

ABSOLUTELY STABLE EXPLICIT SCHEMES FOR REACTION SYSTEMS

CHANG-OCK LEE, CHAE HUN LEEM, EUN-HEE PARK, AND JAE BOUM YOUM

ABSTRACT. We introduce two numerical schemes for solving a system of ordinary differential equations which characterizes several kinds of linear reactions and diffusion from biochemistry, physiology, etc. The methods consist of sequential applications of the simple exact solver for a reversible reaction. We prove absolute stability and convergence of the proposed explicit methods. One is of first order and the other is of second order. Numerical results are included.

1. Introduction

Many phenomena of interest in physiology and biochemistry are characterized by reactions among several chemical species and diffusion in various mediums (see [6–8, 10]). In a closed system, both reactions and diffusion are governed by a system of ordinary differential equations (ODEs)

$$(1.1) \quad \dot{\mathbf{y}}(t) = M\mathbf{y}(t),$$

which guarantees conservation of the total amount of $\mathbf{y}(t)$ for any $t \geq 0$. Since we are concerned with the steady-state solution as well as the transient in simulations of very large systems of chemical reactions or molecular dynamics, we need to take the overall computational cost into consideration. Many physiologists and biochemists prefer explicit methods to implicit methods since implementation of the explicit methods is easier than the others. The popular methods for reaction systems are simple explicit schemes such as Euler's method, Runge-Kutta method, etc. However, it is well-known that conditional stability, the typical weak point of explicit methods, is very fatal for stiff problems. In the past few decades, many studies on numerical methods for stiff ODEs have been done in various aspects (see [1, 2, 4, 5, 11]).

The aim of this paper is to present two absolutely stable explicit schemes which are applicable to a linear reaction system (1.1). In 1978, Rush and

Received April 19, 2008.

2000 *Mathematics Subject Classification.* Primary 35K45, 65L20, 92C30.

Key words and phrases. reaction system, chemical diffusion, absolutely stable scheme, explicit scheme.

This work was partially supported by KRF-2006-013-C00082.

Larsen [9] introduced an iterative procedure for the Hodgkin-Huxley model for cell membrane behavior, which is composed of a circuit equation for currents and a coupled system of nonlinear ODEs for the ionic gates. An integration algorithm was suggested for a numerical solution to the ODEs for the ionic gates, which was based on the exact solution of a linearized ionic gate equation. Similarly, the methods in this paper are motivated by the simple exact solver for a reversible reaction. In spite of their explicitness, we have unconditional stability, that is, stability without any condition on the step size. Furthermore, we prove the convergence of the proposed methods; one is of first order and the other is of second order.

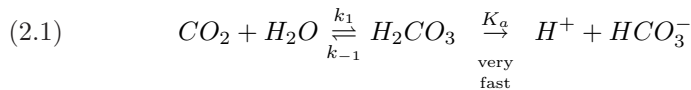
This paper is organized as follows. In Section 2, we introduce the reaction systems of our interest and propose two numerical methods for a linear reaction system. Section 3 provides theoretical results for convergence and stability of the proposed methods. In Section 4, we provide numerical experiments for typical reaction systems.

2. Reaction systems and numerical methods

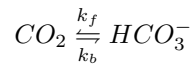
We consider two typical types of reactions: reversible reactions and circular reactions. A reaction of the type



is called the reversible reaction, where k_f and k_b are the rate constants for the forward and backward reactions. One interesting biochemical system to which the reversible first order equations apply is the carbonic acid system:



Equation (2.1) reduces to



(see [8]). Then, the rate equations are written down as

$$(2.2) \quad \begin{aligned} \frac{dA}{dt} &= -k_b A + k_f B \\ \frac{dB}{dt} &= k_b A - k_f B, \end{aligned}$$

where $A(t)$ and $B(t)$ are the concentrations of CO_2 and HCO_3^- as functions of time t . Another interesting example of such a type occurs frequently in metabolic studies. The other type is a circular reaction shown in Figure 1. We

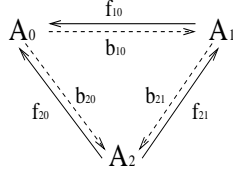


FIGURE 1. A circular reaction system with 3 substances A_0, A_1 and A_2

may write the differential equations describing this process as

$$(2.3) \quad \frac{d}{dt} \begin{bmatrix} A_0(t) \\ A_1(t) \\ A_2(t) \end{bmatrix} = \begin{bmatrix} -b_{10} - b_{20} & f_{10} & f_{20} \\ b_{10} & -f_{10} - b_{21} & f_{21} \\ b_{20} & b_{21} & -f_{21} - f_{20} \end{bmatrix} \begin{bmatrix} A_0(t) \\ A_1(t) \\ A_2(t) \end{bmatrix}.$$

From the fact that the total concentration $A(t) + B(t)$ remains constant for all $t \geq 0$ in a closed system, the exact solution of (2.2) is written in a form

$$\begin{aligned} \begin{bmatrix} A(t) \\ B(t) \end{bmatrix} &= \begin{bmatrix} \frac{1}{k_f+k_b}(k_f+k_b e^{-(k_f+k_b)t}) & \frac{k_f}{k_f+k_b}(1-e^{-(k_f+k_b)t}) \\ \frac{k_b}{k_f+k_b}(1-e^{-(k_f+k_b)t}) & \frac{1}{k_f+k_b}(k_b+k_f e^{-(k_f+k_b)t}) \end{bmatrix} \begin{bmatrix} A(0) \\ B(0) \end{bmatrix} \\ &= \exp\left(t \begin{bmatrix} -k_b & k_f \\ k_b & -k_f \end{bmatrix}\right) \begin{bmatrix} A(0) \\ B(0) \end{bmatrix}. \end{aligned}$$

Similarly, we can find the exact solution for the circular reaction in Figure 1. In general, a reaction system is characterized by a coupled system of ODEs. To solve a relevant eigenvalue problem is the first step in solving such a coupled system exactly (see [12]). But as the number of substances increases, the exact solver suffers from typical difficulties in large scale eigenvalue problems. There are some numerical techniques in common use: the *Euler method*, which is the simplest one, but requires a small size of time step Δt ; the *Runge-Kutta method*, which is more complicated, but allows much bigger time steps to be taken. Now for the circular reaction case, we propose new numerical methods motivated by the above process that is used to find the exact solution to a reversible reaction. For the sake of simplicity, we illustrate these algorithms for a simple circular reaction (2.3) although they are applicable to more complicated reaction systems.

Algorithm 1: CR²

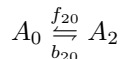
- (1) For each $k \in \mathbb{N}$, let $A_{i,k}$ be the approximate solution to $A_i(t)$ at time $t_k = k\Delta t$.
- (2) For $k = 0, 1, 2, \dots$ do:
 - (a) Set $A_0^{ic} = A_{0,k}, A_1^{ic} = A_{1,k}$ and $A_2^{ic} = A_{2,k}$.

- (b) Find the exact solution $A_0^{temp}(t)$ and $A_1^{temp}(t)$ of the reversible reaction with the initial values A_0^{ic} and A_1^{ic} :

$$A_0 \xrightleftharpoons[b_{10}]^{f_{10}} A_1$$

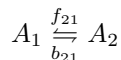
Set $A_0^{ic} = A_0^{temp}(t_{k+1})$ and $A_1^{ic} = A_1^{temp}(t_{k+1})$.

- (c) Find the exact solution $A_0^{temp}(t)$ and $A_2^{temp}(t)$ of the reversible reaction with the initial values A_0^{ic} and A_2^{ic} :



Set $A_{0,k+1} = A_0^{temp}(t_{k+1})$ and $A_2^{ic} = A_2^{temp}(t_{k+1})$.

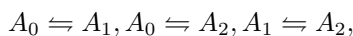
- (d) Find the exact solution $A_1^{temp}(t)$ and $A_2^{temp}(t)$ of the reversible reaction with the initial values A_1^{ic} and A_2^{ic} :



Set $A_{1,k+1} = A_1^{temp}(t_{k+1})$ and $A_{2,k+1} = A_2^{temp}(t_{k+1})$.

Remark 2.1. CR² stands for Consecutive Reversible Reactions.

