Research Article

Adaptive Optimal m-Stage Runge-Kutta Methods for Solving Reaction-Diffusion-Chemotaxis Systems

Jui-Ling Yu

Department of Financial and Computational Mathematics, Providence University, Taichung 43301, Taiwan

Correspondence should be addressed to Jui-Ling Yu, jlyu@pu.edu.tw

We present a class of numerical methods for the reaction-diffusion-chemotaxis system which is significant for biological and chemistry pattern formation problems. To solve reaction-diffusion-chemotaxis systems, efficient and reliable numerical algorithms are essential for pattern generations. Along with the implementation of the method of lines, implicit or semi-implicit schemes are typical time stepping solvers to reduce the effect on time step constrains due to the stability condition. However, these two schemes are usually difficult to employ. In this paper, we propose an adaptive optimal time stepping strategy for the explicit m-stage Runge-Kutta method to solve reaction-diffusion-chemotaxis systems. Instead of relying on empirical approaches to control the time step size, variable time step sizes are given explicitly. Yet, theorems about stability and convergence of the algorithm are provided in analyzing robustness and efficiency. Numerical experiment results on a testing problem and a real application problem are shown.

1. Introduction and Background

The reaction-diffusion-chemotaxis system is the most common model used to describe chemical and biological processes and the most successful model used to describe the generation of pattern formations. Numerical simulation of such system is necessary because complicated chemotaxis terms and highly nonlinear reaction terms increase the difficulty on finding analytical solutions for realistic systems. Even though, some of the early chemotaxis models had analytical solution or one-dimensional simulation with reduced models [1, 2], the development of numerical methods for solving chemotaxis systems does not seem sufficient comparing with other type of PDEs systems. Since interesting patterns may emerge and evolve before they reach steady state [3–8], some particular heed needs to be paid to the

numerical methods to ensure that the numerical simulated patterns literally obey primary model equations [5, 8].

Chemotaxis is a directed motion, shown in motile organism, which moves toward or away from certain chemical or environment. It has been observed in microorganism, plants, animals, even in blood vessels of cancer tumors. For bacteria, chemotaxis was first discovered in the lab by Wilhelm Pfeffer and his associates in *E. coli.* in 1884 [9]. Besides the random motion of bacteria, they also show directed random movements in response to some higher concentration of oxygen. Keller and Segel [1] construct the first differential chemotaxis models. They extend the reaction-diffusion model with a convection term to simulate chemotaxis. Based on their work, many models are constructed, especially on the development of bacteria colonies [3–8, 10].

The general form of the reaction-diffusion-chemotaxis is given by [11].

$$u_t = D_u \Delta u + \nabla \left(u \chi(p) \nabla p \right) + f(p, u). \tag{1.1}$$

Here, u represents the microorganism (bacteria) density; p stands for chemoattractant concentration and is given or obtained from another equation. f is the reaction term which models the interactions between the chemicals. The diffusion term, $D_u \Delta u$, imitates the random motion of particles with positive constant diffusivity D_u. $\nabla \cdot (u\chi(p)\nabla p)$ describes the directed random movement of particles p that are modulated by the concentration gradient of the chemoattractant. The chemotaxis parameter function, χ, indicates the strength of the chemotaxis, which is given by the Keller-Segal model.

$$\chi(x) = \frac{C_0}{(C_1 + x)^s}, \tag{1.2}$$

where C_0 and C_1 are positive constants and $s \geq 1$ is an integer power. Observe that $\lim_{x \to \infty} \chi(x) = 0$ which indicates that particles tend to stay where they are when the local concentration of chemoattractant is large.

The basic idea for illustrating patterns using reaction-diffusion-chemotaxis systems is to demonstrate that eigenvalues of local linearization that are responsible for the growth (decay) can have positive (negative) real parts. Therefore, it can lead to growth at some places and decay at others, resulting in spatially inhomogeneous patterns [12, 13]. Based on this idea, we consider the method of lines (MOL) approach in this paper because it is not only the first step for construction of many numerical methods, but also provides good explanations about the generation of patterns through analyzing the positive (growth modes) and negative (decay modes) of eigenvalues. Other numerical methods used on the chemotaxis model include the fractional step method [14], the Alternating Direction Implicit (ADI) method [15, 16], and the optimal two-stage scheme [5, 8] listed here for readers's references.

The MOL approach reduces the partial differential equations to a system of ordinary differential equations (ODEs) by approximating spatial derivatives with numerous spatial discretization methods, for example, the finite difference, finite element, and spectral methods. The solution can be obtained after following some suitable time integrator. According to various spatial discretization features, the computation work can be facilitated by developing some more sophisticated robust and efficient adaptive time integrators [5, 8].

In this paper, we use standard finite differences of center difference for the Laplacian operator and upwind differences for the chemotaxis term. The reason for adapting upwind

differences is the stability concern. When the chemotaxis dominates, the system behaves more like a hyperbolic system. Then the upwind differences are more stable methods [17]. It is worth to mention that the following time-stepping scheme can also be applied to other discretization methods when the domain containing the eigenvalues is known or can be estimated. The general form of the resulting ODE system after spatial discretization reads

$$\frac{d\mathbf{u}}{dt} = M(t)\mathbf{u} + \mathbf{b(t)}, \tag{1.3}$$

where $M(t)$ is a time-dependent matrix and $\mathbf{b}(t)$ is a reaction vector which also depends on time.

Typically, the time integrator for (1.3) can be classified as implicit or explicit methods. Generally, the numerical scheme for solving implicit method is more complicated than explicit method since it often involves solving systems of algebraic equations or needs to deal with the inverse of matrices. Although implicit methods allow larger step size than explicit ones for those spatial discretization matrices only having eigenvalues with negative real parts, the computational cost may still be high due to the complicated implementation for implicit methods. Most importantly, the problem regarding time constrain still exists if the spatial matrices contain very positive real parts of eigenvalues; even for the implicit method the use of very small time steps is yet unpreventable in order to meet the stability condition. This situation is likely to occur for pattern formation problems for which the spatial discretization matrices comprise both positive and negative real parts for their eigenvalues [5, 8]. Besides, for pattern formation problems, those positive real eigenvalues are the unpredictable ones which evolve along with time. Bear these factors in mind, the explicit methods are actually better ones for the reaction-diffusion-chemotaxis systems.

In the present paper, we start with reviewing the explicit m-stage Runge-Kutta methods and then construct their adaptive optimal time steps to enhance the computation efficiency for the chemotaxis system. The optimal time step formulas are presentable and easy to calculate. Only some simple estimations about the bounds for the largest and smallest eigenvalues of the discretization spatial matrices are required at each time step. The numerical simulations on the test and realistic problems are provided. Comparison will be made in the testing problem to demonstrate that our method is not only stable but also more efficient than other similar methods, referring to CPU time and the number of iterations.

2. Optimal Time Steps for Explicit Runge-Kutta Methods

2.1. Time Independence ODEs Systems

Since interesting patterns are caused by the interaction between growth and decay modes, to analyze system (1.3) and relate it to the pattern formation model (1.1), one can do the local linearization property of the right-hand side of (1.3). The positive and negative real eigenvalues of the linearization matrix then, respectively, correspond to the growth and decay modes. Because of this reason, we will consider the corresponding ODEs induced from the chemotaxis system and introduce our idea by starting with the constant coefficients case, that is,

$$\frac{d\mathbf{u}}{dt} = M\mathbf{u} + \mathbf{b} \tag{2.1}$$

then generalize it to the time-dependent coefficients case. The general formulation of the m-stage Runge-Kutta method is

$$k_1 = \mathbf{f}(t_n, \mathbf{u}_n),$$

$$k_q = \mathbf{f}\left(t_n + \alpha_q \Delta t, \mathbf{u}_n + \Delta t_n \sum_{s=1}^{q-1} \beta_{qs} k_s\right), \quad q = 2, \ldots, m,$$

$$\alpha_q = \sum_{s=1}^{q-1} \beta_{qs}, \tag{2.2}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t_n \sum_{s=1}^{q} \gamma_s k_s, \quad q = 2, \ldots, m,$$

where m is the stage number and $\alpha_q = \sum_{j=1}^{q-1} \beta_{qj}$, $q = 2, \ldots, m$. The subscripts n and $n + 1$ are the iteration level. For simplicity, we first look at the uniform time step size $\Delta t = t_n - t_{n-1}$ for all n. Then $\mathbf{u}_n \approx \mathbf{u}(\mathbf{x}_n)$ is the numerical solution at the nth step. Equations for the unknowns γ_s and β_{qs} can be obtained by matching terms with Taylor's expansion.

We choose single-step explicit methods, which basically are Runge-Kutta methods, because they are easy to implement and especially efficient for ODEs systems derived from the PDEs. The method is set to be explicit since the range of eigenvalues with negative real parts is measurable for chemotaxis systems. In [5, 8], we have developed adaptive two-stage time stepping scheme with optimum time step sizes for solving chemotaxis system and verified that it is reliable and more economical when comparing with other similar methods. We conjecture that the similar strategy used in [5, 8] could be used to derive optimal adaptive three- and four-stage methods. In the two-stage case, the size of the optimal step is determined by the root of the first-order equation hence it is absolute optimal within one time step. If our conjecture is correct, the sizes of optimal steps for higher-stage methods should be the roots of some higher-order equations. But, are these adaptive step sizes absolute optimal? And are they easy to be found as in the two-stage case?

