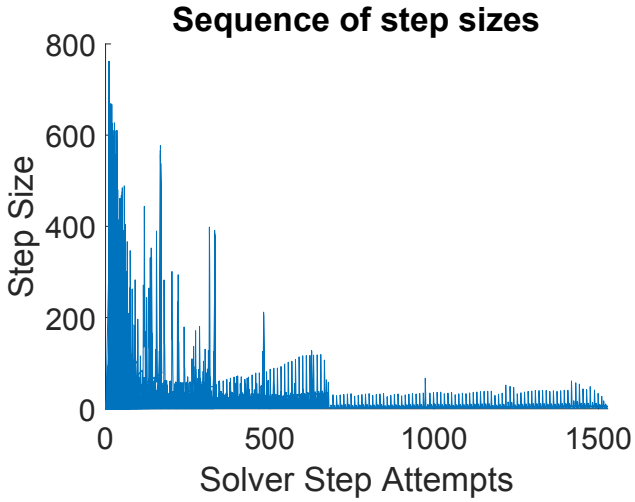


Figure 5.7: Sequence of attempted step sizes when negativity is handled by step rejection and halving. After initial transients, the step size collapses and remains near a tiny plateau, indicating repeated rejections and no practical progress; eventually the solver terminates with "step size too small". This demonstrates why reducing the step size is not a practical positivity-preserving strategy.

Figure 5.7 shows the sequence of solver step attempts when positivity was enforced by rejecting negative solutions and halving the step size. Each point corresponds to one attempted step, whether accepted or rejected. The rapid collapse of the step size is clearly visible,

after a few hundred attempts, h falls by several orders of magnitude and remains clustered near the numerical floor. This indicates that the solver becomes effectively stuck, expending hundreds of trial steps with negligible progress in physical time. Such behavior illustrates why enforcing positivity by step rejections alone is impractical, and motivates the need for dedicated positivity-preserving corrections that maintain progress without choking the step size.

5.2 Invariant preservation

To evaluate the ability of the proposed SDIRK schemes and positivity-preserving corrections to maintain the invariants of production–destruction systems, we evaluated four benchmark problems: MAPK cascade, stratospheric reaction system, and Robertson reaction. For each problem, we report the relative error in invariant preservation for three cases:

1. the baseline SDIRK integrator without correction,
2. the same integrator with correction applied only to the final stage $\mathbf{y}_{n+1}$,
3. the integrator with correction applied to all internal stages $\mathbf{Y}_i$.

5.2.1 Robertson Reaction

The Robertson reaction preserves the total concentration:

$$C = y_1 + y_2 + y_3.$$

Over the time interval $[0, 10^4]$, the relative deviation in mass was:

- Baseline SDIRK: 2.44×10^{-15},
- Final-stage correction: 2.22×10^{-15},
- All-stage correction: 3.99×10^{-15}.

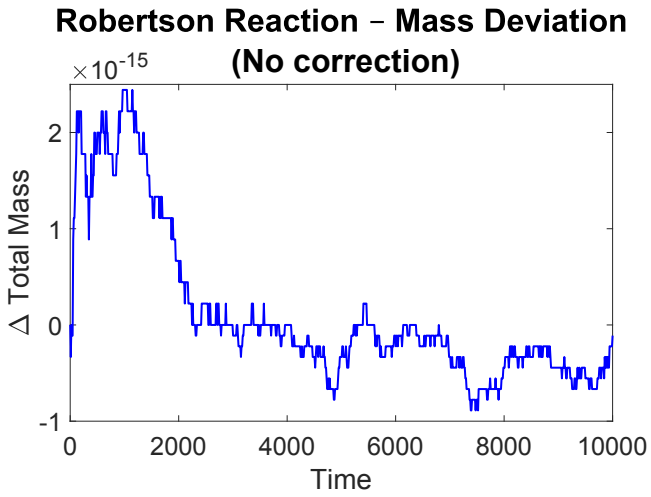


Figure 5.8: Robertson reaction: mass conservation for the baseline SDIRK scheme.

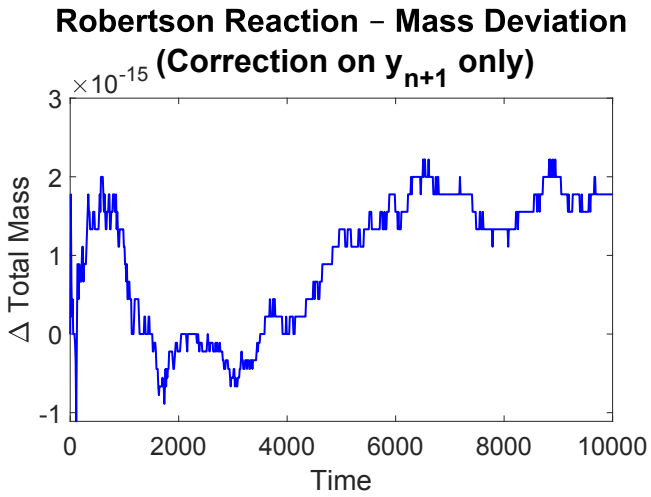


Figure 5.9: Robertson reaction: mass conservation with final-stage correction.

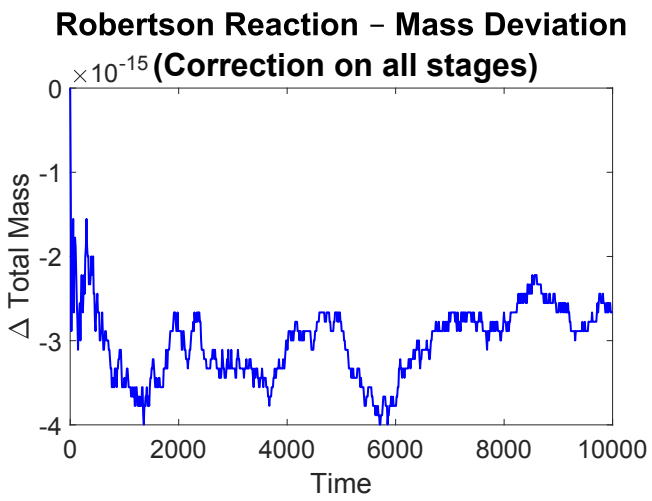


Figure 5.10: Robertson reaction: mass conservation with all-stage correction.

5.2.2 MAPK Cascade

As discussed in Subsection 4.1.2, the MAPK cascade has two linear conservation laws:

$$C_1 = \mathbf{y}_1 + \mathbf{y}_4 + \mathbf{y}_6, \quad C_2 = \mathbf{y}_2 + \mathbf{y}_3 + \mathbf{y}_4 + \mathbf{y}_5.$$

Numerical tests confirm that only one conservation law is preserved depending on the value of α. For $\alpha = 0$, only C_1 is preserved. For $\alpha = 1$, only C_2 is preserved.

These findings are consistent with the structure of $\mathbf{G}(y)$ and prior analysis by [1], who show that for endpoint values of α, one of the conservation laws is lost due to a non-zero left kernel mismatch.

All tests were conducted over the time interval $[0, 200]$. The relative errors in each invariant under different correction strategies are summarized below.