The key idea of Algorithm 1 is that we approximately regard a circular reaction as a consecutive reaction which consists of three separated reversible reactions. Although we solve three reversible reactions in the following order



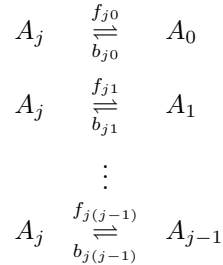
the algorithm does not depend upon the ordering we adopt to split a circular reaction into a chain of reversible reactions. In a form of matrix exponential, Algorithm 1 is represented by the following linear system:

$$\begin{bmatrix} A_{0,k+1} \\ A_{1,k+1} \\ A_{2,k+1} \end{bmatrix} = \exp(\Delta t \bar{M}_{21}) \exp(\Delta t \bar{M}_{20}) \exp(\Delta t \bar{M}_{10}) \begin{bmatrix} A_{0,k} \\ A_{1,k} \\ A_{1,k} \end{bmatrix} \\ \stackrel{\text{def}}{=} L_2 \begin{bmatrix} A_{0,k} \\ A_{1,k} \\ A_{2,k} \end{bmatrix},$$

where \bar{M}_{ij} is a 3 by 3 matrix relevant to the reversible reaction between A_j and A_i such that

$$(2.4) \quad (\bar{M}_{ij})_{lm} = \begin{cases} -b_{ij} & \text{if } (l, m) = (j, j), \\ f_{ij} & \text{if } (l, m) = (j, i), \\ b_{ij} & \text{if } (l, m) = (i, j), \\ -f_{ij} & \text{if } (l, m) = (i, i), \\ 0 & \text{otherwise.} \end{cases}$$

We interpret the Algorithm 1 for n substances inductively, i.e., at $t = t_k$, we assume that the reaction system with j substances A_0, A_1, \dots, A_{j-1} ($1 \leq j \leq n - 1$) is solved, then we solve the parallel reaction that consists of j reversible reactions:

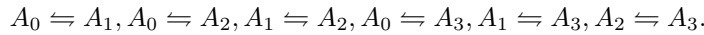


Algorithm 2: SCR²

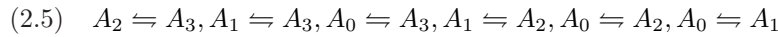
- (1) For each $k \in \mathbb{N}$, let $A_{i,k}^s$ be the approximate solution to $A_i(t)$ at time $t_k = k\Delta t$.
- (2) For $k = 0, 1, 2, \dots$ do:
 - (a) Set $A_0^{ic} = A_{0,k}, A_1^{ic} = A_{1,k}$ and $A_2^{ic} = A_{2,k}$.
 - (b) As in Algorithm 1, solve three reversible reactions according to chosen ordering: $A_0 \rightleftharpoons A_1, A_0 \rightleftharpoons A_2, A_1 \rightleftharpoons A_2$. Define $A_{i,k+1}^{s1}$ by the obtained approximation $A_{i,k+1}$ for $i = 1, 2, 3$.
 - (c) Solve three reversible reactions in reverse order: $A_2 \rightleftharpoons A_1, A_2 \rightleftharpoons A_0, A_1 \rightleftharpoons A_0$. Define $A_{i,k+1}^{s2}$ by the obtained approximation $A_{i,k+1}$ for $i = 1, 2, 3$.
 - (d) Set

$$A_{i,k+1}^s = \frac{A_{i,k+1}^{s1} + A_{i,k+1}^{s2}}{2} \text{ for } i = 1, 2, 3.$$

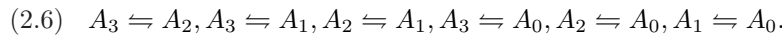
Remark 2.2. For reaction systems with more than three substances, there are two different ways to implement the reverse ordering in the step (2)-(c) of the Algorithm 2; one is to reverse the order of reversible reactions in the step (2)-(b) and the other is to put the substances used in the step (2)-(b) in reverse order. In other words, we suppose that in a reaction system with four substances, we solve the six reversible reactions in the following order



Then, at our convenience we adopt one of two reverse ordering



and



Remark 2.3. Algorithm 2 is a symmetrized version of Algorithm 1 and can be applied to n substances inductively as in the Algorithm 1.

3. Stability and convergence results

Now, we consider a linear reaction system with $(n + 1)$ substances:

$$(3.1) \quad \frac{d}{dt} \begin{bmatrix} A_0(t) \\ A_1(t) \\ \vdots \\ A_n(t) \end{bmatrix} = M_n \begin{bmatrix} A_0(t) \\ A_1(t) \\ \vdots \\ A_n(t) \end{bmatrix},$$

where the matrix M_n with entries $\{m_{ij}\}_{i,j=1}^{n+1}$ satisfies

$$m_{ii} < 0 \quad \forall i, \quad m_{ij} \geq 0 \quad \forall i \neq j, \quad \sum_{i=1}^{n+1} m_{ij} = 0 \quad \forall j.$$

Remark 3.1. The assumption $\sum_i m_{ij} = 0$ enforces the concerned reaction system to be conserved. In fact, the requirement of the conserved system can be dropped since the non-conserved system can be converted into the conserved one by adding an artificial substance. In Section 4, we state in more details how to extend the proposed algorithm to the non-conserved systems.

Focusing on the individual reversible reactions between two substances yields a detailed form of M_n as follows:

$$M_n = \begin{bmatrix} -\sum_{i=1}^n b_{i0} & f_{10} & \cdots & f_{(n-1)0} & f_{n0} \\ b_{10} & -f_{10} - \sum_{i=2}^n b_{i1} & \cdots & f_{(n-1)1} & f_{n1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{(n-1)0} & b_{(n-1)1} & \cdots & -\sum_{j=0}^{n-2} f_{(n-1)j} - b_{n(n-1)} & f_{n(n-1)} \\ b_{n0} & b_{n1} & \cdots & b_{n(n-1)} & -\sum_{j=0}^{n-1} f_{nj} \end{bmatrix}.$$

Note that the linear reaction system (3.1) includes a chain of reversible reactions and circular reactions. We can easily confirm that the real parts of all eigenvalues of M_n are nonpositive. Hence the solution for the problem (3.1) is bounded for any $t > 0$.

Definition. A numerical method for the problem like (3.1) is said to be *stable* if it does not blow up as $t \rightarrow \infty$. Furthermore, a numerical method is called *absolutely stable* if it is stable without any condition on the step size Δt .

3.1. Stability and convergence of CR² for linear reaction systems

When applying the CR² algorithm to (3.1), we get the approximate solution $(A_{i,k+1})_{i=0}^n$ such that

$$(3.2) \quad \begin{bmatrix} A_{0,k+1} \\ A_{1,k+1} \\ \vdots \\ A_{n,k+1} \end{bmatrix} = L_n \begin{bmatrix} A_{0,k} \\ A_{1,k} \\ \vdots \\ A_{n,k} \end{bmatrix},$$

where

$$(3.3) \quad L_n = \left(e^{\Delta t \bar{M}_{n(n-1)}} \dots e^{\Delta t \bar{M}_{n0}} \right) \dots \left(e^{\Delta t \bar{M}_{21}} e^{\Delta t \bar{M}_{20}} \right) e^{\Delta t \bar{M}_{10}}$$

and \bar{M}_{ij} is an $(n + 1) \times (n + 1)$ matrix with entries in (2.4). The expression of the form

$$(3.4) \quad \prod_{\substack{i,j=0 \\ i>j}}^n \exp(\Delta t \bar{M}_{ij})$$

formally implies L_n in (3.3).

Let $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$ be a norm on \mathbb{R}^n .

Theorem 3.2. *The algorithm CR² applied to linear reaction systems is absolutely stable.*

Proof. The CR² algorithm gives an approximate solution to (3.1) such that $\mathbf{y}^{k+1} = L_n \mathbf{y}^k$ with $\mathbf{y}^k = (A_{0,k} \dots A_{n,k})^T$. Note that each matrix operator $\exp(\Delta t \bar{M}_{ij})$ in (3.3) conserves the total concentration of reaction substances since the action of $\exp(\Delta t \bar{M}_{ij})$ is characterized by a reversible reaction between two substances in a closed system. Hence, it follows that

$$(3.5) \quad \|\mathbf{y}^{k+1}\|_1 = \|\mathbf{y}^k\|_1,$$

that is, the approximation does not blow up independently of Δt . □

Remark 3.3. We shall compare the stability of CR² with that of typical numerical ODE solvers in more details. In order to analyze numerical methods for ODEs, a simple scalar ODE

$$(3.6) \quad y'(t) = \lambda y(t)$$

is commonly-used since it is very helpful in predicting the stability behavior of numerical schemes. In such a typical stability analysis with (3.6), two stability properties, *A-stability* and *L-stability* are used especially for the stiff problems [3]. The stability function $R(z)$ is a major tool to measure the stability behavior of numerical solutions. But, since our algorithm is first aimed for a numerical solution to a reaction system not a scalar ODE, it is not easy to adopt directly the analysis techniques based on the stability function. Note that the *A-stability* and *L-stability* properties were originally introduced as criteria which show how well numerical solutions mimic important properties of the exact solution in form of exponential function. In this context, the algorithm CR² might be regarded as an *A-stable* method in the sense that the numerical solution by CR² is bounded independently of both the step size Δt and the spectrum of M_n . In addition, the *L-stability* is essential to a numerical method for stiff ODEs, which makes stiff components of a numerical solution damped out rapidly. A numerical method is said to be *L-stable* if $\lim_{z \rightarrow \infty} |R(z)| = 0$. Focusing on the main role of *L-stability*, it is sufficient to verify how well the proposed