To answer these questions, we begin our study with the three-stage Runge-Kutta method and set $\mathbf{b} = 0$ in (2.1). As in [5, 8], we first fix all the intermediate coefficients at $t = t_n$ for the three-stage Runge-Kutta method. The general time dependence case is proposed in the next section. Denoting $M = M(t_n)$ and applying (2.2) to the three-stage Runge-Kutta method with all the intermediate values being fixed at t_n, we arrive at

$$k_1 = M\mathbf{u}_n,$$

$$k_2 = M(1 + \alpha_2 \Delta t M)\mathbf{u}_n, \tag{2.3}$$

$$k_3 = M\left(1 + \alpha_3 \Delta t M + \alpha_2 \beta_{32} \Delta t^2 M^2\right)\mathbf{u}_n,$$

$$\mathbf{u}_{n+1} = \left(1 + \delta_1 \Delta t M + \delta_2 \Delta t^2 M^2 + \delta_3 \Delta t^3 M^3\right)\mathbf{u}_n, \tag{2.4}$$

where $\delta_i, i = 1, 2, 3$, are combinations of the coefficients β_{qs} and γ_s. Since $\mathbf{u}(t_{n+1}) = e^{M\Delta t}\mathbf{u}(t_n)$ and assuming $\mathbf{u}_n$ is exact up to the order Δt^3, we have $\delta_1 = 1$, $\delta_2 = (1/2!)$, $\delta_3 = (1/3!)$. Similar

arguments also hold for RK methods with other stages. Thus, if we treat it as a standard iterative method, (2.4) is a one-step method [18]:

$$\mathbf{u}_{n+1} = \left[1 + \Delta t M + \frac{1}{2!}(\Delta t M)^2 + \cdots + \frac{1}{m!}(\Delta t M)^m \right] \mathbf{u}_n.$$

$$= p_m(\Delta t)\mathbf{u}_n.$$

(2.5)

In the case of $\mathbf{b} \neq 0$, (2.5) becomes

$$\mathbf{u}_{n+1} = p_m(\Delta t)\mathbf{u}_n + \left(1 + \frac{\Delta t}{2!} M + \cdots + \frac{\Delta t^m}{m!} M^{m-1} \right) \mathbf{b}.$$

(2.6)

Denote $\mathbf{v}$ as the solution by solving the system with the perturbated initial condition; we have

$$\mathbf{v}_{n+1} = p_m(\Delta t)\mathbf{v}_n + \left(1 + \frac{\Delta t}{2!} M + \cdots + \frac{\Delta t^m}{m!} M^{m-1} \right) \mathbf{b}.$$

(2.7)

From (2.6) and (2.7), the accumulation error can be written as

$$|\mathbf{u}_{n+1} - \mathbf{v}_{n+1}| = \left| p_m(\Delta t) \right| \times |\mathbf{u}_n - \mathbf{v}_n|.$$

(2.8)

Note that

$$p_m(\lambda) = 1 + \Delta t \lambda + \frac{1}{2!}(\Delta t \lambda)^2 + \cdots + \frac{1}{m!}(\Delta t \lambda)^m$$

(2.9)

is called the amplification factor, where λ is an eigenvalue of M. It is obtained when a scalar test problem, $u' = \lambda u$, is applied to the single-step method.

Suppose that $\lambda = a + bi = \operatorname{Re}\lambda + \operatorname{Im}\lambda i$. For pattern formation problems, what we are interested in is the long-term behavior of the solution vectors. It is important to ensure that observed patterns are accurate in the sense that they are not coming from the accumulation error of numerical schemes. We therefore make the following definition of stability.

Definition 2.1. An explicit single-step method is said to be stable if $|p_m(\lambda)| < 1$ when $a < 0$ and $|p_m(\lambda)| \le 1 + a\Delta t + O(\Delta t)$ when $a \ge 0$.

Here $O(\Delta t)$ means higher order of Δt. Our definition is designed to guarantee appropriate numerical solutions for both decay and growth cases. Based on this definition, we get restrictions on the step size. Typically, time steps are restricted either when the scheme is explicit and $a < 0$ or when the scheme is implicit and $a > 0$ [5, 8]. Since the amplification factors are functions of complex numbers, it is sometimes difficult to analyze. The following theorem shows that the amplification factor can be approximated by simpler functions of a plus some error terms instead of functions of complex eigenvalues λ. Observe that the order of error terms indicated in Theorem 2.2 is the same as the truncation error of the m-stage Runge-Kutta method. Therefore, we can work with this simpler function without losing the order of accuracy.

2.1.1. Stability Analysis and the Optimization Method

The following theorem states that the m-stage Runge-Kutta method satisfies the stability condition automatically for $a > 0$ but needs constraints on Δt when $a < 0$.

Theorem 2.2. $|p_m(\lambda)|^2 = |p_m(a+bi)|^2 = (1 + \Delta ta + (1/2)\Delta t^2 a^2 + \cdots + (1/m!)\Delta t^m a^m)^2 + O(\Delta t^{m+1})$, *where $m = 1, \ldots, 4$.*

Proof. The proof for $m = 1$ is straightforward, and the proof for $m = 2$ is in [5, 8]. For $m = 3$,

$$
\begin{aligned}
|p_3(\lambda)|^2 &= \left(1 + \Delta ta + \frac{\Delta t^2}{2}\left(a^2 - b^2\right) + \frac{\Delta t^3}{6}\left(a^3 - 3ab^2\right) \right)^2 \\
&\quad + \left(\Delta tb + \Delta t^2 ab + \frac{\Delta t^3}{6}\left(3a^2 b - b^3\right) \right)^2 \\
&= \left(1 + \Delta ta + \frac{1}{2}(\Delta ta)^2 + \frac{1}{6}(\Delta ta)^3 \right)^2 - 2(1 + \Delta ta)\left(\frac{1}{2}\Delta t^2 b^2 + \frac{1}{2}\Delta t^3 ab^2 \right) \\
&\quad + O\left(\Delta t^4\right) + \left(\Delta tb + \Delta t^2 ab + \frac{\Delta t^3}{6}\left(3a^2 b - b^3\right) \right)^2 \\
&= \left(1 + \Delta ta + \frac{1}{2}(\Delta ta)^2 + \frac{1}{6}(\Delta ta)^3 \right)^2 \\
&\quad - \Delta t^2 b^2 - 2\Delta t^3 ab^2 + O\left(\Delta t^4\right) + \Delta t^2 b^2 + 2\Delta t^3 ab^2 + O\left(\Delta t^4\right) \\
&= \left(1 + \Delta ta + \frac{1}{2}(\Delta ta)^2 + \frac{1}{6}(\Delta ta)^3 \right)^2 + O\left(\Delta t^4\right).
\end{aligned}
\tag{2.10}
$$

For $m = 4$,

$$
\begin{aligned}
|p_4(\lambda)|^2 &= \left(1 + \Delta ta + \frac{\Delta t^2}{2}\left(a^2 - b^2\right) + \frac{\Delta t^3}{6}\left(a^3 - 3ab^2\right) + \frac{\Delta t^4}{24}\left(a^4 - 6a^2 b^2 + b^4\right) \right)^2 \\
&\quad + \left(\Delta tb + \Delta t^2 ab + \frac{\Delta t^3}{6}\left(3a^2 b - b^3\right) + \frac{\Delta t^4}{24}\left(4a^3 b - 4ab^3\right) \right)^2 \\
&= \left(1 + \Delta ta + \frac{1}{2}(\Delta ta)^2 + \frac{1}{6}(\Delta ta)^3 + \frac{1}{24}(\Delta ta)^4 \right)^2 \\
&\quad - 2\left(1 + \Delta ta + \frac{1}{2}(\Delta ta)^2 + \frac{1}{6}(\Delta ta)^3 + \frac{1}{24}(\Delta ta)^4 \right) \\
&\quad * \left(\frac{1}{2}\Delta t^2 b^2 + \frac{1}{2}\Delta t^3 ab^2 + \frac{1}{4}\Delta t^4 a^2 b^2 - \frac{1}{24}\Delta t^4 b^4 \right) \\
&\quad + \left(\frac{1}{2}\Delta t^2 b^2 + \frac{1}{2}\Delta t^3 ab^2 + \frac{1}{4}\Delta t^4 a^2 b^2 - \frac{1}{24}\Delta t^4 b^4 \right)^2
\end{aligned}
$$

$$
+ \left(\Delta t b + \Delta t^2 ab + \frac{\Delta t^3}{6} \left(3a^2 b - b^3 \right) + \frac{\Delta t^4}{24} \left(4a^3 b - 4ab^3 \right) \right)^2
$$

$$
= \left(1 + \Delta t a + \frac{1}{2} (\Delta t a)^2 + \frac{1}{6} (\Delta t a)^3 + \frac{1}{24} (\Delta t a)^4 \right)^2
$$

$$
+ \Delta t^4 \left(-2a^2 b^2 + \frac{1}{12} b^4 \right) - 2\Delta t^3 ab^2 - \Delta t^2 b^2 + O\left(\Delta t^5 \right) + \frac{1}{4} \Delta t^4 b^4 + O\left(\Delta t^5 \right)
$$

$$
+ \Delta t^4 \left(-\frac{1}{3} b^4 + 2a^2 b^2 \right) + 2\Delta t^3 ab^2 + \Delta t^2 b^2 + O\left(\Delta t^5 \right)
$$

$$
= \left(1 + \Delta t a + \frac{1}{2} (\Delta t a)^2 + \frac{1}{6} (\Delta t a)^3 + \frac{1}{24} (\Delta t a)^4 \right)^2 + O\left(\Delta t^5 \right).
$$

(2.11)