For $\alpha = 0$

- Baseline SDIRK: $\|C_1 - C_{1,0}\|/C_{1,0} = 9.14 \cdot 10^{-15}$, and $\|C_2 - C_{2,0}\|/C_{2,0} = 5.40 \cdot 10^{-4}$.

- Final-stage correction: $\|C_1 - C_{1,0}\|/C_{1,0} = 9.26 \cdot 10^{-15}$, and $\|C_2 - C_{2,0}\|/C_{2,0} = 5.39 \cdot 10^{-4}$.
- All-stage correction: $\|C_1 - C_{1,0}\|/C_{1,0} = 9.26 \cdot 10^{-15}$, and $\|C_2 - C_{2,0}\|/C_{2,0} = 5.73 \cdot 10^{-4}$.

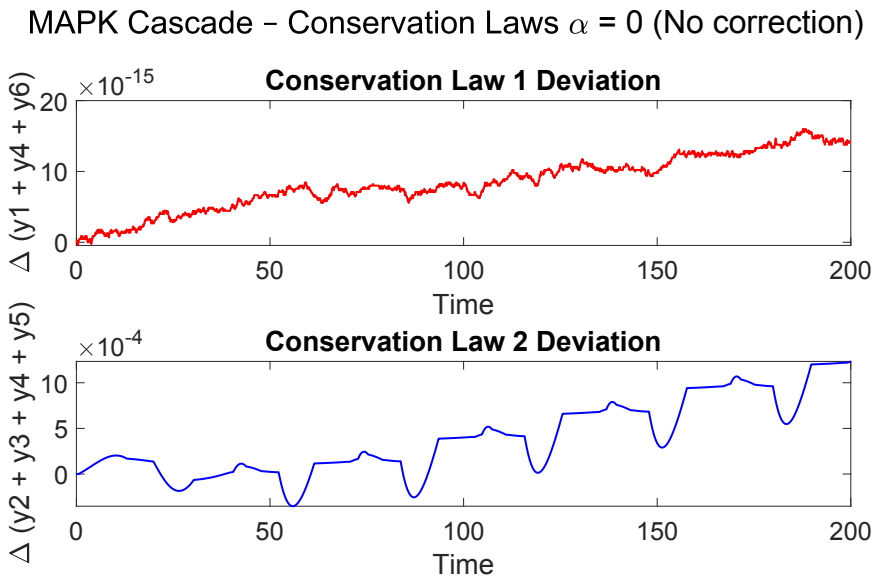


Figure 5.11: MAPK cascade: deviations from the conserved quantities for the baseline SDIRK scheme.

For $\alpha = 1$

- Baseline SDIRK: $\|C_1 - C_{1,0}\|/C_{1,0} = 9.48 \cdot 10^{-4}$, and $\|C_2 - C_{2,0}\|/C_{2,0} = 3.11 \cdot 10^{-15}$.
- Final-stage correction: $\|C_1 - C_{1,0}\|/C_{1,0} = 9.47 \cdot 10^{-4}$, and $\|C_2 - C_{2,0}\|/C_{2,0} = 3.11 \cdot 10^{-15}$.
- All-stage correction: $\|C_1 - C_{1,0}\|/C_{1,0} = 9.21 \cdot 10^{-4}$, and $\|C_2 - C_{2,0}\|/C_{2,0} = 3.11 \cdot 10^{-15}$.

5.2.3 Stratospheric Reaction System

The stratospheric system conserves the total number of oxygen and nitrogen atoms:

$$M_O = [1, 1, 3, 2, 1, 2]^\top \mathbf{y}, \quad M_N = [0, 0, 0, 0, 1, 1]^\top \mathbf{y}.$$

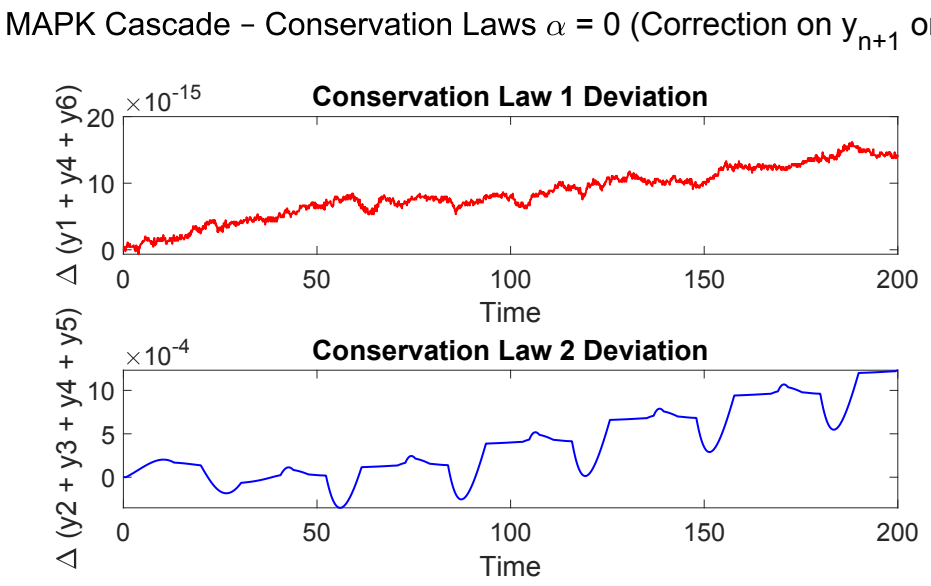


Figure 5.12: MAPK cascade: deviations from the conserved quantities with final-stage correction.

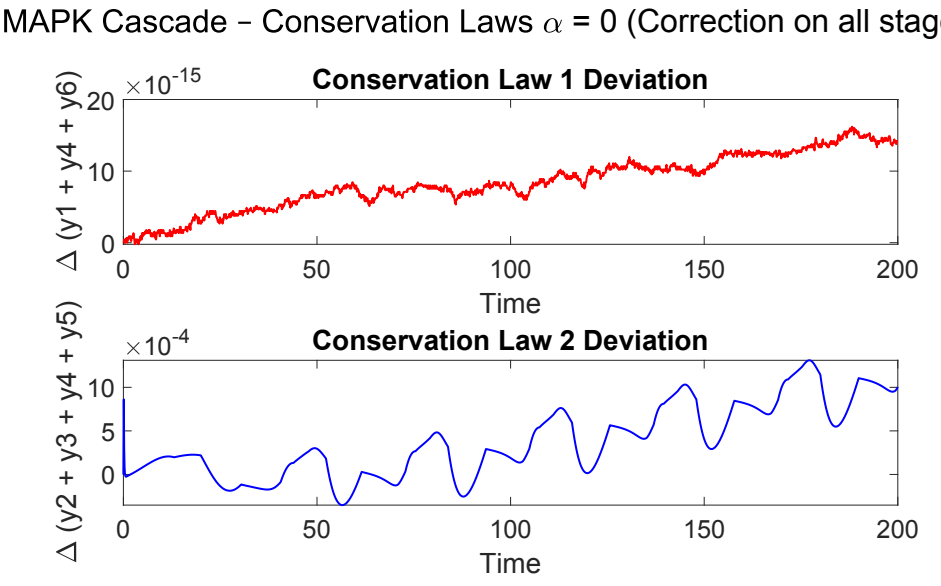


Figure 5.13: MAPK cascade: deviations from the conserved quantities with all-stage correction.