CR^2 maintains the rapidly decaying property of the exact solution to the stiff problems. In a reaction system (3.1), the stiffness ratio of the problem is in connection with difference in magnitude of reaction parameters. Assuming that the only one specific pair $A_{j_s} \rightleftharpoons A_{i_s}$ of reversible reaction reacts very fast relative to the other parts of reactions in (3.1), that is, the problem is stiff, the exact solution includes such a rapidly decaying component. Based on the structure of the CR^2 in (3.3), the operator $\exp(\Delta t \bar{M}_{i_s j_s})$ corresponding to the reversible reaction between A_{j_s} and A_{i_s} makes the approximation by CR^2 preserve well the stiff component of the exact solution.

Let $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$ be a matrix norm on $\mathbb{R}^{n \times n}$.

Theorem 3.4. *The algorithm CR^2 applied to (3.1) has the order of convergence 1.*

Proof. We denote the exact solution of (3.1) by $\mathbf{y}(t) = (A_0(t) \cdots A_n(t))^T$. The CR^2 gives an approximation to (3.1) such that $\mathbf{y}^{k+1} = L_n \mathbf{y}^k$ with $\mathbf{y}^k = (A_{0,k} \cdots A_{n,k})^T$. We will show that

$$\|\mathbf{y}(t_{k+1}) - \mathbf{y}^{k+1}\|_1 \leq C \Delta t,$$

where C is a constant independent of Δt . Using Taylor's theorem yields

$$(3.7) \quad \mathbf{y}(t_{k+1}) = \left(I + \Delta t M_n + \frac{(\Delta t)^2}{2} M_n^2 \right) \mathbf{y}(t_k) + O((\Delta t)^3).$$

Recalling the definition of the matrix exponential in terms of power series, L_n is rewritten as

$$\begin{aligned} L_n &= \prod_{\substack{i,j=0 \\ i>j}}^n (I + \Delta t \bar{M}_{ij} + O((\Delta t)^2)) \\ &= I + \Delta t M_n + (\Delta t)^2 L_{n,2} + O((\Delta t)^3), \end{aligned}$$

where \prod is formally used the same as in (3.4) and the second equality is given by the fact that

$$M_n = \sum_{\substack{i,j=0 \\ i>j}}^n \bar{M}_{ij}.$$

Hence, it follows that

$$(3.8) \quad \mathbf{y}^{k+1} = (I + \Delta t M_n + (\Delta t)^2 L_{n,2}) \mathbf{y}^k + O((\Delta t)^3).$$

Let $\mathbf{e}_k = \mathbf{y}(t_k) - \mathbf{y}^k$ denote the numerical error. Subtracting (3.8) from (3.7) gives

$$\begin{aligned} \mathbf{e}_{k+1} &= (I + \Delta t M_n + (\Delta t)^2 L_{n,2}) \mathbf{e}_k + \left(\frac{M_n^2}{2} - L_{n,2} \right) (\Delta t)^2 \mathbf{y}(t_k) + O((\Delta t)^3) \\ &= L_n \mathbf{e}_k + O((\Delta t)^2). \end{aligned}$$

Since $\|L_n\|_1 = 1$ by (3.5), it follows that

$$\|\mathbf{e}_{k+1}\|_1 \leq \|\mathbf{e}_k\|_1 + C(\Delta t)^2$$

with a constant C independent of Δt and k . We can check by induction on k that

$$\|\mathbf{e}_k\|_1 \leq kC(\Delta t)^2 \quad \forall k = 0, 1, \dots$$

If we restrict t to the finite interval $[0, t_F]$, then we have that

$$\begin{aligned} \|\mathbf{e}_k\|_1 &\leq \lfloor t_F/\Delta t \rfloor C(\Delta t)^2 \\ &\leq Ct_F \Delta t, \quad \forall k = 0, 1, \dots, \lfloor t_F/\Delta t \rfloor. \end{aligned}$$

Since C is independent of Δt , it follows that

$$\lim_{\substack{\Delta t \rightarrow 0 \\ 0 \leq k \leq \lfloor t_F/\Delta t \rfloor}} \|\mathbf{e}_k\|_1 = 0.$$

In other words, the algorithm CR^2 is convergent and has the order of convergence 1. \square

3.2. Convergence of SCR^2 for linear reaction systems

Applying the SCR^2 to (3.1) gives

$$\begin{bmatrix} A_{0,k+1}^{s_1} \\ A_{1,k+1}^{s_1} \\ \vdots \\ A_{n,k+1}^{s_1} \end{bmatrix} = L_n^{s_1} \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ \vdots \\ A_{n,k}^s \end{bmatrix}, \quad \begin{bmatrix} A_{0,k+1}^{s_2} \\ A_{1,k+1}^{s_2} \\ \vdots \\ A_{n,k+1}^{s_2} \end{bmatrix} = L_n^{s_2} \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ \vdots \\ A_{n,k}^s \end{bmatrix}.$$

Then, we have

$$(3.9) \quad \begin{bmatrix} A_{0,k+1}^s \\ A_{1,k+1}^s \\ \vdots \\ A_{n,k+1}^s \end{bmatrix} = L_n^s \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ \vdots \\ A_{n,k}^s \end{bmatrix} = \frac{1}{2}(L_n^{s_1} + L_n^{s_2}) \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ \vdots \\ A_{n,k}^s \end{bmatrix}.$$

As mentioned in Remark 2.2, for reaction systems with more than three substances, there are two possible ways to choose the reverse order used in implementing SCR^2 . Note that how to reverse the order of reactions makes a difference in the relationship between $L_n^{s_1}$ and $L_n^{s_2}$. Hence, we shall analyze the algorithm SCR^2 in two different ways.

First, let us handle the algorithm with the reverse order (2.5).

Lemma 3.5. *If the algorithm SCR^2 is in company with the reverse order (2.5), the matrix L_n^s in (3.9) has the Taylor expansion*

$$\begin{aligned} L_n^s &= L_{n,0}^s + L_{n,1}^s \Delta t + L_{n,2}^s (\Delta t)^2 + O((\Delta t)^3) \\ &= I + M_n \Delta t + \frac{1}{2} M_n^2 (\Delta t)^2 + O((\Delta t)^3), \end{aligned}$$

where M_n is the matrix in (3.1).

Proof. Since SCR^2 is a symmetrized version of CR^2 , each $L_n^{s_i}$ has the same property as L_n in CR^2 . Thus, it is obvious that

$$L_{n,0}^s = I, \quad L_{n,1}^s = M_n.$$

For a linear reaction system with $(n+1)$ substances, recall

$$L_n^{s_1} = \left(e^{\Delta t \bar{M}_{n(n-1)}} \dots e^{\Delta t \bar{M}_{n0}} \right) \dots \left(e^{\Delta t \bar{M}_{21}} e^{\Delta t \bar{M}_{20}} \right) e^{\Delta t \bar{M}_{10}}.$$

According to the reverse order in (2.5), we have

$$L_n^{s_2} = e^{\Delta t \bar{M}_{10}} \left(e^{\Delta t \bar{M}_{20}} e^{\Delta t \bar{M}_{21}} \right) \dots \left(e^{\Delta t \bar{M}_{n0}} \dots e^{\Delta t \bar{M}_{n(n-1)}} \right).$$

Based on the fact that

$$M_n = \sum_{\substack{i,j=0 \\ i>j}}^n \bar{M}_{ij},$$

the power series expansion of each $L_n^{s_i}$ yields

$$L_{n,2}^s = \frac{1}{2} M_n^2. \quad \square$$

Next, we consider the SCR^2 equipped with the reverse order (2.6), where the reaction substances are put in reverse order. In order to look into the procedure of the SCR^2 algorithm applied to a reaction system (3.1), we first consider a circular reaction with 3 substances shown in Figure 1. Two parallel solution steps in SCR^2 applied to the reaction system is represented as

$$\begin{bmatrix} A_{0,k+1}^{s_1} \\ A_{1,k+1}^{s_1} \\ A_{2,k+1}^{s_1} \end{bmatrix} = L_2^{s_1} \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ A_{2,k}^s \end{bmatrix}, \quad \begin{bmatrix} A_{2,k+1}^{s_2} \\ A_{1,k+1}^{s_2} \\ A_{0,k+1}^{s_2} \end{bmatrix} = \bar{L}_2^{s_2} \begin{bmatrix} A_{2,k}^s \\ A_{1,k}^s \\ A_{0,k}^s \end{bmatrix}.$$