The geometric interpretation is that, for spatial matrices $M(t_n)$, the eigenvalues with negative real parts are assumed to be contained in the left-plane ellipse. Let us denote $1 + \Delta t a + (1/2)\Delta t^2 a^2 + \cdots + (1/m!)\Delta t^m a^m$ as $p_m(a, \Delta t)$. To examine whether numerically generated patterns truly obey the original models, one needs to verify if the negative modes $(a < 0)$ are diminished and positive modes $(a > 0)$ are growing accordingly. That is, when the negative modes occur, $p_m(a, \Delta t)$ is less than one, and is greater than one when the spatial matrix $M(t_n)$ happens to have positive modes. From $p_m(a, \Delta t)$, we find that when $a > 0$, $p_m(a, \Delta t)$ is definitely greater than unity, hence the stability condition is automatically satisfied; when $a < 0$, $p_m(a, \Delta t)$ is then possible to be less than unity. It is necessary to put the limitation on the time steps to ensure the stability condition $p_m(a, \Delta t) < 1$ is preserved. Besides, the negative modes should be diminished in an effective way so that the computational cost can be reduced. Under these consideration, we derive the optimal time step which not only diminishes decay modes efficiently but also guarantees the stability conditions. Before moving on the theorems about showing how we get the optimal time steps for m-stage Runge-Kutta method, let us recall the Gerschgorin theorem [19]. It says that for matrices with real coefficients, the distribution for the real parts of eigenvalues can be estimated simply by the discs centered at the diagonal elements. So we can assume that the range of negative real parts of eigenvalues of the discretized spatial matrices is known. Here, only negative real eigenvalues are needed as they are the only ones responsible for the stability and restriction of the step sizes for explicit schemes. We are now ready to give relevant theorems for constructing optimal step sizes by effectively damping the negative modes. □

Theorem 2.3. *For $a < 0$ and m being even, $p_m(a, \Delta t) > 0$ and is concave up for all $\Delta t > 0$.*

Proof. For $m = 2$, it is a second-degree polynomial with a positive leading coefficient. The proof is straightforward. Note that $p_2(a, \Delta t) > 0$.

Assume $p_{m-2}(a, \Delta t) > 0$ for all even m. For m being even and $m > 2$, we have

$$
\frac{\partial^2 p_m(a, \Delta t)}{\partial \Delta t^2} = a^2 p_{m-2}(a, \Delta t) > 0.
$$

(2.12)

$\circ$: $p_3(-12, \Delta t)$
$*$: $p_3(-10, \Delta t)$
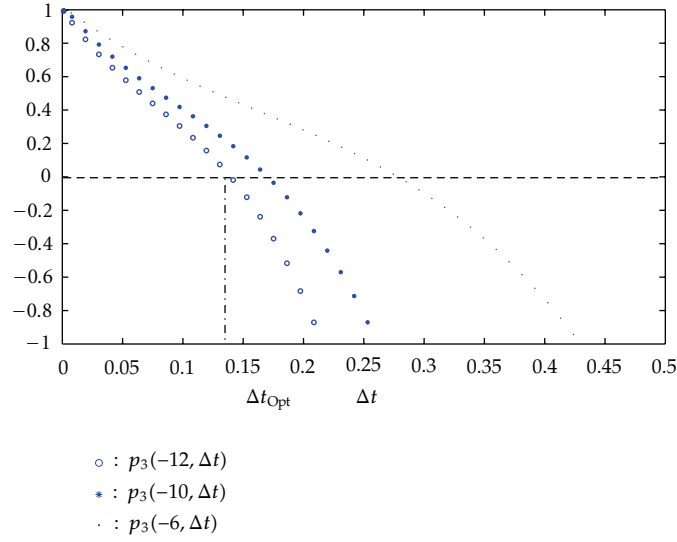$\cdot$: $p_3(-6, \Delta t)$

Figure 1: Graphs for $p_3(a, \Delta t)$ where $a = -6$: $\cdot$, $a = -10$: $*$, $a = -12$: $\circ$.

It follows from mathematical induction, $\partial^2/\partial \Delta t^2 p_m(a, \Delta t) > 0$ for all even m. Hence, $p_m(a, \Delta t)$ is concave up w.r.t. Δt if m is even.

Moreover, $p_m(a, \Delta t) > 0$ for all even m, because $p_m(a, \Delta t) > \exp(a\Delta t) > 0$ for even m.

$\square$

Corollary 2.4. *For $a < 0$ and m being an odd number, $p_m(a, \Delta t) < 1$ and $p_m(a, \Delta t)$ is strictly decreasing for all $\Delta t > 0$.*

Proof. For $m = 1$, the proof is trivial. For $m > 1$, $p_m(a, 0) = 1$ and

$$\frac{\partial p_m(a, \Delta t)}{\partial \Delta t} = a p_{m-1}(a, \Delta t) < 0 \tag{2.13}$$

since $a < 0$ and $m - 1$ is an even number.

Therefore, we conclude $p_m(a, \Delta t) < 1$ for all odd m and $\Delta t > 0$. Moreover, $p_m(a, \Delta t)$ is strictly decreasing with respect to Δt. $\square$

Theorem 2.5 (Case for m being odd). *Assume that $-a_{\max} \le a \le -a_{\min}$, where $a_{\max} > a_{\min} > 0$ are two positive numbers. Then the solution of the following min-max problem occurs at $\Delta t = \Delta t_{\text{opt}}$ in which Δt satisfies $f_m(-a_{\max}, \Delta t) = 0$, where m is an odd number. The min-max problem is stated as*

$$\min_{\Delta t > 0} \max_{-a_{\max} \le a \le -a_{\min}} p_m(a, \Delta t). \tag{2.14}$$

Proof. Note that $p_m(a_1, \Delta t) > p_m(a_2, \Delta t)$ if $a_1 > a_2$ and $a_1, a_2 < 0$. For any fixed Δt, the maximum of $p_m(a, \Delta t)$ is obtained at $p_m(-a_{\min}, \Delta t)$. So the min-max occurs when $p_m(-a_{\max}, \Delta t) = 0$. Since $p_m(a, \Delta t)$ is continuous, $p_m(a, 0) = 1 > 0$, and $p_m(a, \Delta t) < 0$ as
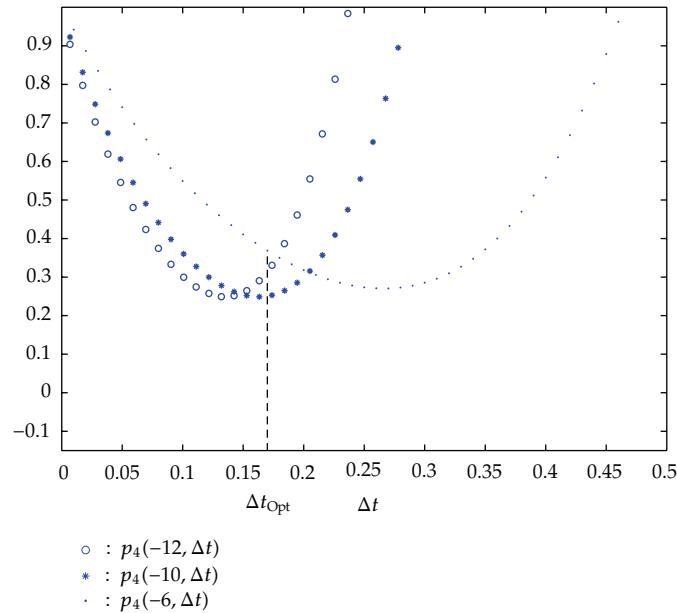
$\circ$: $p_4(-12, \Delta t)$
$*$: $p_4(-10, \Delta t)$
$\cdot$: $p_4(-6, \Delta t)$

Figure 2: Graphs for $p_4(a, \Delta t)$ where $a = -6$: $\cdot$, $a = -10$: $*$, $a = -12$: $\circ$.

$\Delta t \rightarrow \infty$, the solution exists by Intermediate Value Theorem. Furthermore, from Corollary 2.4, $p_m(a, \Delta t)$ is monotone decreasing w.r.t. Δt, so the solution is unique. □

Theorem 2.6 (Case for m being even). *Assume that $-a_{\max} \leq a \leq -a_{\min}$, where $a_{\max} > a_{\min} > 0$ are two positive numbers. Then the solution of the following min-max problem occurs at $\Delta t = \Delta t_{\mathrm{opt}}$ in which Δt satisfies $p_m(-a_{\max}, \Delta t) = p_m(-a_{\min}, \Delta t)$, where m is an even number ($m = 2, 4$). The min-max problem is stated as*

$$\min_{\Delta t > 0} \max_{-a_{\max} \leq a \leq -a_{\min}} p_m(a, \Delta t). \tag{2.15}$$

Proof. For each fixed a, $p_m(a, \Delta t)$ is a quadratic-like polynomial function in Δt. Then for any fixed Δt, the maximum of $p_m(a, \Delta t)$ is obtained at $p_m(-a_{\min}, \Delta t)$, $\Delta t < \Delta t_{\mathrm{opt}}$, and at $p_m(-a_{\max}, \Delta t)$, $\Delta t > \Delta t_{\mathrm{opt}}$. So the minimax occurs when $p_m(-a_{\max}, \Delta t) = p_m(-a_{\min}, \Delta t)$, that is,

$$\sum_{i=0}^{m} \frac{(-a_{\max} \Delta t)^m}{m!} = \sum_{i=0}^{m} \frac{(-a_{\min} \Delta t)^m}{m!}. \tag{2.16}$$

This leads to the solution. □

Let us denote $g_m(\Delta t) = p_m(-a_{\max}, \Delta t) - p_m(-a_{\min}, \Delta t)$. The following theorem states that the optimal time step is unique for m being even. For m being odd, it is shown in Theorem 2.5.