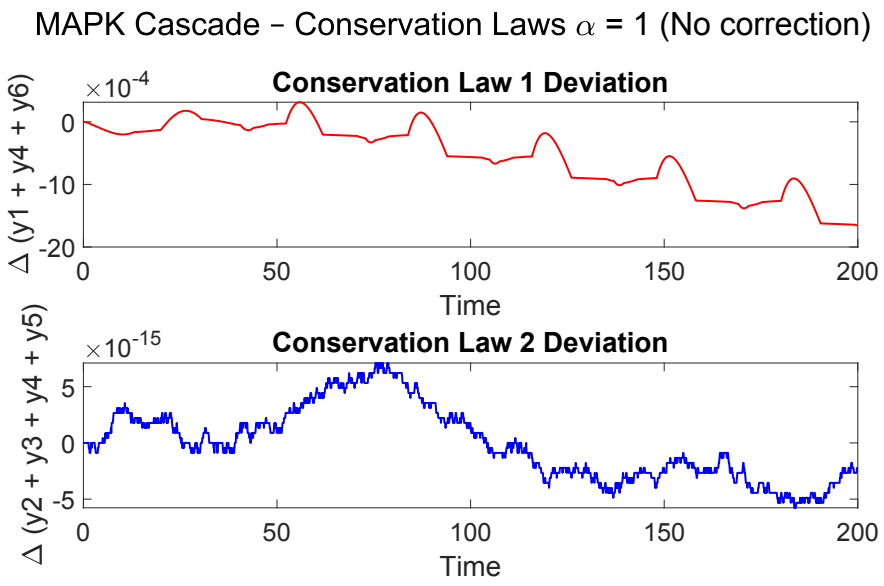


Figure 5.14: MAPK cascade: deviations from the conserved quantities for the baseline SDIRK scheme.

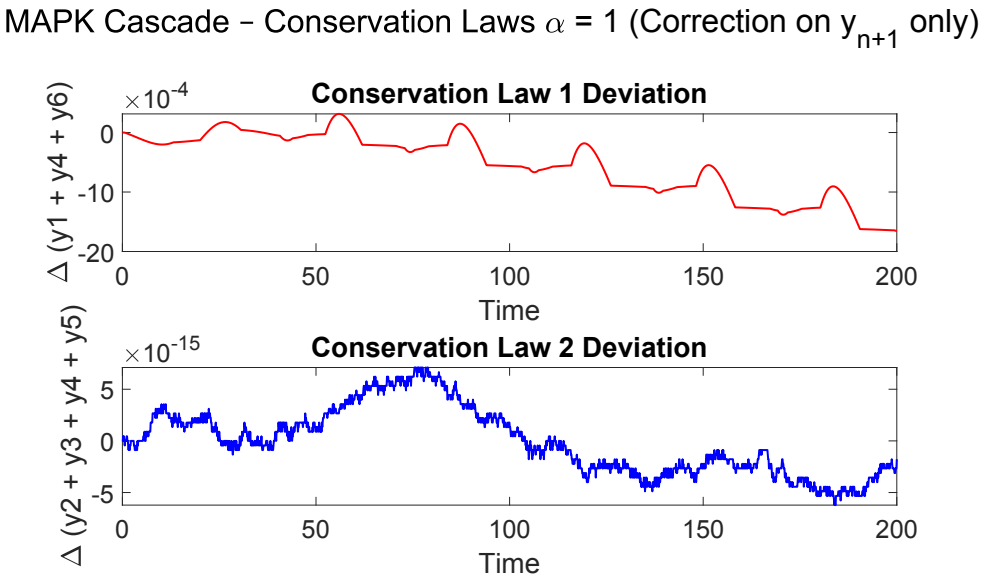


Figure 5.15: MAPK cascade: deviations from the conserved quantities with final-stage correction.

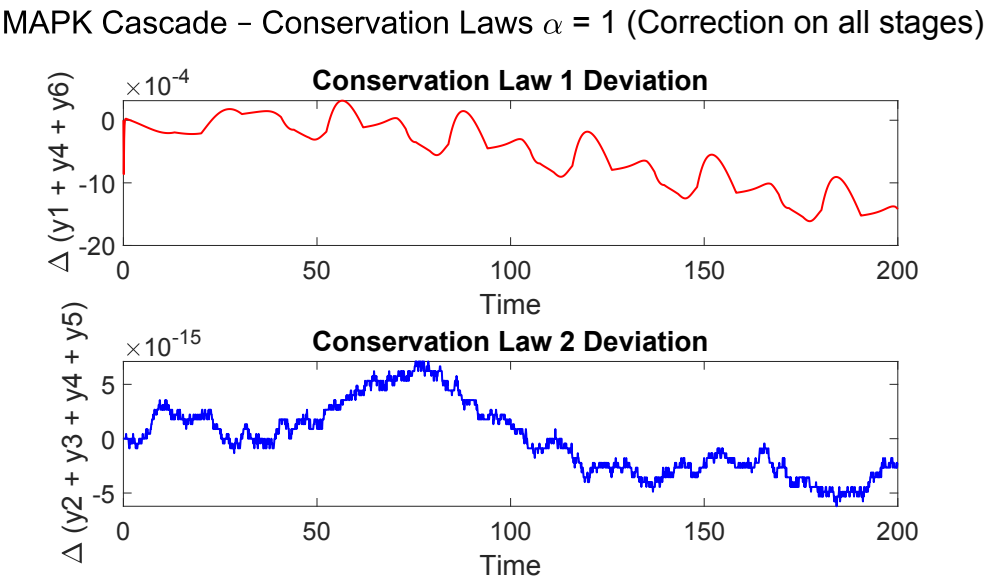


Figure 5.16: MAPK cascade: deviations from the conserved quantities with all-stage correction.

Over a one-day simulation ($[12 \cdot 3600, 36 \cdot 3600]$ seconds), the maximum relative atom conservation errors were:

- Baseline SDIRK: 4.89×10^{-14} for oxygen, 7.18×10^{-15} for nitrogen,
- Final-stage correction: 4.89×10^{-14} for oxygen, 7.39×10^{-15} for nitrogen,
- All-stage correction: 1.49×10^{-12} for oxygen, 7.18×10^{-15} for nitrogen.

5.2.4 Summary

For all benchmark problems, the invariant deviations remained at or near machine precision for stiff linear conservation laws (e.g., Robertson). For MAPK and stratospheric reaction, deviations remained within acceptable tolerance (10^{-4}–10^{-12}).