Then, we have

$$\begin{bmatrix} A_{0,k+1}^s \\ A_{1,k+1}^s \\ A_{2,k+1}^s \end{bmatrix} = \frac{1}{2} (L_2^{s_1} + \bar{L}_2^{s_2}) \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ A_{2,k}^s \end{bmatrix},$$

where

$$L_2^{s_2} = P_2 \bar{L}_2^{s_2} P_2, \quad P_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Considering Taylor expansions of $L_2^{s_1}$ and $\bar{L}_2^{s_2}$ yields

$$L_{2,2}^{s_1} - \frac{1}{2} M_2^2 = \frac{1}{2} \begin{bmatrix} 0 & -b_{20}f_{10} + b_{21}f_{10} - b_{21}f_{20} & \text{(I)} \\ b_{10}b_{20} - b_{10}b_{21} + b_{20}f_{21} & 0 & \text{(II)} \\ -b_{10}b_{20} + b_{10}b_{21} - b_{20}f_{21} & b_{20}f_{10} - b_{21}f_{10} + b_{21}f_{20} & 0 \end{bmatrix}$$

and

$$\bar{L}_{2,2}^{s_2} - \frac{1}{2} P_2 M_2^2 P_2 = \frac{1}{2} \begin{bmatrix} 0 & -b_{20}f_{10} + b_{21}f_{10} - b_{21}f_{20} & \text{(III)} \\ b_{10}f_{20} - f_{10}f_{21} + f_{20}f_{21} & 0 & \text{(IV)} \\ -b_{10}f_{20} + f_{10}f_{21} - f_{20}f_{21} & b_{20}f_{10} - b_{21}f_{10} + b_{21}f_{20} & 0 \end{bmatrix},$$

where

$$\begin{aligned} \text{(I)} &= b_{10}f_{20} - f_{10}f_{21} + f_{20}f_{21} & \text{(II)} &= -b_{10}f_{20} + f_{10}f_{21} - f_{20}f_{21} \\ \text{(III)} &= b_{10}b_{20} - b_{10}b_{21} + b_{20}f_{21} & \text{(IV)} &= -b_{10}b_{20} + b_{10}b_{21} - b_{20}f_{21}. \end{aligned}$$

Then, we have

$$(3.10) \quad L_{2,2}^{s_1} + L_{2,2}^{s_2} - M_2^2 = 0.$$

In general, the approximation $(A_{i,k+1}^s)_{i=0}^n$ obtained by SCR² applied to (3.1) is characterized as

$$(3.11) \quad \begin{bmatrix} A_{0,k+1}^s \\ A_{1,k+1}^s \\ \vdots \\ A_{n,k+1}^s \end{bmatrix} = L_n^s \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ \vdots \\ A_{n,k}^s \end{bmatrix} = \frac{1}{2}(L_n^{s_1} + L_n^{s_2}) \begin{bmatrix} A_{0,k}^s \\ A_{1,k}^s \\ \vdots \\ A_{n,k}^s \end{bmatrix},$$

where

$$L_n^{s_2} = P_n \bar{L}_n^{s_2} P_n$$

and P_n is the $(n + 1) \times (n + 1)$ permutation matrix such that

$$(P_n)_{ij} = \begin{cases} 1 & \text{if } i + j = n + 2, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 3.6. *Consider a parallel reaction with $(n + 1)$ substances, that is composed of n reversible reactions:*



\vdots



Applying the CR² algorithm to this problem in the following ordering:

$$A_0 \rightleftharpoons A_n, A_1 \rightleftharpoons A_n, \dots, A_{n-1} \rightleftharpoons A_n$$

gives

$$\begin{bmatrix} A_{0,k+1} \\ A_{1,k+1} \\ \vdots \\ A_{n,k+1} \end{bmatrix} = \Lambda_n \begin{bmatrix} A_{0,k} \\ A_{1,k} \\ \vdots \\ A_{n,k} \end{bmatrix},$$

where $(A_{i,k+1})_{i=0}^n$ is an approximation to $(A_i(t))_{i=0}^n$ at $t = t_{k+1}$ and Λ_n is an $(n + 1) \times (n + 1)$ matrix. Then, we have

$$\Lambda_n = \begin{bmatrix} & & & f_{n0}\Delta t \\ & \text{diag}(1 - b_{nj}\Delta t)_{j=0}^{n-1} & & \vdots \\ & & & f_{n(n-1)}\Delta t \\ b_{n0}\Delta t & \cdots & b_{n(n-1)}\Delta t & 1 - (\sum_{j=0}^{n-1} f_{nj})\Delta t \end{bmatrix} + O((\Delta t)^2).$$

Proof. It is easy to show that the statement holds for $n = 2$ by using Taylor's theorem. Suppose that the statement is true for a parallel reaction which consists of n substances. Consider a parallel reaction with $(n + 1)$ substances. By the inductive hypothesis, the approximate solution $(A_{i,k+1})_{i=0}^n$ to the solution $(A_i(t))_{i=0}^n$ at $t = t_{k+1}$ obtained by applying CR² to (3.12) is written as follows:

$$(3.13a) \quad \begin{bmatrix} A_{0,k+1} \\ A_n^{temp} \end{bmatrix} = \begin{bmatrix} 1 - b_{n0}\Delta t + O((\Delta t)^2) & f_{n0}\Delta t + O((\Delta t)^2) \\ b_{n0}\Delta t + O((\Delta t)^2) & 1 - f_{n0}\Delta t + O((\Delta t)^2) \end{bmatrix} \begin{bmatrix} A_{0,k} \\ A_{n,k} \end{bmatrix}$$

$$(3.13b) \quad \begin{bmatrix} A_{1,k+1} \\ \vdots \\ A_{n-1,k+1} \\ A_{n,k+1} \end{bmatrix} = \Lambda_{n-1} \begin{bmatrix} A_{1,k} \\ \vdots \\ A_{n-1,k} \\ A_n^{temp} \end{bmatrix},$$

where

$$(3.14) \quad \Lambda_{n-1} = \begin{bmatrix} & & & f_{n1}\Delta t \\ & \text{diag}(1 - b_{nj}\Delta t)_{j=1}^{n-1} & & \vdots \\ & & & f_{n(n-1)}\Delta t \\ b_{n1}\Delta t & \cdots & b_{n(n-1)}\Delta t & 1 - (\sum_{j=1}^{n-1} f_{nj})\Delta t \end{bmatrix} + O((\Delta t)^2).$$

By (3.13), we have

$$(3.15) \quad \Lambda_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & & \\ 0 & & \Lambda_{n-1} & \end{bmatrix} \begin{bmatrix} 1 - b_{n0}\Delta t + O((\Delta t)^2) & 0 & \cdots & 0 & f_{n0}\Delta t + O((\Delta t)^2) \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ b_{n0}\Delta t + O((\Delta t)^2) & 0 & \cdots & 0 & 1 - f_{n0}\Delta t + O((\Delta t)^2) \end{bmatrix}.$$

By substituting (3.14) into (3.15), we have the expansion form of Λ_n , that is, the statement is true for a parallel reaction with $(n + 1)$ substances. \square

Lemma 3.7. *Under the same assumptions as in Lemma 3.6, we assume that*

$$\Lambda_n = \Lambda_{n,0} + \Lambda_{n,1}\Delta t + \Lambda_{n,2}(\Delta t)^2 + O((\Delta t)^3).$$