Theorem 2.7. *If m is even, $g_m(\Delta t)$ has only one positive real solution.*

Proof.

$$g_m(\Delta t) = \Delta t (a_{\max} - a_{\min}) \left[\sum_{i=1}^{m} (-\Delta t)^{m-1} \frac{(a_{\max}^m - a_{\min}^m)}{m!(a_{\max} - a_{\min})} \right] \tag{2.17}$$

$$= \Delta t (a_{\max} - a_{\min}) q_m(\Delta t).$$

It follows from (2.17) and the fact that $a_{\max} \neq a_{\min}$, to find the zeros of $g_m(\Delta t)$ is enough to solve $q_m(\Delta t) = 0$. Since $q_m(0) = 1$ and $q_m(\Delta t) < 0$ as Δt tends to infinity, positive real solutions hence exist for $g_m(\Delta t)$ by the Intermediate Value Theorem. Moreover, $q_m(\Delta t)$ is strictly decreasing (Figures 1 and 2). Therefore, the solution is unique. $\qquad \square$

2.2. Time-Dependent ODEs Systems

In the previous section, we derived simpler forms of amplification factors as well as theorems for setting optimal time steps for constant ODEs systems. The optimal time step is obtained under the assumption that the intermediate values of Runge-Kutta methods are fixed at current time. Under the stability consideration, we continue our study on searching optimal time steps and relevant schemes when the ODEs system and Runge-Kutta methods are time dependent. The general form of the time dependent system is

$$\frac{d\mathbf{u}}{dt} = M(t)\mathbf{u} + \mathbf{b(t)}. \tag{2.18}$$

Derivation of Adaptive Optimal Time Step for the Three-Stage Runge-Kutta Method

We start with the three-stage Runge-kutta method. As before, we first set the reaction vector to be zero.

$$
\begin{aligned}
k_1 &= M_n \mathbf{u}_n, \\
k_2 &= M_{n+\alpha_2} (u_n + \beta_{21} \Delta t k_1), \\
k_3 &= M_{n+\alpha_3} (u_n + \Delta t (\beta_{31} k_1 + \beta_{32} k_2)), \\
\mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t (\gamma_1 k_1 + \gamma_2 k_2 + \gamma_3 k_3).
\end{aligned}
\tag{2.19}
$$

Here, $M_n = M(t_n)$. Expand $M_{n+\alpha_i}$ at t_n by Taylor expansion, then $M_{n+\alpha_i} = M_n + \alpha_i \Delta t_n M_n' + ((\alpha_i \Delta t_n)^2/2!) M_n'' + O(\Delta t^3)$, where $i = 2, 3$.

Therefore,

$$k_1 = M_n \mathbf{u}_n,$$

$$k_2 = \left[M_n + \left(\alpha_2 M_n' + \beta_{21} M_n^2 \right) \Delta t_n + \left(\frac{\alpha_2^2}{2} M_n'' + \alpha_2 \beta_{21} M_n M_n' \right) \Delta t_n^2 \right] \mathbf{u}_n + O\left(\Delta t_n^3 \right),$$

$$k_3 = \left[M_n + \alpha_3 M'_n \Delta t_n + (\beta_{31} + \beta_{32}) M_n^2 \Delta t_n + \frac{\alpha_3^2}{2} M''_n \Delta t_n^2 \right.$$

$$\left. + (\beta_{31}\alpha_3 + \beta_{32}(\alpha_2 + \alpha_3)) M'_n M_n \Delta t_n^2 + \beta_{32}\beta_{21} M_n^3 \Delta t_n^2 \right] \mathbf{u}_n + O\left(\Delta t_n^3\right).$$

(2.20)

Since by consistency,

$$\sum_{i=1}^{3} \gamma_i = 1,$$

$$\alpha_q = \sum_{j=1}^{q-1} \beta_{qj}, \quad q = 2, 3,$$

$$\gamma_2 \alpha_2 + \gamma_3 \alpha_3 = \frac{1}{2},$$

$$\gamma_2 \alpha_2^2 + \gamma_3 \alpha_3^2 = \frac{1}{3},$$

$$\gamma_3 \alpha_2 \beta_{32} = \frac{1}{6}.$$

(2.21)

Followed by (2.20) and the consistency condition, (2.19) can be simplified as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \left[M_n \Delta t_n + \frac{1}{2} M'_n \Delta t_n^2 + \frac{1}{6} M''_n \Delta t_n^3 \right.$$

$$\left. + \frac{1}{2} M_n^2 \Delta t_n^2 + \frac{1}{2} M'_n M_n \Delta t_n^3 + \frac{1}{6} M_n^3 \Delta t_n^3 \right] \mathbf{u}_n + O\left(\Delta t_n^4\right).$$

(2.22)

Now, we assume

$$\sum_{i=1}^{3} \tau_i = 1,$$

$$\tau_2 \mu + \tau_3 = \frac{1}{2},$$

$$\tau_2 \mu^2 + \tau_3 = \frac{1}{3}.$$

(2.23)

Then, we can rewrite (2.22) as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \left[(\tau_1 + \tau_2 + \tau_3) M_n \Delta t_n + (\tau_2 \mu + \tau_3) M'_n \Delta t_n^2 + \frac{1}{2} \left(\tau_2 \mu^2 + \tau_3 \right) M''_n \Delta t_n^3 \right.$$

$$+ \frac{1}{2} (\tau_1 + \tau_2 + \tau_3)^2 M_n^2 \Delta t_n^2 + (\tau_1 + \tau_2 + \tau_3)(\tau_2 \mu + \tau_3) M'_n M_n \Delta t_n^3$$

$$\left. + \frac{1}{6} (\tau_1 + \tau_2 + \tau_3)^3 M_n^3 \Delta t_n^3 \right] \mathbf{u}_n + O\left(\Delta t_n^4\right).$$

(2.24)

This way, we arrive the following desirable form through rearrangement:

$$
\begin{aligned}
\mathbf{u}_{n+1} = \mathbf{u}_n + \Bigg[& \tau_1 M_n + \tau_2 \bigg(M_n + \mu \Delta t_n M'_n + \frac{(\mu \Delta t_n)^2}{2} M''_n \bigg) \\
& + \tau_3 \bigg(M_n + \Delta t_n M'_n + \frac{\Delta t_n^2}{2} M''_n \bigg) \\
& + \frac{1}{2} \Big[(\tau_1 + \tau_2 + \tau_3) M_n + (\tau_2 + \tau_3 \mu) M'_n \Delta t_n + O\big(\Delta t_n^2\big) \Big]^2 \Delta t_n^2 \\
& + \frac{1}{6} (\tau_1 + \tau_2 + \tau_3)^3 M_n^3 \Delta t_n^3 \Bigg] \mathbf{u}_n + O\big(\Delta t_n^4\big).
\end{aligned}
\tag{2.25}
$$

Let

$$
J_n = \tau_1 M_n + \tau_2 M_{n+\mu} + \tau_3 M_{n+1},
\tag{2.26}
$$

where

$$
\begin{aligned}
\tau_2 &= -\frac{1}{6\mu(\mu - 1)}, \\
\tau_3 &= \frac{3\mu - 2}{6(\mu - 1)}.
\end{aligned}
\tag{2.27}
$$

Then the previous equation implies

$$
\mathbf{u}_{n+1} = \left[1 + J_n \Delta t_n + \frac{1}{2} (J_n \Delta t_n)^2 + \frac{1}{3!} (J_n \Delta t_n)^3 \right] \mathbf{u}_n + O\big(\Delta t_n^4\big).
\tag{2.28}
$$

The above result illustrates that the time-dependent three-stage Runge-Kutta method can be treated as a single-step method without losing the order of accuracy. In addition, it shares the same form with (2.5) if J_n is replaced by M. Next, we derive a similar argument for the four-stage Runge-Kutta method.

Derivation of Adaptive Optimal Time Step for the Four-Stage Runge-Kutta Method

Applying the four-stage Runge-Kutta method to (2.18), we get

$$
\begin{aligned}
k_1 &= M_n \mathbf{u}_n, \\
k_2 &= M_{n+\alpha_2} (\mathbf{u}_n + \beta_{21} \Delta t k_1), \\
k_3 &= M_{n+\alpha_3} (\mathbf{u}_n + \Delta t (\beta_{31} k_1 + \beta_{32} k_2)), \\
k_4 &= M_{n+\alpha_4} (\mathbf{u}_n + \Delta t (\beta_{41} k_1 + \beta_{42} k_2 + \beta_{43} k_3)), \\
\mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t (r_1 k_1 + r_2 k_2 + r_3 k_3 + r_4 k_4).
\end{aligned}
\tag{2.29}
$$

Since $M_{n+\alpha_i} = M_n + \alpha_i \Delta t_n M'_n + ((\alpha_i \Delta t_n)^2/2!)M''_n + ((\alpha_i \Delta t_n)^3/3!)M'''_n + O(\Delta t^4)$

$$
k_2 = \left[M_n + \left(\alpha_2 M'_n + \beta_{21} M_n^2 \right) \Delta t_n + \left(\frac{\alpha_2^2}{2} M''_n + \alpha_2 \beta_{21} M_n M'_n \right) \Delta t_n^2 \right.
$$

$$
\left. + \left(\frac{\alpha_2^3}{3!} M'''_n + \frac{\alpha_2^2 \beta_{21}}{2} M''_n M_n \right) \Delta t_n^3 \right] \mathbf{u}_n + O\left(\Delta t_n^4 \right),
$$

$$
k_3 = \left[M_n + \alpha_3 M'_n \Delta t_n + (\beta_{31} + \beta_{32}) M_n^2 \Delta t_n + \frac{\alpha_3^2}{2} M''_n \Delta t_n^2 \right.
$$

$$
+ (\beta_{31}\alpha_3 + \beta_{32}(\alpha_2 + \alpha_3)) M'_n M_n \Delta t_n^2 + \beta_{32}\beta_{21} M_n^3 \Delta t_n^2 + \frac{\alpha_3^3}{3!} M'''_n \Delta t_n^3
$$

$$
+ \frac{1}{2}\left(\beta_{31}\alpha_3^2 + \beta_{32}\left(\alpha_2^2 + \alpha_3^2 \right) \right) M''_n M_n \Delta t_n^3 + \beta_{32}\alpha_2\alpha_3 (M'_n)^2 \Delta t_n^3
$$