Importantly, applying positivity-preserving corrections did not significantly deteriorate in-

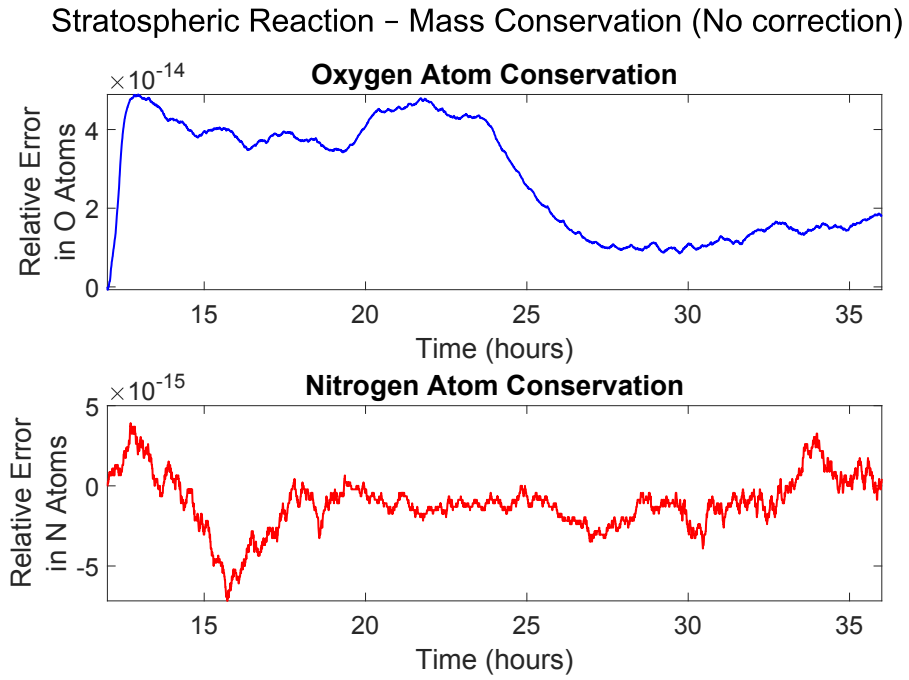


Figure 5.17: Stratospheric system: atom conservation for the baseline SDIRK scheme.

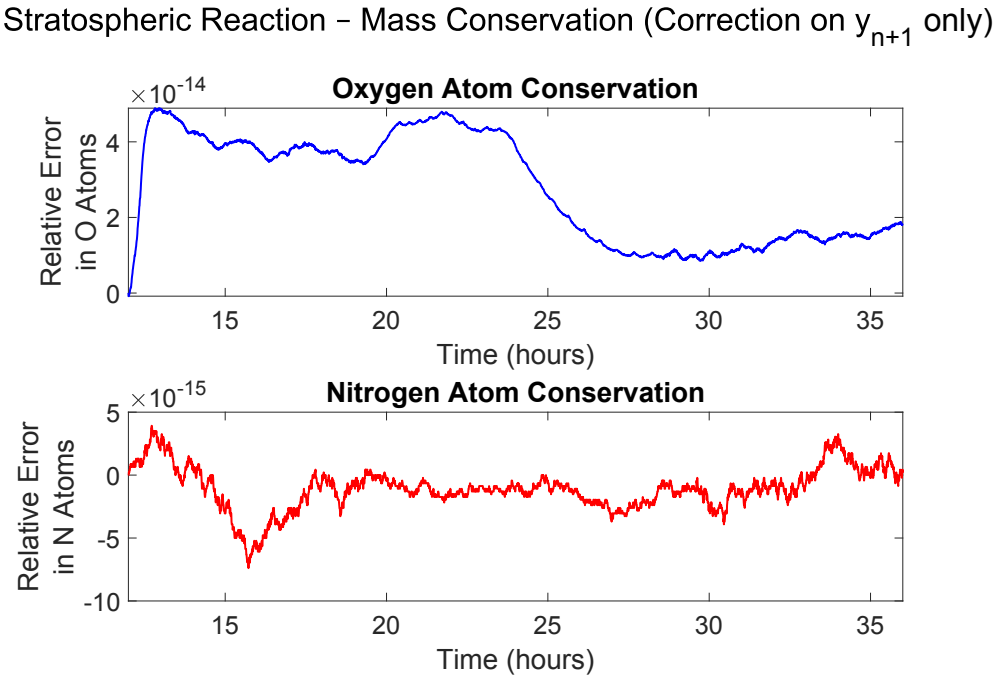


Figure 5.18: Stratospheric system: atom conservation with final-stage correction.

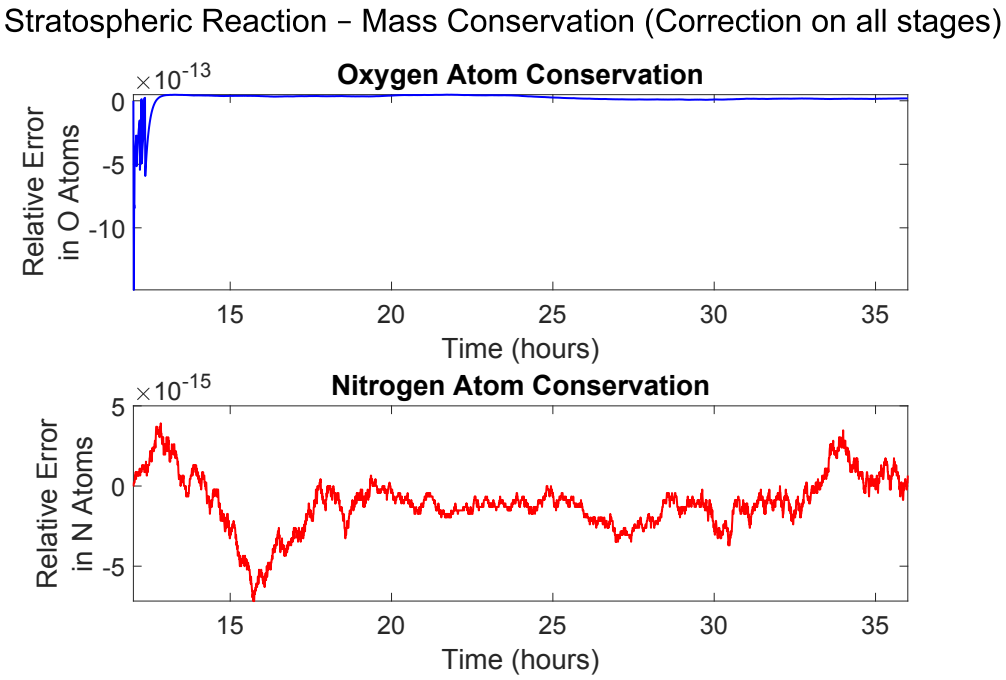


Figure 5.19: Stratospheric system: atom conservation with all-stage correction.

variant preservation. In some cases (e.g., MAPK), the correction slightly reduced the deviation, while in others (e.g., oxygen in the stratospheric case) it slightly increased due to accumulated round-off. Overall, the corrections retained the conservation properties of the underlying SDIRK schemes.

5.3 Order Validation

To assess whether the positivity-preserving corrections affect the formal accuracy of the underlying methods, we performed order validation on four representative problems: the Robertson reaction system, the stratospheric reaction mechanism, and the MAPK cascade. These problems span a wide range of stiffness and nonlinear dynamics, providing a thorough testbed for the proposed approach.

The adaptive step-size integration mode was used. Relative and absolute tolerances were varied over several orders of magnitude, and the global error at the final time was compared to a reference solution computed with very tight tolerances (10^{-14}).