Then,

$$\Lambda_{n,2} = \begin{bmatrix} \frac{b_{n0}}{2}(b_{n0} + f_{n0}) & 0 & 0 & \cdots & 0 & \text{(I)} \\ b_{n0}f_{n1} & \frac{b_{n1}}{2}(b_{n1} + f_{n1}) & 0 & \ddots & \vdots & \text{(II)} \\ b_{n0}f_{n2} & b_{n1}f_{n2} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & 0 & \vdots \\ b_{n0}f_{n(n-1)} & b_{n1}f_{n(n-1)} & \cdots & \cdots & \frac{b_{n(n-1)}}{2}(b_{n(n-1)} + f_{n(n-1)}) & \text{(III)} \\ \text{(IV)} & \text{(V)} & \cdots & \cdots & -\frac{b_{n(n-1)}}{2}(b_{n(n-1)} + f_{n(n-1)}) & \text{(VI)} \end{bmatrix},$$

where

$$\begin{aligned} \text{(I)} &= -\frac{f_{n0}}{2}(b_{n0} + f_{n0}) & \text{(II)} &= -\frac{f_{n1}}{2}(2f_{n0} + b_{n1} + f_{n1}) \\ \text{(III)} &= -\frac{f_{n(n-1)}}{2}\left(2\sum_{i=0}^{n-2} f_{ni} + b_{n(n-1)} + f_{n(n-1)}\right) & \text{(IV)} &= -\frac{b_{n0}}{2}(b_{n0} + f_{n0}) + 2\sum_{i=1}^{n-1} f_{ni} \\ \text{(V)} &= -\frac{b_{n1}}{2}(b_{n1} + f_{n1}) + 2\sum_{i=2}^{n-2} f_{ni} & \text{(VI)} &= \frac{1}{2}\sum_{i=0}^{n-1} f_{ni}(b_{ni} + f_{ni}) + \sum_{i < j} f_{ni}f_{nj}. \end{aligned}$$

Proof. By the Taylor expansion of Λ_2 , the statement holds for $n = 2$. Suppose that the statement is true for $\Lambda_{n-1,2}$. By Lemma 3.6, we have

$$\Lambda_{n-1} = I + \begin{bmatrix} -b_{n1} & & & f_{n1} \\ & \ddots & & \vdots \\ & & -b_{n(n-1)} & f_{n(n-1)} \\ b_{n1} & \cdots & b_{n(n-1)} & -\sum_{i=1}^{n-1} f_{ni} \end{bmatrix} \Delta t + \Lambda_{n-1,2}(\Delta t)^2 + O((\Delta t)^3).$$

It follows

$$\begin{aligned} \Lambda_{n,2} &= \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_{n-1,2} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & -b_{n1} & & & f_{n1} \\ \vdots & & \ddots & & \vdots \\ 0 & & & -b_{n(n-1)} & f_{n(n-1)} \\ 0 & b_{n1} & \cdots & b_{n(n-1)} & -\sum_{i=1}^{n-1} f_{ni} \end{bmatrix} \begin{bmatrix} -b_{n0} & & & f_{n0} \\ & 0 & & \\ & & \ddots & \\ b_{n0} & & & 0 \\ & & & & -f_{n0} \end{bmatrix} \\ &+ \frac{1}{2} \begin{bmatrix} b_{n0}(b_{n0} + f_{n0}) & & & -f_{n0}(b_{n0} + f_{n0}) \\ & 0 & & \\ & & \ddots & \\ -b_{n0}(b_{n0} + f_{n0}) & & & f_{n0}(b_{n0} + f_{n0}) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_{n-1,2} \end{bmatrix} + \begin{bmatrix} \frac{b_{n0}}{2}(b_{n0} + f_{n0}) & 0 & \cdots & 0 & -\frac{f_{n0}}{2}(b_{n0} + f_{n0}) \\ b_{n0}f_{n1} & 0 & \cdots & 0 & -f_{n0}f_{n1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n0}f_{n(n-1)} & 0 & \cdots & 0 & -f_{n0}f_{n(n-1)} \\ \text{(I)} & 0 & \cdots & 0 & \text{(II)} \end{bmatrix}, \end{aligned}$$

where

$$(I) = -\frac{b_{n0}}{2} \left(b_{n0} + f_{n0} + 2 \sum_{i=1}^{n-1} f_{ni} \right), \quad (II) = \frac{f_{n0}}{2} \left(b_{n0} + f_{n0} + 2 \sum_{i=1}^{n-1} f_{ni} \right).$$

Hence, by the inductive hypothesis, we conclude that the statement is true for $\Lambda_{n,2}$. \square

Lemma 3.8. *If the algorithm SCR^2 is in company with the reverse order (2.6), the matrix L_n^s in (3.11) has the Taylor expansion*

$$\begin{aligned} L_n^s &= L_{n,0}^s + L_{n,1}^s \Delta t + L_{n,2}^s (\Delta t)^2 + O((\Delta t)^3) \\ &= I + M_n \Delta t + \frac{1}{2} M_n^2 (\Delta t)^2 + O((\Delta t)^3), \end{aligned}$$

where M_n is the matrix in (3.1).

Proof. As mentioned in the proof of Lemma 3.5, it is obvious that

$$L_{n,0}^s = I, \quad L_{n,1}^s = M_n.$$

Let $D_n^{s_i} = L_{n,2}^{s_i} - \frac{1}{2} M_n^2$ for $i = 1, 2$. To complete the proof, it suffices to show that

$$D_n^{s_1} + D_n^{s_2} = 0.$$

Because $D_n^{s_2}$ can be associated with $D_n^{s_1}$ by the permutation matrix P_n , we only need to characterize entries of $D_n^{s_1}$.

We first claim that

$$(3.16) \quad D_n^{s_1} = \sum_{i < j < k} D_{(i,j,k)}, \quad 0 \leq i, j, k \leq n,$$

where

$$D_{(i,j,k)} = \frac{1}{2} \begin{bmatrix} \begin{matrix} (i+1)\text{th} \\ \downarrow \\ 0 \end{matrix} & \begin{matrix} (j+1)\text{th} \\ \downarrow \\ -b_{ki}f_{ji} + b_{kj}f_{ji} - b_{kj}f_{ki} \end{matrix} & \begin{matrix} (k+1)\text{th} \\ \downarrow \\ (I) \end{matrix} \\ \begin{matrix} b_{ji}b_{ki} - b_{ji}b_{kj} + b_{ki}f_{kj} \\ -b_{ji}b_{ki} + b_{ji}b_{kj} - b_{ki}f_{kj} \end{matrix} & 0 & \begin{matrix} (II) \\ 0 \end{matrix} \end{bmatrix} \begin{matrix} \leftarrow (i+1)\text{th} \\ \leftarrow (j+1)\text{th} \\ \leftarrow (k+1)\text{th} \end{matrix}$$

with

$$(I) = b_{ji}f_{ki} - f_{ji}f_{kj} + f_{ki}f_{kj}, \quad (II) = -b_{ji}f_{ki} + f_{ji}f_{kj} - f_{ki}f_{kj}$$

and other entries are zeros. The claim is verified by induction on n . By the previous argument, we already know that

$$D_2^{s_1} = D_{(0,1,2)}.$$

We assume that the claim is true for a general reaction system with n substances A_0, A_1, \dots, A_{n-1} , i.e.,

$$D_{n-1}^{s_1} = \sum_{i < j < k} D_{(i,j,k)}, \quad 0 \leq i, j, k \leq n-1.$$

Let us see how $L_n^{s_1}$ and M_n change when adding A_n into the general reaction system with n substances. Since

$$\begin{aligned} L_n^{s_1} &= \Lambda_n \begin{bmatrix} L_{n-1}^{s_1} & 0 \\ 0 & 1 \end{bmatrix} \\ &= (I + \Lambda_{n,1}\Delta t + \Lambda_{n,2}(\Delta t)^2 + O((\Delta t)^3)) \begin{bmatrix} I + M_{n-1}\Delta t + L_{n-1,2}^{s_1}(\Delta t)^2 + O((\Delta t)^3) & 0 \\ 0 & 1 \end{bmatrix}, \end{aligned}$$

it follows that

$$L_{n,2}^{s_1} = \begin{bmatrix} L_{n-1,2}^{s_1} & 0 \\ 0 & 0 \end{bmatrix} + \Lambda_{n,1} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} + \Lambda_{n,2}.$$

Also, noting the fact that

$$M_n = \begin{bmatrix} & & & f_{n0} \\ & M_{n-1} - \text{diag}(b_{nj})_{j=0}^{n-1} & & \vdots \\ & & \dots & f_{n(n-1)} \\ b_{n0} & & & -\sum_{j=0}^{n-1} f_{nj} \end{bmatrix},$$

Lemma 3.6 gives

$$\begin{aligned} M_n^2 &= \left(\begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} + \Lambda_{n,1} \right)^2 \\ &= \begin{bmatrix} M_{n-1}^2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} \Lambda_{n,1} + \Lambda_{n,1} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} + \Lambda_{n,1}^2. \end{aligned}$$