$$
+ \beta_{32}\beta_{21}(\alpha_2 + \alpha_3) M'_n M_n^2 \Delta t_n^3 \right] \mathbf{u}_n + O\left(\Delta t_n^4 \right), \tag{2.30}
$$

$$
k_4 = \left[M_n + \alpha_4 M'_n \Delta t_n + (\beta_{41} + \beta_{42} + \beta_{43}) M_n^2 \Delta t_n + \frac{1}{2}\alpha_4^2 M''_n \Delta t_n^2 \right.
$$

$$
+ (\beta_{41}\alpha_4 + \beta_{42}(\alpha_2 + \alpha_4) + \beta_{43}(\alpha_3 + \alpha_4)) M'_n M_n \Delta t_n^2
$$

$$
+ (\beta_{42}\beta_{21} + \beta_{43}(\beta_{31} + \beta_{32})) M_n^3 \Delta t_n^2 + \frac{1}{3!}\alpha_4^3 M'''_n \Delta t_n^3
$$

$$
+ \frac{1}{2}\left(\beta_{41}\alpha_4^2 + \beta_{42}\left(\alpha_2^2 + \alpha_4^2 \right) + \beta_{43}\left(\alpha_3^2 + \alpha_4^2 \right) \right) M''_n M_n \Delta t_n^3
$$

$$
+ (\beta_{42}\beta_{21}(\alpha_2 + \alpha_4) + \beta_{43}\beta_{31}(\alpha_3 + \alpha_4) + \beta_{43}\beta_{32}(\alpha_2 + \alpha_3 + \alpha_4)) M'_n M_n^2 \Delta t_n^3
$$

$$
+ (\beta_{42}\alpha_2\alpha_4 + \beta_{43}\alpha_3\alpha_4)(M'_n)^2 \Delta t_n^3 + \beta_{43}\beta_{32}\beta_{21} M_n^4 \Delta t_n^3 \right] \mathbf{u}_n + O\left(\Delta t_n^4 \right).
$$

By consistency of the method,

$$
\sum_{i=1}^{4} \gamma_i = 1,
$$

$$
\alpha_q = \sum_{j=1}^{q-1} \beta_{qj}, \quad q = 2, \ldots, 4,
$$

$$
\gamma_2 \alpha_2 + \gamma_3 \alpha_3 + \gamma_4 \alpha_4 = \frac{1}{2},
$$

$$
\gamma_2 \alpha_2^2 + \gamma_3 \alpha_3^2 + \gamma_4 \alpha_4^2 = \frac{1}{3},
$$

$$\gamma_2 \alpha_2^3 + \gamma_3 \alpha_3^3 + \gamma_4 \alpha_4^3 = \frac{1}{4},$$

$$\gamma_3 \alpha_2 \beta_{32} + \gamma_4 (\alpha_2 \beta_{42} + \beta_{43} \alpha_3) = \frac{1}{6},$$

$$\gamma_3 \beta_{32} \alpha_2 \alpha_3 + \gamma_4 \beta_{42} \alpha_2 \alpha_4 + \gamma_4 \beta_{43} \alpha_3 \alpha_4 = \frac{1}{8},$$

$$\gamma_3 \alpha_2^2 \beta_{32} + \gamma_4 \alpha_2^2 \beta_{42} + \gamma_4 \alpha_3^2 \beta_{43} = \frac{1}{12},$$

$$\gamma_4 \alpha_2 \beta_{43} \beta_{32} = \frac{1}{24}.$$

$$(2.31)$$

Thus,

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \left[M_n \Delta t_n + \frac{1}{2} M_n' \Delta t_n^2 + \frac{1}{6} M_n'' \Delta t_n^3 + \frac{1}{24} M_n''' \Delta t_n^4 \right.$$

$$+ \frac{1}{2} M_n^2 \Delta t_n^2 + \frac{1}{2} M_n' M_n \Delta t_n^3 + \frac{1}{6} M_n'' M_n \Delta t_n^4 + \frac{1}{8} (M_n')^2 \Delta t_n^4 \qquad (2.32)$$

$$\left. + \frac{1}{6} M_n^3 \Delta t_n^3 + \frac{1}{4} M_n' M_n^2 \Delta t_n^4 + \frac{1}{24} M_n^4 \Delta t_n^4 \right] \mathbf{u}_n + O\left(\Delta t_n^5 \right).$$

We further assume

$$\sum_{i=1}^{4} \tau_i = 1,$$

$$\tau_2 \mu_1 + \tau_3 \mu_2 + \tau_4 = \frac{1}{2},$$

$$\tau_2 \mu_1^2 + \tau_3 \mu_2^2 + \tau_4 = \frac{1}{3}, \qquad (2.33)$$

$$\tau_2 \mu_1^3 + \tau_3 \mu_2^3 + \tau_4 = \frac{1}{4}.$$

The previous equation hence can be written as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \left[(\tau_1 + \tau_2 + \tau_3 + \tau_4) M_n \Delta t_n + (\tau_2 \mu_1 + \tau_3 \mu_2 + \tau_4) M_n' \Delta t_n^2 \right.$$

$$+ \frac{1}{2} \left(\tau_2 \mu_1^2 + \tau_3 \mu_2^2 + \tau_4 \right) M_n'' \Delta t_n^3 + \frac{1}{6} \left(\tau_2 \mu_1^3 + \tau_3 \mu_2^3 + \tau_4 \right) M_n''' \Delta t_n^4$$

$$+ \frac{1}{2} (\tau_1 + \tau_2 + \tau_3 + \tau_4)^2 M_n^2 \Delta t_n^2 + (\tau_1 + \tau_2 + \tau_3 + \tau_4)$$

$$* (\tau_2 \mu_1 + \tau_3 \mu_2 + \tau_4) M_n' M_n \Delta t_n^3$$

$$+ \frac{1}{2} (\tau_1 + \tau_2 + \tau_3 + \tau_4) \left(\tau_2 \mu_1^2 + \tau_3 \mu_2^2 + \tau_4 \right) M_n'' M_n \Delta t_n^4$$

$$+ \frac{1}{2} (\tau_2 \mu_1 + \tau_3 \mu_2 + \tau_4)^2 (M'_n)^2 \Delta t_n^4 + \frac{1}{6} (\tau_1 + \tau_2 + \tau_3 + \tau_4)^3 M_n^3 \Delta t_n^3$$

$$+ \frac{1}{2} (\tau_1 + \tau_2 + \tau_3 + \tau_4)(\tau_2 \mu_1 + \tau_3 \mu_2 + \tau_4) M'_n M_n^2 \Delta t_n^4$$

$$+ \frac{1}{24} M_n^4 \Delta t_n^4 \Bigg] \mathbf{u}_n + O\left(\Delta t_n^5\right).$$

$$(2.34)$$

That is,

$$\mathbf{u}_{n+1} = \left[1 + J_n \Delta t_n + \frac{1}{2} (J_n \Delta t_n)^2 + \frac{1}{3!} (J_n \Delta t_n)^3 + \frac{1}{4!} (J_n \Delta t_n)^4 \right] \mathbf{u}_n + O\left(\Delta t_n^5\right), \qquad (2.35)$$

where

$$J_n = \tau_1 M_n + \tau_2 M_{n+\mu_1} + \tau_3 M_{n+\mu_2} + \tau_4 M_{n+1},$$

$$\tau_2 = \frac{2\mu_2 - 1}{12(-\mu_2 + \mu_1)(\mu_1 - 1)\mu_1},$$

$$\tau_3 = -\frac{-1 + 2\mu_1}{12\mu_2(\mu_1\mu_2 + \mu_2 - \mu_2^2 - \mu_1)}, \qquad (2.36)$$

$$\tau_4 = \frac{6\mu_1\mu_2 - 4\mu_2 + 3 - 4\mu_1}{12(\mu_2 - 1)(\mu_1 - 1)}.$$

We now turn to examine the more general situation in which the reaction term is not zero, that is, $\mathbf{b} \neq 0$. In this case, we have

$$\mathbf{u}_{n+1} = \left(\sum_{i=0}^{m} \frac{(\Delta t_n J_n)^i}{i!} \right) \mathbf{u}_n + \mathbf{F}(\mathbf{b}_n, \mathbf{b}_{n+\mu_1}, \dots, \mathbf{b}_{n+1}) + O\left(\Delta t_n^{i+1}\right). \qquad (2.37)$$

Similar to our discussion in the previous section, the accumulation error for (2.37) is

$$|\mathbf{u}_{n+1} - \mathbf{v}_{n+1}| = \left| \sum_{i=0}^{m} \frac{(\Delta t_n J_n)^i}{i!} \right| \times |\mathbf{u}_n - \mathbf{v}_n| + O\left(\Delta t_n^{i+1}\right). \qquad (2.38)$$

We remark again that the amplification factor $\sum_{i=0}^{m} ((\Delta t_n J_n)^i / i!)$ in (2.38) at the nth local step has the same form with (2.5) except M being replaced by J_n. Therefore, we can apply the strategy demonstrated in Theorems 2.5 and 2.6 to look for the optimal time steps for Runge-Kutta schemes. The present $a_{\max}$ and $a_{\min}$ appeared in the optimal step size formulas then, respectively, represent the greatest and smallest real parts of the eigenvalues of the matrix J_n. Notice that by the similar argument outlined in the previous proofs, the stability is preserved with this step size.

An Explicit Adaptive Optimal Time Stepping Scheme for Solving (2.18)

(1) Set up initial condition $\mathbf{u_0}$ and initial step size Δt_0.

(2) Set J_n in accordance with the stage number m, and apply m-stage Runge-Kutta schemes.

(3) Use the Gerschgorin theorem to estimate the interval, $[-a_{\max}^n, -a_{\min}^n]$ for negative real parts of the eigenvalues of J_n.