We compared two variants of the positivity-preserving correction:

1. **Final-stage correction:** only the stiffly accurate stage $\mathbf{y}_{n+1}$ is clipped and rescaled.
2. **Full-stage correction:** all intermediate stages $\mathbf{Y}_i$ are clipped before being used in the final combination.

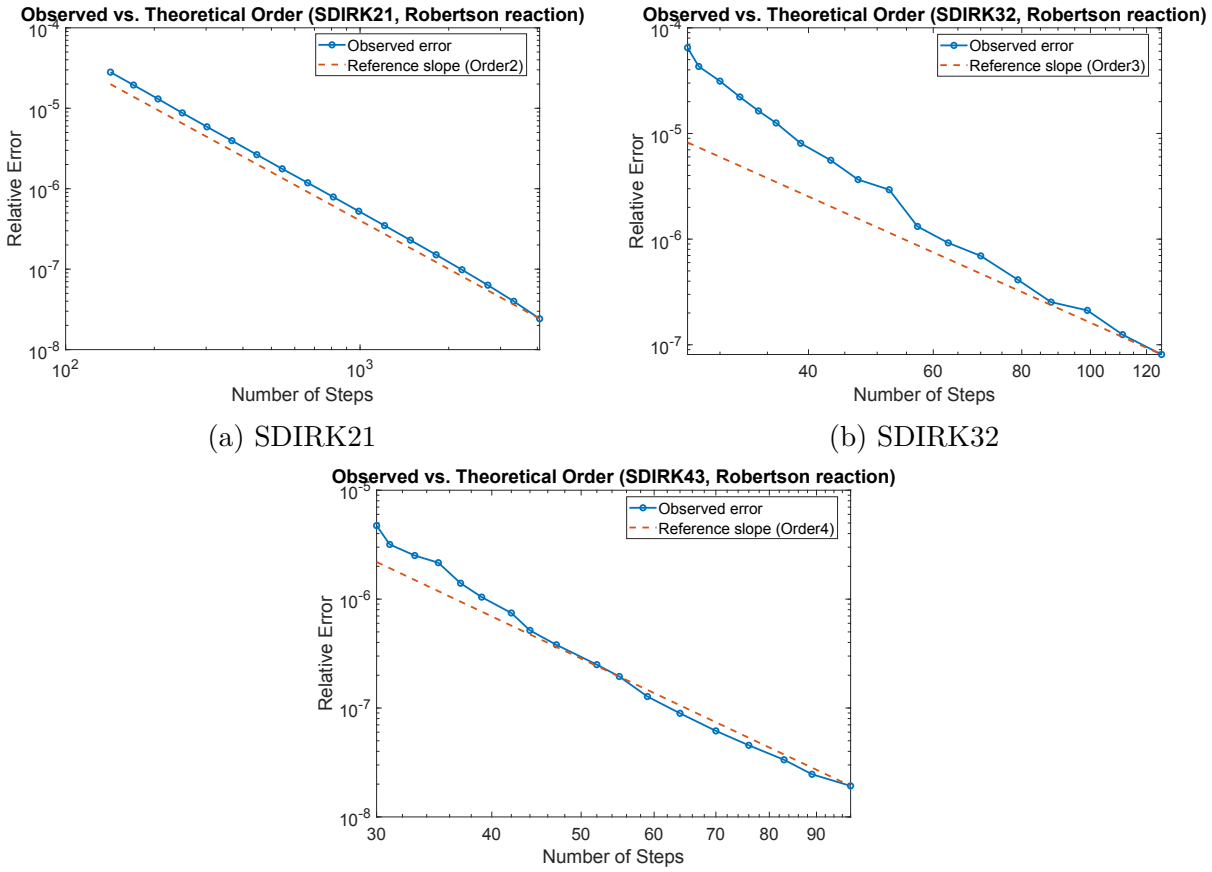
5.3.1 Robertson Reaction

The Robertson reaction is a classic stiff chemical kinetics model with widely separated time scales. To avoid dominance of long-term truncation error, we restricted the integration interval to $[0, 5000]$ with tolerances from 10^{-5} to 10^{-8}.

For the final-stage correction, the observed slopes were -2.07, -4.27, and -4.70 (Figure 5.20). For full-stage correction, the slopes remained consistent: -2.07, -4.28, and -4.62 (Figure 5.21). This again confirms that invariant-preserving corrections do not reduce the order even for a highly stiff problem.

5.3.2 MAPK Cascade

The MAPK cascade is a moderately stiff multiscale biochemical signaling network. To observe the asymptotic convergence regime, we integrated over the interval $[0, 60]$ using SDIRK21, SDIRK32, and SDIRK43 with adaptive tolerances ranging from 10^{-5} to 10^{-8}; the reference solution was computed with tolerance 10^{-14}.