Then, we have that

$$\begin{aligned} D_n^{s_1} &= L_{n,2}^{s_1} - \frac{1}{2}M_n^2 \\ &= \begin{bmatrix} L_{n-1,2}^{s_1} - \frac{1}{2}M_{n-1}^2 & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{2}\Lambda_{n,1} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} \Lambda_{n,1} + \Lambda_{n,2} - \frac{1}{2}\Lambda_{n,1}^2. \end{aligned}$$

Looking at M_{n-1} , $\Lambda_{n,1}$ and $\Lambda_{n,2}$ carefully, all (i, j) entries contain subindex n , $(i-1)$ or $(j-1)$. It implies that we can write

$$\frac{1}{2}\Lambda_{n,1} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} \Lambda_{n,1} + \Lambda_{n,2} - \frac{1}{2}\Lambda_{n,1}^2 = \sum_{i < j} B_{ijn} \quad 0 \leq i, j < n,$$

where B_{ijn} is a matrix whose entries consist of products of b_{ni} , f_{ni} , b_{nj} , f_{nj} , b_{ji} , and f_{ji} . Now let us compute B_{ijn} in details. For a matrix M , we define $(M)_{ijn}$ by a matrix whose entries consist of products of b_{ni} , f_{ni} , b_{nj} , f_{nj} , b_{ji} and f_{ji} among entries of M . We detail B_{ijn} for all i, j with $0 \leq i, j < n$ as follows:

$$\begin{aligned} E_1 &= \left(\Lambda_{n,1} \begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} \right)_{ijn} \\ &= \begin{array}{ccc} \begin{matrix} (i+1)\text{th} \\ \downarrow \\ b_{ji}b_{ni} \end{matrix} & \begin{matrix} (j+1)\text{th} \\ \downarrow \\ -b_{ni}f_{ji} \end{matrix} & \begin{matrix} (n+1)\text{th} \\ \downarrow \\ 0 \end{matrix} \\ \begin{matrix} -b_{ji}b_{nj} \\ b_{ji}(b_{nj} - b_{ni}) \end{matrix} & \begin{matrix} b_{nj}f_{ji} \\ f_{ji}(b_{ni} - b_{nj}) \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \end{array} \begin{array}{l} \leftarrow (i+1)\text{th} \\ \leftarrow (j+1)\text{th} \\ \leftarrow (n+1)\text{th} \end{array} \end{array}$$

$$\begin{aligned}
E_2 &= \left(\begin{bmatrix} M_{n-1} & 0 \\ 0 & 0 \end{bmatrix} \Lambda_{n,1} \right)_{ijn} \\
&= \begin{array}{ccc} (i+1)\text{th} & (j+1)\text{th} & (n+1)\text{th} \\ \downarrow & \downarrow & \downarrow \\ \begin{bmatrix} b_{ji}b_{ni} & -f_{ji}b_{nj} & -b_{ji}f_{ni} + f_{ji}f_{nj} \\ -b_{ji}b_{ni} & f_{ji}b_{nj} & b_{ji}f_{ni} - f_{ji}f_{nj} \\ 0 & 0 & 0 \end{bmatrix} & \leftarrow (i+1)\text{th} \\ & \leftarrow (j+1)\text{th} \\ & \leftarrow (n+1)\text{th} \end{array} \\
E_3 &= (\Lambda_{n,2})_{ijn} \\
&= \begin{array}{ccc} (i+1)\text{th} & (j+1)\text{th} & (n+1)\text{th} \\ \downarrow & \downarrow & \downarrow \\ \begin{bmatrix} \frac{1}{2(n-1)}b_{ni}(b_{ni} + f_{ni}) & 0 & \text{(I)} \\ b_{ni}f_{nj} & \frac{1}{2(n-1)}b_{nj}(b_{nj} + f_{nj}) & \text{(II)} \\ -\frac{1}{2(n-1)}b_{ni}(b_{ni} + f_{ni}) - b_{ni}f_{nj} & -\frac{1}{2(n-1)}b_{nj}(b_{nj} + f_{nj}) & \text{(III)} \end{bmatrix} & \leftarrow (i+1)\text{th} \\ & \leftarrow (j+1)\text{th} \\ & \leftarrow (n+1)\text{th} \end{array} \\
\text{(I)} &= -\frac{1}{2(n-1)}f_{ni}(b_{ni} + f_{ni}) \\
\text{(II)} &= -\frac{1}{2(n-1)}f_{nj}(b_{nj} + f_{nj}) - f_{ni}f_{nj} \\
\text{(III)} &= \frac{1}{2(n-1)}f_{ni}(b_{ni} + f_{ni}) + \frac{1}{2(n-1)}f_{nj}(b_{nj} + f_{nj}) + f_{ni}f_{nj}.
\end{aligned}$$

Note that $(n-1)$ appears in the denominators. In the $(i+1, i+1)$ entry, $\frac{1}{2}b_{ni}(b_{ni} + f_{ni})$ is independent of j . Therefore, we distribute it equally to all $(\Lambda_{n,2})_{ijn}$ of which $(i+1)$ th column is nonzero. In other entries, $(n-1)$ appears with the same reason.

$$\begin{aligned}
E_4 &= \left((\Lambda_{n,1})^2 \right)_{ijn} \\
&= \begin{array}{ccc} (i+1)\text{th} & (j+1)\text{th} & (n+1)\text{th} \\ \downarrow & \downarrow & \downarrow \\ \begin{bmatrix} \frac{1}{(n-1)}b_{ni}(b_{ni} + f_{ni}) & b_{nj}f_{ni} & \text{(I)} \\ b_{ni}f_{nj} & \frac{1}{(n-1)}b_{nj}(b_{nj} + f_{nj}) & \text{(II)} \\ -\frac{1}{(n-1)}b_{ni}(b_{ni} + f_{ni}) - b_{ni}f_{nj} & \text{(III)} & \text{(IV)} \end{bmatrix} & \leftarrow (i+1)\text{th} \\ & \leftarrow (j+1)\text{th} \\ & \leftarrow (n+1)\text{th} \end{array} \\
\text{(I)} &= -\frac{1}{(n-1)}f_{ni}(b_{ni} + f_{ni}) - f_{ni}f_{nj} \\
\text{(II)} &= -\frac{1}{(n-1)}f_{nj}(b_{nj} + f_{nj}) - f_{ni}f_{nj} \\
\text{(III)} &= -\frac{1}{(n-1)}b_{nj}(b_{nj} + f_{nj}) - b_{nj}f_{ni} \\
\text{(IV)} &= \frac{1}{(n-1)}f_{ni}(b_{ni} + f_{ni}) + \frac{1}{(n-1)}f_{nj}(b_{nj} + f_{nj}) + 2f_{ni}f_{nj},
\end{aligned}$$

where for each E_l , $E_l(m_1, m_2) = 0 \quad \forall m_1, m_2 \notin \{i+1, j+1, n+1\}$. Then, we have

$$\begin{aligned}
B_{ijn} &= \frac{1}{2}E_1 - \frac{1}{2}E_2 + E_3 - \frac{1}{2}E_4 \\
&= \frac{1}{2} \begin{array}{ccc} (i+1)\text{th} & (j+1)\text{th} & (n+1)\text{th} \\ \downarrow & \downarrow & \downarrow \\ \begin{bmatrix} 0 & -b_{ni}f_{ji} + b_{nj}f_{ji} - b_{nj}f_{ni} & \text{(I)} \\ b_{ji}b_{ni} - b_{ji}b_{nj} + b_{ni}f_{nj} & 0 & \text{(II)} \\ -b_{ji}b_{ni} + b_{ji}b_{nj} - b_{ni}f_{nj} & b_{ni}f_{ji} - b_{nj}f_{ji} + b_{nj}f_{ni} & 0 \end{bmatrix} & \leftarrow (i+1)\text{th} \\ & \leftarrow (j+1)\text{th} \\ & \leftarrow (n+1)\text{th} \end{array} \\
&= D_{(i,j,n)},
\end{aligned}$$

where

$$(I) = b_{ji}f_{ni} - f_{ji}f_{nj} + f_{ni}f_{nj}, \quad (II) = -b_{ji}f_{ni} + f_{ji}f_{nj} - f_{ni}f_{nj}.$$

The inductive hypothesis yields

$$\begin{aligned} D_n^{s_1} &= \sum_{i < j < k < n} D_{(i,j,k)} + \sum_{i < j < n} D_{(i,j,n)} \\ &= \sum_{i < j < k} D_{(i,j,k)}. \end{aligned}$$

Similarly, we obtain the relationship

$$\bar{L}_{n,2}^{s_2} - \frac{1}{2}P_n M_n^2 P_n = \sum_{i < j < k} D_{(n-k,n-j,n-i)}.$$

Note that

$$D_{(i,j,k)} + P_n D_{(n-k,n-j,n-i)} P_n = 0.$$

Consequently, we find that

$$\begin{aligned} D_n^{s_2} &= L_{n,2}^{s_2} - \frac{1}{2}M_n^2 \\ &= P_n(\bar{L}_{n,2}^{s_2} - \frac{1}{2}P_n M_n^2 P_n)P_n \\ &= P_n\left(\sum_{i < j < k} D_{(n-k,n-j,n-i)}\right)P_n \\ &= -\sum_{i < j < k} D_{(i,j,k)}, \end{aligned}$$

that is,

$$D_n^{s_1} + D_n^{s_2} = 0.$$

The proof is complete. \square

By using the same arguments as in Theorem 3.4 based on Lemma 3.5 and Lemma 3.8, we can easily get the following theorem.