(4) Place Δt_{n+1},

(i) $m = 2$: $\Delta t_{n+1} = 2/(a_{\max}^n + a_{\min}^n)$,

(ii) $m = 3$: $\Delta t_{n+1} = 2.512745327/a_{\max}^n$,

(iii) $m = 4$:

$$
\begin{aligned}
\Delta t_{n+1} = \frac{1}{3}\Bigg(& 172 a_{\max}^6 + 192 a_{\max}^5 a_{\min} + 276 a_{\max}^4 a_{\min}^2 + 448 a_{\max}^3 a_{\min}^3 \\
& + 276 a_{\min}^4 a_{\max}^2 + 192 a_{\max} a_{\min}^5 + 172 a_{\min}^6 \\
& + 36\Big(29 a_{\max}^6 + 30 a_{\max}^5 a_{\min} + 51 a_{\max}^4 a_{\min}^2 + 68 a_{\max}^3 a_{\min}^3 + 51 a_{\min}^4 a_{\max}^2 \\
& \qquad + 30 a_{\max} a_{\min}^5 + 29 a_{\min}^6 \Big)^{1/2} a_{\max}^3 \\
& + 36\Big(29 a_{\max}^6 + 30 a_{\max}^5 a_{\min} + 51 a_{\max}^4 a_{\min}^2 + 68 a_{\max}^3 a_{\min}^3 + 51 a_{\min}^4 a_{\max}^2 \\
& \qquad + 30 a_{\max} a_{\min}^5 + 29 a_{\min}^6 \Big)^{1/2} a_{\min} a_{\max}^2 \\
& + 36\Big(29 a_{\max}^6 + 30 a_{\max}^5 + 51 a_{\max}^4 a_{\min}^2 + 68 a_{\max}^3 a_{\min}^3 + 51 a_{\min}^4 a_{\max}^2 \\
& \qquad + 30 a_{\max} a_{\min}^5 + 29 a_{\min}^6 \Big)^{1/2} a_{\max} a_{\min}^2 \\
& + 36\Big(29 a_{\max}^6 + 30 a_{\max}^5 a_{\min} + 51 a_{\max}^4 a_{\min}^2 + 68 a_{\max}^3 a_{\min}^3 + 51 a_{\min}^4 a_{\max}^2 \\
& \qquad + 30 a_{\max} a_{\min}^5 + 29 a_{\min}^6 \Big)^{1/2} a_{\min}^3 \Bigg)^{1/3} \Bigg/ \left(a_{\max}^3 + a_{\min} a_{\max}^2 + a_{\max} a_{\min}^2 + a_{\min}^3 \right) \\
& - \frac{4}{3}\left(5 a_{\max}^4 + 10 a_{\max}^3 a_{\min} + 6 a_{\max}^2 a_{\min}^2 + 10 a_{\max} a_{\min}^3 + 5 a_{\min}^4 \right) \Bigg/ \\
& \left(\left(a_{\max}^3 + a_{\min} a_{\max}^2 + a_{\max} a_{\min}^2 + a_{\min}^3 \right) \right. \\
& \qquad \times \left(172 a_{\max}^6 + 192 a_{\max}^5 a_{\min} + 276 a_{\max}^4 a_{\min}^2 + 448 a_{\max}^3 a_{\min}^3 \right.
\end{aligned}
$$

$$+ 276a_{\min}^4 a_{\max}^2 + 192a_{\max}a_{\min}^5 + 172a_{\min}^6$$

$$+ 36\Big(29a_{\max}^6 + 30a_{\max}^5 a_{\min} + 51a_{\max}^4 a_{\min}^2 + 68a_{\max}^3 a_{\min}^3 + 51a_{\min}^4 a_{\max}^2$$

$$+30a_{\max}a_{\min}^5 + 29a_{\min}^6\Big)^{1/2} ar^3$$

$$+ 36\Big(29a_{\max}^6 + 30a_{\max}^5 a_{\min} + 51a_{\max}^4 a_{\min}^2 + 68a_{\max}^3 a_{\min}^3 + 51a_{\min}^4 a_{\max}^2$$

$$+30a_{\max}a_{\min}^5 + 29a_{\min}^6\Big)^{1/2} a_{\min}a_{\max}^2$$

$$+ 36\Big(29a_{\max}^6 + 30a_{\max}^5 a_{\min} + 51a_{\max}^4 a_{\min}^2 + 68a_{\max}^3 a_{\min}^3 + 51a_{\min}^4 a_{\max}^2$$

$$+30a_{\max}a_{\min}^5 + 29a_{\min}^6\Big)^{1/2} a_{\max}a_{\min}^2$$

$$+ 36\Big(29a_{\max}^6 + 30a_{\max}^5 a_{\min} + 51a_{\max}^4 a_{\min}^2 + 68a_{\max}^3 a_{\min}^3 + 51a_{\min}^4 a_{\max}^2$$

$$+30a_{\max}a_{\min}^5 + 29a_{\min}^6\Big)^{1/2} a_{\min}^3\Big)^{1/3}\Big)$$

$$+ \frac{4}{3}\Big(a_{\max}^2 + a_{\max}a_{\min} + a_{\min}^2\Big)\Big/\Big(a_{\max}^3 + a_{\min}a_{\max}^2 + a_{\max}a_{\min}^2 + a_{\min}^3\Big),$$

$$(2.39)$$

and then let $t_{n+1} = t_n + \Delta t_{n+1}$. When an $a_{\max}^n$ is equal to or close to zero, simply choose a reasonable step size for Δt_{n+1}.

Note that the time step for $m = 4$ can be solved by mathematical software such as MAPLE from the following equation:

$$\sum_{i=1}^{4} \frac{1}{i!}\Big(a_{\max}^i - a_{\min}^i\Big)\Delta t_{n+1}^{i-1} = 0. \qquad (2.40)$$

2.2.1. The "Best" Optimal Step Size and the Time-Dependent Homogeneous Systems

We point out that for those eigenvalues with negative real parts, one may also assume they are enclosed in an ellipse with one of the vertex lying at the origin, that is, $a_{\min}^n = 0$. In this way, only $a_{\max}^n$ are needed in forming optimal time step formulas and time steps Δt_{n+1} in step 4 are switched as

(i) $m = 2$: $\Delta t_{n+1} = 2/a_{\max}^n$,

(ii) $m = 3$: $\Delta t_{n+1} = 2.512745327/a_{\max}^n$,

(iii) $m = 4$: $\Delta t_{n+1} = 2.785293563/a_{\max}^n$,

where $a_{\max}^n$ is defined as before in step 2. We remark that when Δt_{n+1} is specified as above values, $p_m(a, \Delta t)$ is equal to one. The stability condition does not seem to be satisfied. However, this hidden problem is avoidable by requiring Gerschgorin theorem or other

equivalent theorems in step 3 as an estimation for the distribution of real part of eigenvalues. This is because the theorem provides an overestimation for the distribution of negative real part of eigenvalues. As a result, $p_m(a, \Delta t)$ is less than one. Hence, it is safe to use the above time step.

In particular, when the reaction vector is negligible, we can further write the m-stage Runge-kutta method as some equivalent m-stage formulas while the order of accuracy remains invariant. For example, $m = 3$, we have

$$\mathbf{u}_n^{(0)} = \mathbf{u}_n + \frac{\Delta t_n}{3} J_n \mathbf{u}_n,$$
$$\mathbf{u}_n^{(1)} = \mathbf{u}_n + \frac{\Delta t_n}{2} J_n \mathbf{u}_n^{(0)}, \qquad (2.41)$$
$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t J_n \mathbf{u}_n^{(1)}.$$

For $m = 4$,

$$\mathbf{u}_n^{(0)} = \mathbf{u}_n + \frac{\Delta t_n}{4} J_n \mathbf{u}_n,$$
$$\mathbf{u}_n^{(1)} = \mathbf{u}_n + \frac{\Delta t_n}{3} J_n \mathbf{u}_n^{(0)},$$
$$\mathbf{u}_n^{(2)} = \mathbf{u}_n + \frac{\Delta t_n}{2} J_n \mathbf{u}_n^{(1)}, \qquad (2.42)$$
$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t_n J_n \mathbf{u}_n^{(2)}.$$

Of course, J_n is defined according to their stage number.

3. Numerical Examples

3.1. Overview

In this section, we demonstrate the implementation and the efficiency of our optimal adaptive time stepping method by setting a one-dimension testing problem and simulating some problems from a series of real biological experiments. The 1D testing problem with the known analytic solution is used to show the implementation process of our scheme. It is also designed to show the rate of convergence as well as the computational cost. Those results are illustrated through comparison with standard m-stage Runge-Kutta methods and the Runge-Kutta-Fehlberg (RKF) method. To test our numerical methods on real biological experiments, we apply our optimal adaptive two-stage Runge-Kutta method to a two-dimensional reaction-diffusion-chemotaxis model for a series of biological experiments on bacterial chemotaxis reported in [9]. Some results will be shown here.