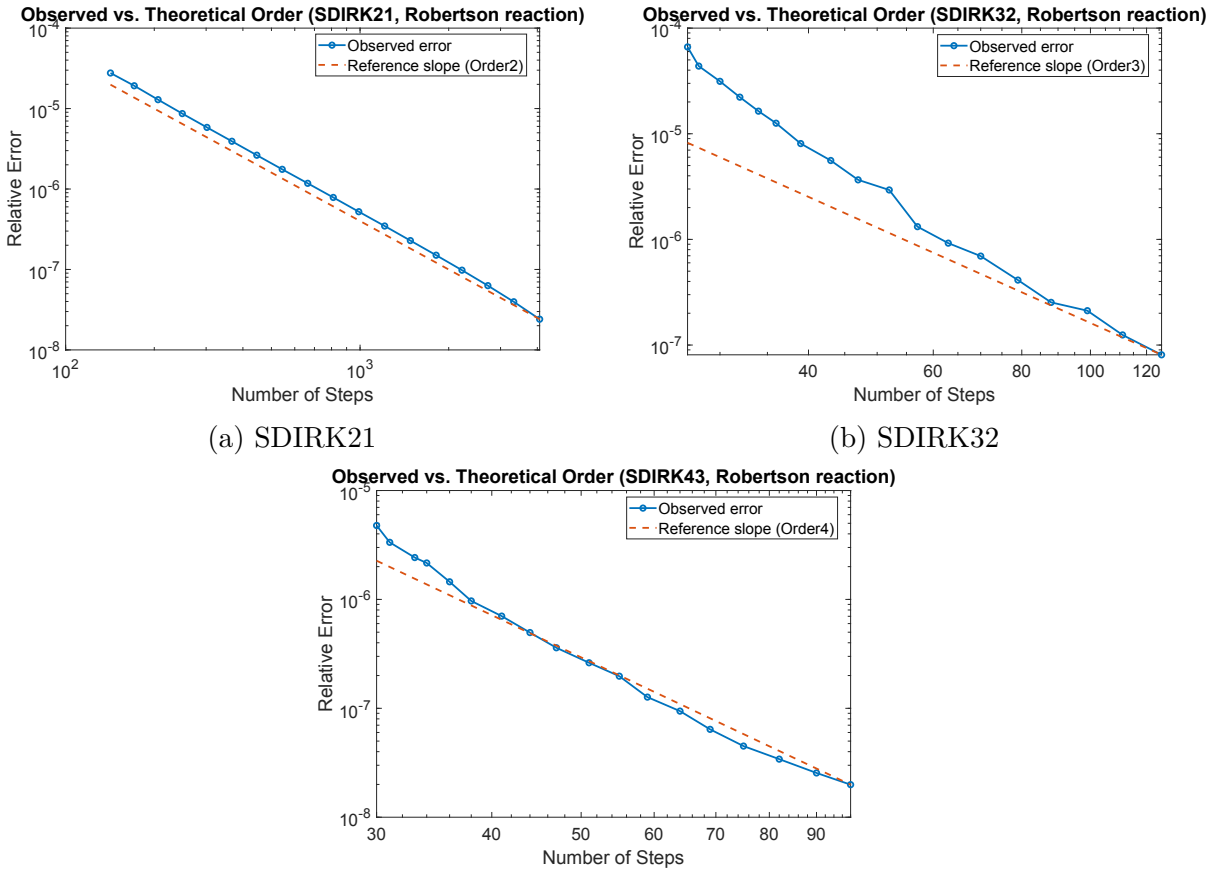
(a) SDIRK21

(b) SDIRK32

(c) SDIRK43

Figure 5.20: Robertson reaction: observed convergence order for SDIRK methods with positivity correction applied only to the final stage.

With final-stage correction, the measured slopes were close to the theoretical values: -1.98 (SDIRK21), -2.76 (SDIRK32), and -3.81 (SDIRK43), confirming that the correction does not alter the convergence rate (Figure 5.22). With full-stage correction, the slopes remained essentially unchanged: -1.98, -2.75, and -3.86 (Figure 5.23). Thus, for a moderately stiff biochemical network, both correction modes preserve the expected asymptotic order.

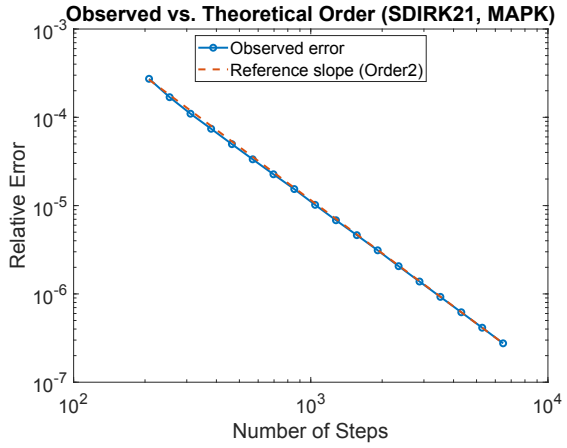


(a) SDIRK21

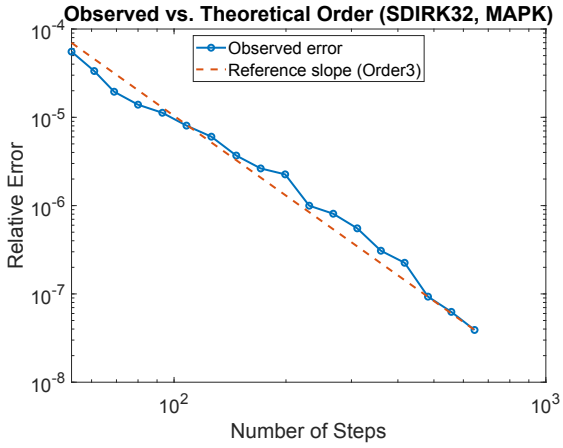
(b) SDIRK32

(c) SDIRK43

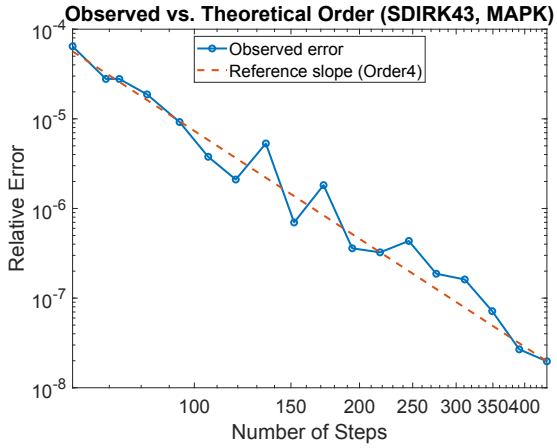
Figure 5.21: Robertson reaction: observed convergence order for SDIRK methods with positivity correction applied to all stages.



(a) SDIRK21

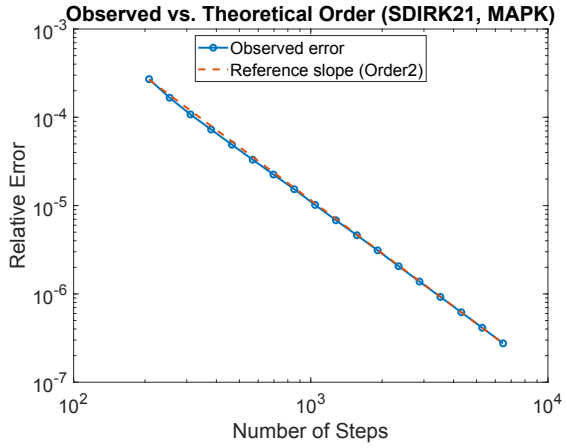


(b) SDIRK32

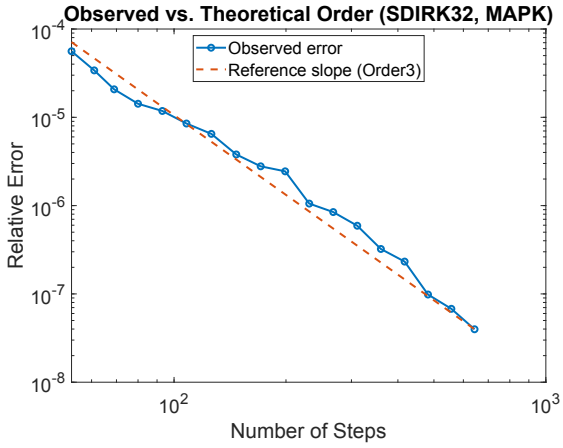


(c) SDIRK43

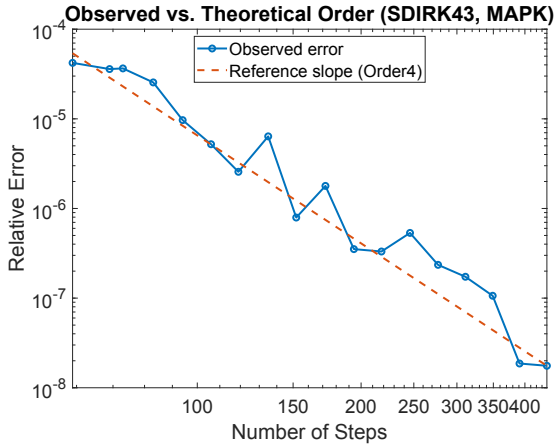
Figure 5.22: MAPK cascade: observed convergence order for SDIRK methods with positivity correction applied only to the final stage.



(a) SDIRK21



(b) SDIRK32



(c) SDIRK43

Figure 5.23: MAPK cascade: observed convergence order for SDIRK methods with positivity correction applied only to the final stage.