Theorem 3.9. *The algorithm SCR² applied to (3.1) has the order of convergence 2, which is in company with the reverse order either (2.5) or (2.6).*

4. Numerical experiments

In this section, we present numerical results which verify the theoretical results in the previous sections. In addition, we provide an example of the non-conserved system which can be converted to the conserved one.

Figure 2-(a) depicts a circular reaction with three substances A, B and C with initial values $A(0) = 1, B(0) = 2$ and $C(0) = 3$. The system of ODEs

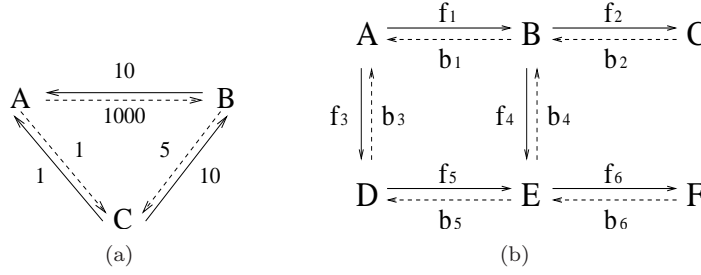


FIGURE 2. (a) A circular reaction (b) A general reaction system

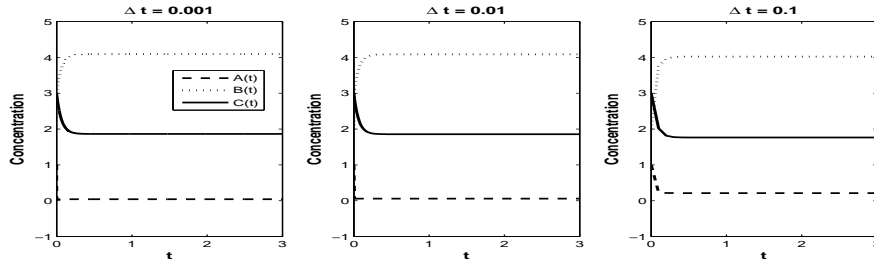


FIGURE 3. Stability of the CR² algorithm

TABLE 1. Convergence behaviors of CR², SCR² and BDF2 by means of $\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ and $\sum_{i=1}^N \|\mathbf{y}(t_i) - \mathbf{y}^i\|_1 \Delta t$ with Δt varying from 10^{-1} to 10^{-6} where $N = t_F/\Delta t$ with $t_F = 3$

Δt	$\ \mathbf{y}(t_F) - \mathbf{y}^N\ _1$			$\sum_{i=1}^N \ \mathbf{y}(t_i) - \mathbf{y}^i\ _1 \Delta t$		
	CR ²	SCR ²	BDF2	CR ²	SCR ²	BDF2
10^{-1}	3.4182e-01	1.6979e-01	3.0723e-12	1.0295	5.2196e-01	6.4610e-02
10^{-2}	3.2857e-02	1.4643e-02	1.5698e-13	9.9615e-02	4.4658e-02	3.5786e-03
10^{-3}	2.1366e-03	3.0403e-04	2.3378e-12	6.5804e-03	9.3141e-04	4.9409e-04
10^{-4}	1.8653e-04	3.0979e-06	2.4701e-11	5.7888e-04	9.5901e-06	8.9323e-06
10^{-5}	1.8376e-05	3.1126e-08	4.2676e-11	5.7083e-05	9.6640e-08	9.6306e-08
10^{-6}	1.8348e-06	3.7633e-10	7.4494e-10	5.7002e-06	1.0620e-09	2.5923e-09

becomes

$$(4.1) \quad \frac{d}{dt} \begin{bmatrix} A(t) \\ B(t) \\ C(t) \end{bmatrix} = \begin{bmatrix} -1001 & 10 & 1 \\ 1000 & -15 & 10 \\ 1 & 5 & -11 \end{bmatrix} \begin{bmatrix} A(t) \\ B(t) \\ C(t) \end{bmatrix} = M \begin{bmatrix} A(t) \\ B(t) \\ C(t) \end{bmatrix}.$$

Many explicit numerical methods such as Euler’s method and the fourth order Runge-Kutta method (RK4 method) in common use have their pros and cons. The structure of explicit methods is so simple that they are popular and may

diminish computing cost at each time step. But, due to its conditional stability, the restriction on the choice of the size of time step is troublesome. For a stiff problem such as (4.1), the drawback of explicit methods is more severe. Note that Euler's method for (4.1) is stable only if

$$\Delta t \leq \frac{S_1}{\lambda_{\min}} \approx 1.9782 \times 10^{-3} \text{ with } S_1 = -2$$

and the RK4 method for (4.1) is stable only if

$$\Delta t \leq \frac{S_2}{\lambda_{\min}} \approx 2.7531 \times 10^{-3} \text{ with } S_2 \approx -2.7853,$$

where λ_{\min} is the minimum nonzero eigenvalue of M and

$$S_i = \min_{z \in \partial D_i \subset \mathbb{C}} \operatorname{Re}(z), \quad i = 1, 2$$

with the stability domain D_i of Euler's method and the RK4 method, respectively. Figure 3 shows that the CR² algorithm gives a stable approximation even if a large Δt is used. For a stiff problem, Gear's method is widely used in order to avoid the problems of instability due to the stiffness. We investigate the difference between two proposed methods and one of Gear's method, a two-step backward differentiation formula (BDF2) which is an *A-stable* method of order 2. BDF2 takes a long time to solve a linear system. Moreover, we should pay special attention during the process of solving the relevant linear system with BDF2 because it may be ill-conditioned for a large Δt . Hence, two explicit algorithms CR² and SCR² are superior to BDF2 in view of efficiency and simplicity of implementation. To observe the convergence speed of three methods, two different errors $\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ and $\sum_{i=1}^N \|\mathbf{y}(t_i) - \mathbf{y}^i\|_1 \Delta t$ are shown in Table 1, where $\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ is computed at the terminal point $t_F = 3$ and $\sum_{i=1}^N \|\mathbf{y}(t_i) - \mathbf{y}^i\|_1 \Delta t$ is calculated over all discrete time steps $t = t_i$ with $N = t_F/\Delta t$. Note that the performance of SCR² is comparable to that of BDF2 over all discrete time steps while BDF2 shows more accurate results at the terminal point t_F .

Next, we consider a somewhat complicated general reaction system in Figure 2-(b) with initial values

$$A(0) = 1, \quad B(0) = 0, \quad C(0) = 0, \quad D(0) = 0, \quad E(0) = 0 \quad \text{and} \quad F(0) = 0,$$

where

$$\begin{aligned} f_1 &= 0.5, & f_2 &= 0.01, & f_3 &= 5.0, & f_4 &= 0.1, & f_5 &= 0.1, & f_6 &= 1.0, \\ b_1 &= 0.05, & b_2 &= 0.001, & b_3 &= 0.5, & b_4 &= 0.01, & b_5 &= 0.01, & b_6 &= 1.0. \end{aligned}$$

In order to illustrate convergence properties of two algorithms CR² and SCR², two kinds of errors measured in the l_1 -norm and estimates of the order of convergence are summarized in Tables 2 and 3. Table 2 gives errors $\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ for varying Δt and estimates of the convergence order corresponding to $\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ where $N = t_F/\Delta t$ with the terminal point $t_F = 10$. The numbers presented in Table 2 imply that when the time step Δt is halved, the errors

TABLE 2. Convergence behaviors of the algorithms CR² and SCR² by means of $\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ with Δt varying from 1/2 to 1/256 where $N = t_F/\Delta t$ with $t_F = 10$