3.2. Testing Problems

Consider

$$u_t = u_{xx} - (txu)_x + f, \quad x \in (0, 2\pi), \ t > 0, \tag{3.1}$$

where $u = u(x,t)$ and $f = (tx - 1)e^{-t}\cos(x - t) + te^{-t}\sin(x - t)$. The three terms on the right-hand side represent diffusion, chemotaxis, and reaction, respectively. Assume the periodic boundary condition: $u(o,t) = u(2\pi, t), t \geq 0$ and the initial condition: $u(x,0) = \sin x$. The analytic solution of (3.1) is

$$u(x,t) = e^{-t}\sin(x - t). \tag{3.2}$$

To apply MOL, we simply use the difference scheme for the spatial discretization. The standard second-order difference is applied to the diffusion term. The upwind scheme is employed in the chemotaxis for some stability consideration [17]. After the discretization, we then apply m-stage Runge-Kutta method with the adaptive optimal time stepping scheme as the time integration strategy. By denoting $h = 2\pi/N$, $x_i = ih$, $i = 0, 1, \ldots, N$, and the numerical solution as u_i, the resulting ODE system is

$$\frac{d\mathbf{u}}{dt} = M(t)\mathbf{u} + \mathbf{f(t)}, \tag{3.3}$$

where

$$\mathbf{u} = (u_0, u_1, \ldots, u_N)^T,$$

$$\mathbf{f(t)} = \left(f(x_0, t), f(x_1, t), \ldots, f(x_N, t)\right)^T,$$

$M(t_n)$

$$= \begin{pmatrix}
\dfrac{-2}{h^2} - \dfrac{x_0 t_{n-1}}{h} & \dfrac{1}{h^2} & & & \dfrac{1}{h^2} - t_{n-1} & 0 \\
\dfrac{1}{h^2} + \dfrac{x_0 t_{n-1}}{h} & \dfrac{-2}{h^2} - \dfrac{x_1 t_{n-1}}{h} & \dfrac{1}{h^2} & & 0 & 0 \\
& \dfrac{1}{h^2} + \dfrac{x_1 t_{n-1}}{h} & \dfrac{-2}{h^2} - \dfrac{x_2 t_{n-1}}{h} & \dfrac{1}{h^2} & 0 & 0 \\
& & \ddots & \ddots & \ddots & \\
& & & \ddots & \ddots & \ddots \\
& & & \ddots & \ddots & & \ddots \\
0 & & \dfrac{1}{h^2} & & 0 & \dfrac{1}{h^2} + \dfrac{x_{N-1} t_{n-1}}{h} & \dfrac{-2}{h^2} - \dfrac{x_N t_{n-1}}{h}
\end{pmatrix}. \tag{3.4}$$

Generally, the actual eigenvalues of the spatial matrix $M(t)$ are not available. One may use the power method and the inverse power method [20] to estimate the largest and smallest magnitude of eigenvalues. But these methods are usually time consuming. Moreover, only negative real parts of the eigenvalues are needed for our scheme. Therefore, we suggest using Gerschgorin theorem to estimate the distribution of eigenvalues. As we should show immediately, it is a simple and inexpensive way to find the range for real parts of the eigenvalues. It is therefore adequate for implementing our scheme. In fact, if a formula for the distribution of eigenvalues or actual eigenvalues can be found ahead of time, then it is not necessary to use the theorem to estimate the range at each time step. A simple version of the Gerschgorin theorem is stated as follows.

Theorem 3.1 (Gerschgorin). *The union of all discs*

$$K_i := \left\{ \mu \in |\mu - a_{ii}| \leq \sum_{k=1,\, k \neq i}^{n} |a_{ik}| \right\} \tag{3.5}$$

contains all eigenvalues of the $n \times n$ matrix $A = [a_{ik}]$.

The theorem states that if A is a real matrix, the range of the real parts of the eigenvalues can be easily detected by checking the intersection intervals of these discs with the real axis.

From Gerschgorin Theorem, we obtain

$$-a_{\max} \leq \mathrm{Re}(\lambda) < 0, \tag{3.6}$$

where

$$a_{\max} = \frac{4}{\Delta x^2} + t_n \left(\frac{4\pi}{\Delta x} + 1 \right). \tag{3.7}$$

Thus, for m-stages, where $m = 2, \ldots, 4$, the optimal time steps at the current step are

$$m = 2: \ \Delta t_{\mathrm{opt}} = \frac{2}{(4/\Delta x^2) + t_n((4\pi/\Delta x) + 1)},$$

$$m = 3: \ \Delta t_{\mathrm{opt}} = \frac{2.512745327}{(4/\Delta x^2) + t_n((4\pi/\Delta x) + 1)}, \tag{3.8}$$

$$m = 4: \ \Delta t_{\mathrm{opt}} = \frac{2.785293563}{(4/\Delta x^2) + t_n((4\pi/\Delta x) + 1)}.$$

Figure 3 shows the relative error between the numerical and the analytic solutions. Observe that the relative error for the standard two-stage Runge-Kutta method increases dramatically around $t = 5.2$, but the optimal time-stepping m-stage Runge-Kutta methods still converge to the analytic solution.

Tables 1, 2, and 3 demonstrate the number of iteration for our scheme (Optimal m-stage RK) with that of two other methods: standard m-stage Runge-Kutta method with fixed
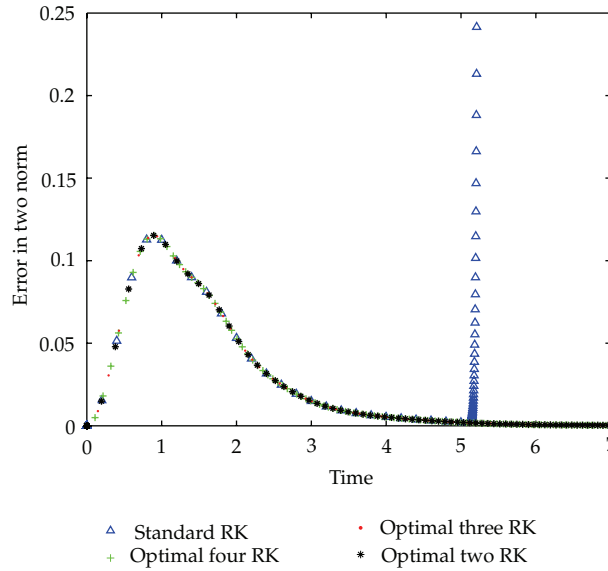
Figure 3: Relative error between numerical simulations and the analytic solution.

Table 1: Comparison for number of iterations among optimal two-stage RK, RKF 23, and standard two-stage RK. The optimal two-stage RK method is counted twice for the function evaluation per iteration, while the RKF 23 is counted four times per iteration. "Div" is the abbreviation for Divergent.

Method/Time	1	2	3	4	5
Optimal two-stage RK	638	1278	1918	2558	3200
RKF 23	1175	2765	4310	6075	8250
Std two-stage RK ($\Delta t = 0.0025$)	800	1600	2400	Div	Div

Table 2: Comparison for number of iterations among the optimal three-stage RK, RKF 34, and standard three-stage RK. The optimal three-stage RK method is counted three times for the function evaluation per iteration, while the RKF 34 is five times per iteration.

Method/Time	1	2	3	4	5
Optimal three-stage RK	678	1524	2535	3717	5067
RKF 34	3925	6325	7935	9655	11570
Std three-stage RK ($\Delta t = 0.003$)	1002	2001	3000	Div	Div

step size (Std m-stage RK) and the Runge-Kutta-Fehlberg methods (RKF). These tables basically indicate that while the optimal time stepping scheme always ensure convergence, the standard m-stage Runge-Kutta methods or other adaptive methods like RKF either diverge or converge slower since they require a larger number of iterations. Similar results can also be illustrated by calculating CPU time following from Tables 4, 5, and 6. The computations were done on Dell Latitude D620 2600.

Tables 4–6 illustrate the computational efficiency for our method with CPU time.

3.3. Further Discussion on the Efficiency of the Optimal RK Method

We continue our discussion about the computational efficiency for optimal RK methods with time steps mentioned in Sections 2.2. We are interested in measuring the efficiency difference

Table 3: Comparison for number of iterations among the optimal four-stage RK, RKF 45, and standard four-stage RK. The optimal four-stage RK method is counted four times for the function evaluation per iteration while the RKF 45 is six times per iteration.

Method/Time	1	2	3	4	5
Optimal four-stage RK	816	1832	3052	4472	6096
RKF 45	3480	5598	7152	8910	10866
Std four-stage RK ($\Delta t = 0.004$)	1000	2000	Div	Div	Div

Table 4: Comparison of CPU time among the optimal two-stage RK, RKF 23, and standard two-stage RK with $\Delta t = 0.001$.

CPU/Time	10	20	30	40	50
Optimal two-stage RK	0.5940	1.8430	3.7960	6.4690	9.7820
RKF 23	1.3910	3.9060	7.4530	12.0620	17.6090
Std two-stage RK	0.9840	3.5310	Div	Div	Div

Table 5: Comparison of CPU time among the optimal three-stage RK, RKF 34, and standard three-stage RK with $\Delta t = 0.0005$ and $\Delta t = 0.001$, respectively.

CPU/Time	10	20	30	40	50
Optimal three-stage RK	0.8910	2.8280	5.7970	9.7970	15.0000
RKF 34	1.7190	4.5150	8.5630	13.6880	20.0780
Std three-stage RK	3.1710	6.6410	11.0630	13.1250	19.2340
Std three-stage RK	1.6880	3.310	Div	Div	Div

Table 6: Comparison of CPU time among the optimal four-stage RK, RKF 45, and standard four-stage RK with $\Delta t = 0.001$.

CPU/Time	10	20	30	40	50
Optimal four-stage RK	1.0000	3.2030	6.5780	11.1560	16.9530
RKF 45	1.8750	5.0000	9.5000	15.3280	23.6410
Std four-stage RK	2.2660	4.6250	Div	Div	Div

between RKF and our method. To make one's point, we take the optimal four-stage RK method as an example. Let us start with the "best" optimal step size as in Section 2.2.1. Then the current time step for the optimal four-stage RK is

$$\Delta t = \frac{2.785293563}{a_{max}}. \tag{3.9}$$

To get the "worst" case, set $a_{max} = a_{min}$. Then

$$\Delta t = \frac{1.596071638}{a_{max}}. \tag{3.10}$$

Now, let us assign Δt to be the average of (3.9) and (3.10). We obtain

$$\Delta t = \frac{2.1906826005}{a_{max}}. \tag{3.11}$$

Table 7: Comparison of CPU time between the optimal four-stage RK with the average time step of the "best" and the "worst" cases, and the RKF 45.