5.3.3 Stratospheric Reaction Mechanism

The stratospheric reaction mechanism is an extremely stiff, positivity-sensitive atmospheric chemistry model. We integrated over the interval $[19 \cdot 3600, 29 \cdot 3600]$ seconds to capture a representative stiff regime.

For the final-stage correction, the observed slopes were -1.51 (SDIRK21), -3.47 (SDIRK32), and -5.47 (SDIRK43) (Figure 5.24). The higher-than-expected slopes for the higher-order methods reflect rapid error damping typical of extremely stiff systems.

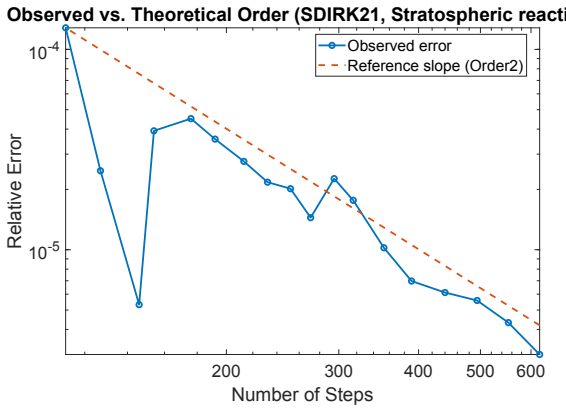
When applying full-stage correction, the slopes slightly decreased but still followed the expected order trend: -1.95, -2.84, and -3.76 (Figure 5.25). Thus, even in extreme stiffness, full-stage clipping maintains the nominal order without introducing significant degradation.

5.3.4 Summary

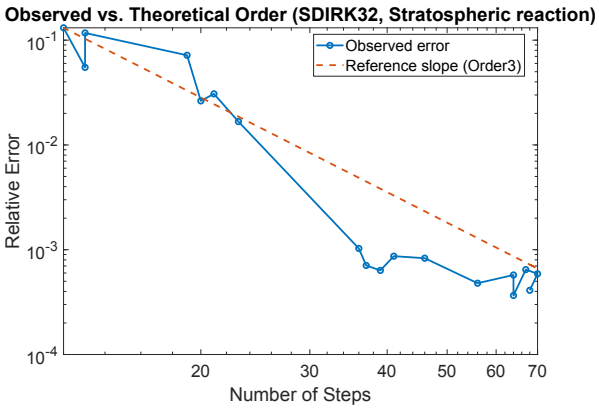
Across all test problems, results demonstrate that the proposed correction mechanism preserves the formal order of the implicit SDIRK methods across diverse classes of problems—stiff chemical kinetics, multiscale biochemical signaling networks, and nonlinear dispersive systems—both in adaptive and fixed step-size settings.

5.4 Efficiency Analysis

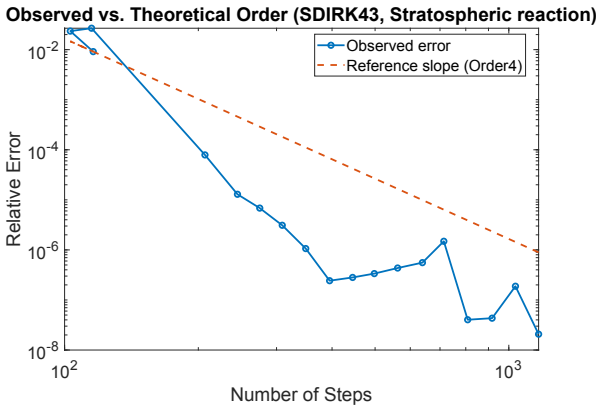
To evaluate the computational efficiency of the proposed positivity-preserving integrator, we benchmarked its performance against a standard Runge-Kutta implementation without corrections. The comparison was conducted using three representative production-destruction models: the stratospheric reaction, the MAPK cascade, and the Robertson reaction. Each



(a) SDIRK21

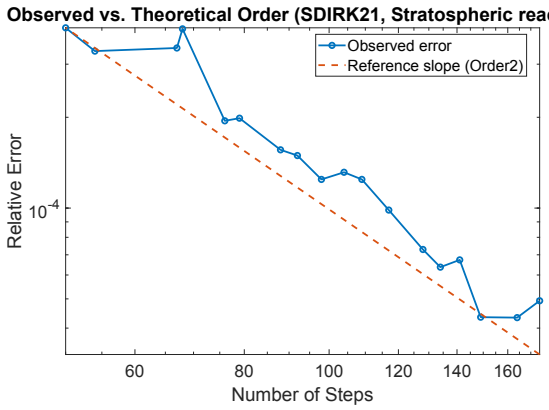


(b) SDIRK32

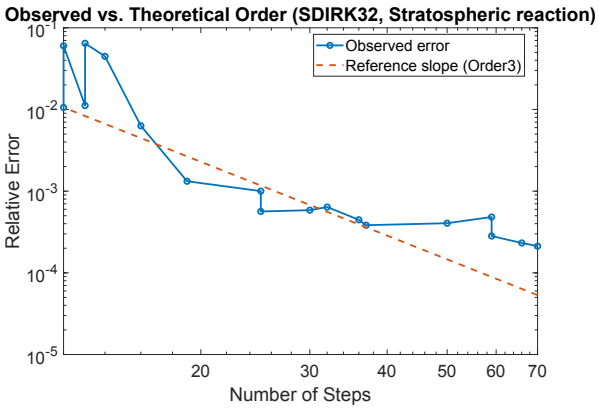


(c) SDIRK43

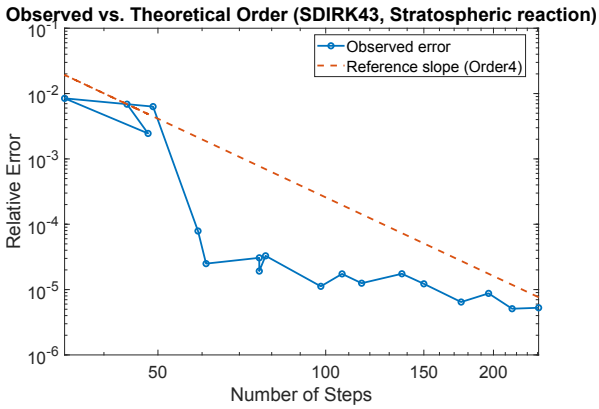
Figure 5.24: Stratospheric reaction: observed convergence order for SDIRK methods with positivity correction applied only to the final stage.



(a) SDIRK21



(b) SDIRK32



(c) SDIRK43

Figure 5.25: Stratospheric reaction: observed convergence order for SDIRK methods with positivity correction applied to all stages.

system presents different levels of stiffness and conservation demands, providing a diverse proving ground for efficiency analysis.

All simulations were run using fixed relative and absolute tolerances of 10^{-7} to ensure consistent accuracy across methods. For each problem, we measured the wall-clock time required to complete the integration over its natural time span: 3 days ($[12 \cdot 3600, 84 \cdot 3600]$) for stratospheric reaction, $[0, 200]$ for MAPK cascade, and $[0, 10^4]$ for the Robertson reaction. Each experiment was repeated five times, and average runtimes were computed to mitigate fluctuations due to background processes. Timing was recorded using MATLAB's tic and toc commands. All tests were conducted using the SDIRK43 scheme, a fourth-order singly diagonally implicit Runge-Kutta method, both with and without the proposed positivity-preserving correction.