Δt	CR ²		SCR ²	
	$\ \mathbf{y}(t_F) - \mathbf{y}^N\ _1$	order	$\ \mathbf{y}(t_F) - \mathbf{y}^N\ _1$	order
1/2	5.7923e-02		1.0089e-02	
1/4	2.7704e-02	1.0640	3.4552e-03	1.5460
1/8	1.3454e-02	1.0421	1.2101e-03	1.5137
1/16	6.6209e-03	1.0229	3.6181e-04	1.7418
1/32	3.2836e-03	1.0118	9.9202e-05	1.8668
1/64	1.6350e-03	1.0059	2.5992e-05	1.9323
1/128	8.1584e-04	1.0030	6.6537e-06	1.9659
1/256	4.0750e-04	1.0015	1.6833e-06	1.9829

TABLE 3. Convergence behaviors of the algorithms CR² and SCR² by means of $\sum_{i=1}^N \|\mathbf{y}(t_i) - \mathbf{y}^i\|_1 \Delta t$ with Δt varying from 1/2 to 1/256 where $N = t_F/\Delta t$ with $t_F = 10$

Δt	CR ²		SCR ²	
	$\sum_{i=1}^N \ \mathbf{y}(t_i) - \mathbf{y}^i\ _1 \Delta t$	order	$\sum_{i=1}^N \ \mathbf{y}(t_i) - \mathbf{y}^i\ _1 \Delta t$	order
1/2	1.4005		2.6305e-01	
1/4	6.6803e-01	1.0680	7.0304e-02	1.9036
1/8	3.2096e-01	1.0575	1.8225e-02	1.9477
1/16	1.5670e-01	1.0344	4.7130e-03	1.9512
1/32	7.7354e-02	1.0185	1.2067e-03	1.9656
1/64	3.8422e-02	1.0095	3.0594e-04	1.9797
1/128	1.9147e-02	1.0048	7.7070e-05	1.9890
1/256	9.5573e-03	1.0024	1.9344e-05	1.9943

$\|\mathbf{y}(t_F) - \mathbf{y}^N\|_1$ of CR² and SCR² decay linearly and quadratically, respectively. On the other hand, Table 3 displays the errors $\sum_{i=1}^N \|\mathbf{y}(t_i) - \mathbf{y}^i\|_1 \Delta t$ and estimates of the convergence order. According to Table 3, we confirm that the algorithms CR² and SCR² converge of order 1 and 2, respectively, over the whole interval $(0, 10]$.

Finally, we slightly relax the requirement of the conserved system. Let us consider a non-conserved system

$$(4.2) \quad \frac{d}{dt} \begin{bmatrix} A(t) \\ B(t) \end{bmatrix} = \begin{bmatrix} -10 & 0.5 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} A(t) \\ B(t) \end{bmatrix} = M_v \begin{bmatrix} A(t) \\ B(t) \end{bmatrix},$$

TABLE 4. Convergence behaviors of the algorithms CR² and SCR² by means of $\sum_{i=1}^N \|\mathbf{y}_v(t_i) - \mathbf{y}_v^i\|_1 \Delta t$ with Δt varying from 10^{-1} to 10^{-5} where $\mathbf{y}_v(t) = [A(t), B(t)]^T$ and $N = t_F/\Delta t$ with $t_F = 5$

Δt	CR ²		SCR ²	
	$\sum_{i=1}^N \ \mathbf{y}_v(t_i) - \mathbf{y}_v^i\ _1 \Delta t$	order	$\sum_{i=1}^N \ \mathbf{y}_v(t_i) - \mathbf{y}_v^i\ _1 \Delta t$	order
10^{-1}	2.7803e-01		1.3788e-01	
10^{-2}	2.6722e-02	1.0172	1.7209e-03	1.9038
10^{-3}	2.6594e-03	1.0021	1.7650e-05	1.9890
10^{-4}	2.6580e-04	1.0002	1.7695e-07	1.9989
10^{-5}	2.6579e-05	1.0000	1.7660e-09	2.0009

where $A(0) = 1$ and $B(0) = 10$. We introduce an artificial substance C which plays a role as a sink substance in the following conserved system

$$(4.3) \quad \frac{d}{dt} \begin{bmatrix} A(t) \\ B(t) \\ C(t) \end{bmatrix} = \begin{bmatrix} -10 & 0.5 & 0 \\ 1 & -1 & 0 \\ 9 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} A(t) \\ B(t) \\ C(t) \end{bmatrix} = M_c \begin{bmatrix} A(t) \\ B(t) \\ C(t) \end{bmatrix},$$

where $C(0) = 0$. It is easily expected that the modification of the non-conserved system into the conserved system can be done provided M_v is columnwise-diagonally dominant. Table 4 shows the errors for the approximations obtained by applying the algorithms CR² and SCR² to (4.3).

Remark 4.1. Let us consider a non-conserved system where the relevant matrix M_v violates the columnwise-diagonal dominance. Similarly to (4.3), the non-conserved system can be converted into the conserved one by adding an artificial substance which makes each column of the resultant matrix M_g add up to 0. But, M_g differs from M_c in (4.3) in the fact that M_g has negative off-diagonal entries. Unfortunately, our mathematical verification of the proposed methods does not cover such a case. But in practice, we numerically observed that the algorithms CR² and SCR² applied to the above case give the same performance as in a linear reaction system (3.1) in view of both stability and convergence.

5. Conclusions

We proposed two numerical schemes CR² and SCR² applicable to linear reaction systems. Since the CR² algorithm is motivated by the exact solver for a reversible reaction with two substances, it is very easy to implement. On the other hand, the SCR² algorithm, a symmetrized version of the CR² algorithm, is enhanced in view of the convergence speed. Unlike most of explicit methods, CR² and SCR² are absolutely stable in spite of their explicitness. We

analyzed stability and convergence properties of CR^2 and SCR^2 and confirmed the theoretical results by the numerical experiments.

References

- [1] J. C. Butcher and H. Podhaisky, *On error estimation in general linear methods for stiff ODEs*, Appl. Numer. Math. **56** (2006), no. 3-4, 345–357.
- [2] K. Eriksson, C. Johnson, and A. Logg, *Explicit time-stepping for stiff ODEs*, SIAM J. Sci. Comput. **25** (2003/04), no. 4, 1142–1157.
- [3] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations. II*, Springer Series in Computational Mathematics, 14. Springer-Verlag, Berlin, 1991.
- [4] M. Inc, *New L-stable method for numerical solutions of ordinary differential equations*, Appl. Math. Comput. **188** (2007), no. 1, 779–785.
- [5] A. Iserles, *Composite methods for numerical solution of stiff systems of ODEs*, SIAM J. Numer. Anal. **21** (1984), no. 2, 340–351.
- [6] J. Keener and J. Sneyd, *Mathematical Physiology*, Interdisciplinary Applied Mathematics, 8. Springer-Verlag, New York, 1998.
- [7] M. W. J. M. Musters, J. B. Bassingthwaite, V. Panday, N. A. W. van Riel, and G. J. van der Vusse, *Computational modeling of cardiac fatty acid uptake and utilization*, 173–224, G. J. van den Vusse, Amsterdam, Lipobiology, Elsevier, 2004.
- [8] N. C. Price and R. A. Dwek, *Principles and Problems in Physical Chemistry for Biochemists*, Oxford University Press, Oxford, 1979.
- [9] S. Rush and H. Larsen, *A practical algorithm for solving dynamic membrane equations*, IEEE Trans. Biomed. Eng. **25** (1978), 389–392.
- [10] W. Simon, *Mathematical Techniques for Biology and Medicine*, Third edition. Dover Publications, Inc., New York, 1986.
- [11] M. Tokman, *Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods*, J. Comput. Phys. **213** (2006), no. 2, 748–776.
- [12] W. Walter, *Ordinary Differential Equations*, Graduate Texts in Mathematics, 182. Readings in Mathematics. Springer-Verlag, New York, 1998.

CHANG-OCK LEE
 DEPARTMENT OF MATHEMATICAL SCIENCES
 KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
 DAEJEON 305-701, KOREA
E-mail address: colee@kaist.edu

CHAE HUN LEEM
 DEPARTMENT OF PHYSIOLOGY
 COLLEGE OF MEDICINE
 UNIVERSITY OF ULSAN
 SEOUL 138-736, KOREA
E-mail address: leemch@amc.seoul.kr

EUN-HEE PARK
 DEPARTMENT OF MATHEMATICAL SCIENCES
 KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
 DAEJEON 305-701, KOREA
E-mail address: mfield@kaist.ac.kr

JAE BOUM YOUM
DEPARTMENT OF PHYSIOLOGY
COLLEGE OF MEDICINE
INJE UNIVERSITY
PUSAN 614-735, KOREA
E-mail address: youmjb@inje.ac.kr