CPU/Time	10	20	30	40	50
Optimal four-stage RK	1.0000	3.2030	6.5780	11.1560	16.9530
Four-stage RK Δt = (3.11)	1.2660	4.0470	8.3600	14.2190	21.5160
RKF 45	1.8750	5.0000	9.5000	15.3280	23.6410

Table 7 shows that even with the "medium" optimal time step (3.11), our numerical scheme still performs better than the RKF 45. In addition, the computational efficiency of RKF 45 is approximately equal to the optimal four-stage RK when the time step is chosen as (3.11).

3.4. Computer Simulations of Chemotaxis in Bacteria

Our numerical experiment is based on the biological experiments reported in (2.28) and the corresponding mathematical model given in (2.28). Interesting patterns are developed when *Escherichia coli* or *Salmonella typhimurium* is incoulated on intermediates of the tricarboxylic acid (TCA) cycle. Bacteria form symmetrical rings of spots or strips if they are inoculated in semisolid agar on the intermediates of TCA cycle. When placed in a liquid medium and fed on intermediates of the TCA cycle, they secrete aspartate as a chemoattractant. Cells then arrange themselves into high density aggregates. In this case, aggregates form randomly as time evolves and fade in a time interval less than the bacteria generation. Here, attractant chemicals are excreted by the bacterial themselves instead of being added into the Petri dish artificially.

A nondimensionalized mathematical models were proposed in (2.28) for the liquid medium experiments in (2.28). The model is a classical chemotaxis system, which reads as

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(D_u \nabla u - \frac{\alpha u}{(1+v)^2} \nabla v \right),$$

$$\frac{\partial v}{\partial t} = \Delta v + \omega \frac{u^2}{\mu + u^2},$$

(3.12)

where u, v, and ω represent cell density, concentration of chemotactic substance, and succinate, respectively. Simulations were done with zero flux boundary condition for both u and v. Initial conditions are set to be zero for v and the random perturbation at $u_0 = 1$ for the initial condition of u. The noise amplitude is of order 10^{-1}. They were created by the C^{++} random number generator. The initial, boundary conditions for u and v, and the noise amplitude are chosen to be the same as described in (2.28) except for the random numbers which may not be identical due to the use of different softwares.

Numerical simulations are showed in Figure 4. Simulations did capture the characteristics as described in (2.28) qualitatively. The aggregates first are divided into small regions and then cojoin with others.
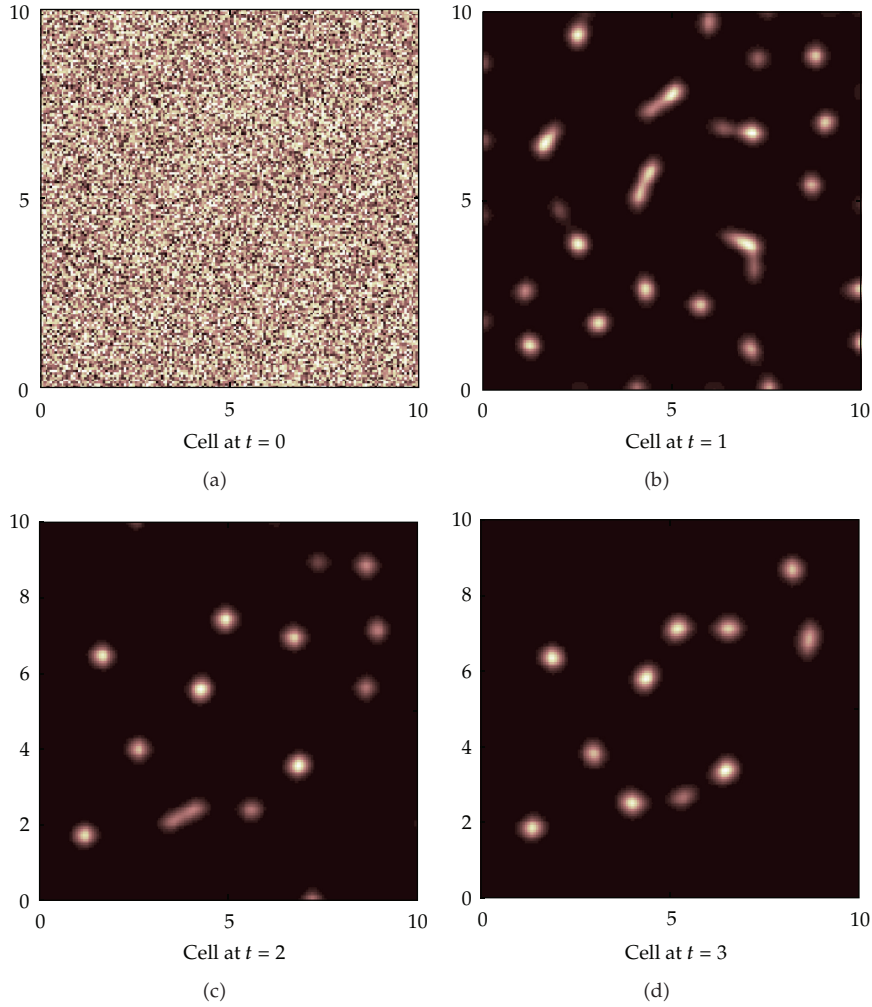
Figure 4: Numerical simulations for bacterial aggregates they exposed to TCA cycle: from left to right, top to bottom. The aggregates are separated into small regions and then cojoin other aggregates. The aggregates become fewer but larger.

4. Conclusions and Discussions

We investigate an optimal adaptive time stepping strategy for fully explicit m-stage Runge-Kutta methods to solve general reaction-diffusion-chemotaxis systems. It is designed specially for handling spatially nonhomogeneous pattern formations. The method is based on m-stage Runge-Kutta methods and simple estimation on the optimal step size at each iteration. With some optimization techniques, we efficiently discard the decay modes and only allow the growth modes to grow. In this way, we guarantee that the numerically generated patterns follow the primary model equations. The scheme is proven to be extremely robust and accurate by theoretical analysis, numerical experiments on the testing problem, and real pattern simulations on bacterial chemotaxis. The numerical experiments also show that our method is superior to RKF in solving reaction-diffusion-chemotaxis systems.

Acknowledgment

References

[1] E. Keller and L. Segel, "Model for chemotaxis," *Journal of Theoretical Biology*, vol. 30, pp. 225–234, 1971.

[2] D. Lauffenburger, C. Kennedy, and R. Aris, "Traveling bands of chemotactic bacteria in the context of population growth," *Bulletin of Mathematical Biology*, vol. 46, pp. 19–40, 1984.

[3] E. O. Budrene and H. C. Berg, "Complex patterns formed by motile cells of Escherichia Coli," *Nature*, vol. 349, no. 6310, pp. 603–633, 1991.

[4] C. Chiu and F. C. Hoppensteadt, "Mathematical models and simulations of bacterial growth and chemotaxis in a diffusion gradient chamber," *Journal of Mathematical Biology*, vol. 42, no. 2, pp. 120–144, 2001.

[5] C. Chiu and J. L. Yu, "An optimal adaptive time-stepping scheme for solving reaction-diffusion-chemotaxis systems," *Mathematical Biosciences and Engineering*, vol. 4, no. 2, pp. 187–203, 2007.

[6] E. Ben-Jacob, I. Cohen, I. Golding, and Y. Kozlovsky, "Modeling branching and chiral colonial patterning of lubricating bacteria," in *Mathematical Models for Biological Pattern Formation*, K. M. Philip and G. O. Hans, Eds., vol. 121 of *The IMA Volumes in Mathematics and Its Applications*, pp. 211–253, Springer, New York, NY, USA, 2001.

[7] R. Tyson, S. R. Lubkin, and J. D. Murray, "Model and analysis of chemotactic bacterial patterns in a liquid medium," *Journal of Mathematical Biology*, vol. 38, no. 4, pp. 359–375, 1999.

[8] J. Yu, *A fully explicit optimal two-stage numerical scheme for solving reaction-diffusion-chemotaxis systems*, Ph.D. thesis, Michigan State University, 2005.

[9] J. Adler, "Chemotaxis in bacteria," *Science, New Series*, vol. 153, no. 3737, pp. 708–716, 1966.

[10] E. O. Budriené, A. A. Polezhaev, and M. O. Ptitsyn, "Mathematical modelling of intercellular regulation causing the formation of spatial structures in bacterial colonies," *Journal of Theoretical Biology*, vol. 135, no. 3, pp. 323–341, 1988.

[11] J. D. Murray, *Mathematical Biology*, vol. 19 of *Biomathematics*, Springer, Berlin, Germany, 3rd edition, 1993.

[12] P. C. Fife, *Mathematical Aspects of Reacting and Diffusing Systems*, vol. 28 of *Lecture Notes in Biomathematics*, Springer, Berlin, Germany, 1979.

[13] J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, vol. 258 of *Grundlehren der Mathematischen Wissenschaften*, Springer, New York, NY, USA, 1983.

[14] R. Tyson, L. G. Stern, and R. J. LeVeque, "Fractional step methods applied to a chemotaxis model," *Journal of Mathematical Biology*, vol. 41, no. 5, pp. 455–475, 2000.

[15] C. Chiu, F. C. Hoppensteadt, and W. Jäger, "Analysis and computer simulation of accretion patterns in bacterial cultures," *Journal of Mathematical Biology*, vol. 32, pp. 841–855, 1994.

[16] C. Chiu and N. Walkington, "An ADI method for hysteretic reaction-diffusion systems," *Society for Industrial and Applied Mathematics Journal on Numerical Analysis*, vol. 34, no. 3, pp. 1185–1206, 1997.

[17] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, Calif, USA, 1989.

[18] G. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.

[19] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1989.

[20] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer, New York, NY, USA, 2nd edition, 1992.