The stratospheric reaction system, which is highly stiff and includes numerous positivity-sensitive interactions, showed a substantial runtime increase when corrections were applied. The average runtime rose from approximately 30.2 seconds without corrections to 110.5 seconds with them, leading to an overhead of roughly 266%. In contrast, the MAPK cascade results demonstrated minimal runtime impact, with the corrected integrator averaging 0.516 seconds compared to 0.560 seconds without correction. Interestingly, for the Robertson reaction, the corrected version showed significant improvement, reducing runtime from 0.485 seconds to 0.090 seconds on average. The efficiency improvement is exceeding 80%.

These results demonstrate that while positivity-preserving corrections can introduce overhead in some stiff problems, they can also lead to improvement in performance in other situations. Overall, the proposed correction mechanism is a practical and effective solution for a wide range of stiff ODE systems.

Chapter 6

Conclusions

This thesis addressed the development and evaluation of a positivity-preserving correction strategy for the numerical integration of stiff systems of ordinary differential equations, particularly those representing production–destruction processes. Recognizing that standard integration methods may violate fundamental structural properties such as non-negativity and conservation, we proposed a post-processing approach compatible with general time integrators, including singly diagonally implicit Runge–Kutta (SDIRK) schemes. The correction procedure, based on clipping negative entries and applying structure-preserving scaling, was shown to maintain both positivity and exact preservation of mass or other linear invariants without requiring changes to the base integration algorithm.

Extensive numerical experiments involving the stratospheric reaction system, the MAPK cascade, and the Robertson reaction demonstrated that the proposed correction reliably enforces positivity and invariant conservation across a wide range of stiff and multiscale problems. Order validation tests confirmed that the formal order of accuracy of the base methods was preserved, even when corrections were applied at every stage. Efficiency benchmarks revealed that while the correction step introduces computational overhead in some stiff regimes, it can also improve robustness and performance in challenging scenarios.

In conclusion, the presented positivity-preserving correction strategy provides a practical, general, and effective way to enforce physical constraints such as positivity, mass conservation, and accuracy in the numerical solution of stiff ODEs. It extends the applicability of

classical time integration methods to a broad class of scientific and engineering problems, from atmospheric chemistry to nonlinear wave propagation.

Bibliography

[1] Blanes, Sergio, Iserles, Arieh, and Macnamara, Shev. Positivity-preserving methods for ordinary differential equations. *ESAIM: M2AN*, 56(6):1843–1870, 2022. doi: 10.1051/m2an/2022042. URL https://doi.org/10.1051/m2an/2022042.

[2] Hans Burchard, Eric Deleersnijder, and Andreas Meister. A high-order conservative patankar-type discretisation for stiff systems of production–destruction equations. *Applied Numerical Mathematics*, 47(1):1–30, 2003. ISSN 0168-9274. doi: https://doi.org/10.1016/S0168-9274(03)00101-6. URL https://www.sciencedirect.com/science/article/pii/S0168927403001016.

[3] Hans Burchard, Eric Deleersnijder, and Andreas Meister. Application of modified patankar schemes to stiff biogeochemical models for the water column. *Ocean Dynamics*, 55(3):326–337, Dec 2005. ISSN 1616-7228. doi: 10.1007/s10236-005-0001-x. URL https://doi.org/10.1007/s10236-005-0001-x.

[4] Syvert P. Nørsett Ernst Hairer, Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff problems*, page 205. Springer Series in Computational Mathematics. Springer, Berlin, Heidelberg, 1993.

[5] Huang, Juntao, Izgin, Thomas, Kopecz, Stefan, Meister, Andreas, and Shu, Chi-Wang. On the stability of strong-stability-preserving modified patankar–runge–kutta schemes. *ESAIM: M2AN*, 57(2):1063–1086, 2023. doi: 10.1051/m2an/2023005. URL https://doi.org/10.1051/m2an/2023005.

[6] Willem Hundsdorfer and Jan G Verwer. *Numerical solution of time-dependent advection-*

diffusion-reaction equations, volume 33 of *Springer Series in Computational Mathematics*, pages 3–4. Springer Science & Business Media, Berlin, Heidelberg, 2013.

[7] Thomas Izgin, Stefan Kopecz, Angela Martiradonna, and Andreas Meister. On the dynamics of first and second order geco and gbbks schemes. *Applied Numerical Mathematics*, 193:43–66, 2023. ISSN 0168-9274. doi: https://doi.org/10.1016/j.apnum.2023.07.014. URL https://www.sciencedirect.com/science/article/pii/S0168927423001976.

[8] Izgin, Thomas, Kopecz, Stefan, and Meister, Andreas. On lyapunov stability of positive and conservative time integrators and application to second order modified patankar–runge–kutta schemes. *ESAIM: M2AN*, 56(3):1053–1080, 2022. doi: 10.1051/m2an/2022031. URL https://doi.org/10.1051/m2an/2022031.

[9] Christopher A. Kennedy and Mark H. Carpenter. Diagonally implicit runge-kutta methods for ordinary differential equations. a review. Technical report, NASA Technical Report, 03 2016.

[10] David I. Ketcheson, Colin B. Macdonald, and Sigal Gottlieb. Optimal implicit strong stability preserving runge–kutta methods. *Applied Numerical Mathematics*, 59(2):373–392, 2009. ISSN 0168-9274. doi: https://doi.org/10.1016/j.apnum.2008.03.034. URL https://www.sciencedirect.com/science/article/pii/S0168927408000688.

[11] S. Kopecz and A. Meister. On order conditions for modified patankar–runge–kutta schemes. *Applied Numerical Mathematics*, 123:159–179, 2018. ISSN 0168-9274. doi: https://doi.org/10.1016/j.apnum.2017.09.004. URL https://www.sciencedirect.com/science/article/pii/S0168927417301861.

[12] Stefan Kopecz and Andreas Meister. Unconditionally positive and conservative third order modified patankar–runge–kutta discretizations of production–destruction sys-

tems. *BIT Numerical Mathematics*, 58(3):691–728, Sep 2018. ISSN 1572-9125. doi: 10.1007/s10543-018-0705-1. URL https://doi.org/10.1007/s10543-018-0705-1.

[13] Stefan Kopecz and Andreas Meister. On the existence of three-stage third-order modified patankar–runge–kutta schemes. *Numerical Algorithms*, 81(4):1473–1484, Aug 2019. ISSN 1572-9265. doi: 10.1007/s11075-019-00680-3. URL https://doi.org/10.1007/s11075-019-00680-3.

[14] Angela Martiradonna, Gianpiero Colonna, and Fasma Diele. Geco: Geometric conservative nonstandard schemes for biochemical systems. *Applied Numerical Mathematics*, 155:38–57, 2020. ISSN 0168-9274. doi: https://doi.org/10.1016/j.apnum.2019.12.004. URL https://www.sciencedirect.com/science/article/pii/S0168927419303368. Structural Dynamical Systems: Computational Aspects held in Monopoli (Italy) on June 12-15, 2018.