

Research Article

Parareal Algorithms Implemented with IMEX Runge-Kutta Methods

Zhiyong Wang¹ and Shulin Wu²

¹School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 610731, China

²School of Science, Sichuan University of Science and Engineering, Zigong, Sichuan 643000, China

Correspondence should be addressed to Zhiyong Wang; uestczhywang@gmail.com

Received 17 June 2014; Revised 31 August 2014; Accepted 1 September 2014

Academic Editor: Zhike Peng

Copyright © 2015 Z. Wang and S. Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Parareal algorithm is a very powerful parallel computation method and has received much interest from many researchers over the past few years. The aim of this paper is to investigate the performance of parareal algorithm implemented with IMEX Runge-Kutta (RK) methods. A stability criterion of the parareal algorithm coupled with IMEX RK methods is established and the advantage (in the sense of stability) of implementing with this kind of RK methods is numerically investigated. Finally, numerical examples are given to illustrate the efficiency and performance of different parareal methods.

1. Introduction

The parareal algorithm was presented by Lions et al. in [1] as a numerical method to solve time dependent problems in parallel. The peculiarity of this algorithm is that it approximates successfully the solution later in time before having fully accurate approximations from earlier time points. This algorithm is becoming more and more popular in scientific and engineering computation and many excellent results have been obtained.

As mentioned above, the parareal algorithm was first introduced in [1] and an improved version which aims to solve nondifferentiable PDEs was discussed by Bal and Maday in [2]. Some further modifications and improvements can be found in [3], in which the authors use the “filtering” technique to overcome the so-called *resonance* or *beating* phenomenon (see also [4, 5]) that triggers numerical instability when the algorithm is applied to linear structure dynamics. The stability and convergence are the main subjects of theoretical analysis of the algorithm which have been investigated widely by many researchers; see, for example, [6–8]. Nowadays, this algorithm, as well as its variants [3, 9–12], has been used in many fields by many researchers, such as morphological transformation simulations [13], structural (fluid) dynamics

simulations [4, 5, 14], optimal control [15, 16], Hamiltonian simulations [17, 18], turbulent plasma simulations [19, 20], and solution of Volterra integral equations [21]. (There is an increasing interest in parareal, and it is very possible that some important references are not mentioned here.)

For ODEs,

$$\begin{aligned} y'(t) &= f(t, y(t)), \quad t \in [0, T], \\ y(0) &= y_0, \end{aligned} \tag{1}$$

to formulate the parareal algorithm, the time domain $[0, T]$ of interest is first partitioned into several time-slices whose boundary points are treated as coarse time-grids. And then, the algorithm consists of three basic steps. First, using some numerical propagator, say $\mathcal{G}_{\Delta T}$, the solution is approximated on each coarse time-grid to provide a seed—that is, an initial condition—to the time-slice. Second, another propagator, denoted by $\mathcal{F}_{\Delta t}$, is applied independently and therefore concurrently in each time-slice to advance the solution from the starting point of this time-slice to its end point. Finally, an iterative process is invoked to improve the accuracy of the seeds and eliminate the jumps of the solution on the coarse time-grids. In most cases, the numerical propagators $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$ are defined by traditional RK method with

equal length of the time-slices, such as Radau methods, Lobatto methods, and Gauss methods, and there are lots of investigations about theoretical properties and performance of practical application of the algorithm in such case. See, for example, [4, 5, 7, 9, 14] and references therein.

There is also some research about special choice of the underlying numerical methods that are used to define $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$. For example, Guibert and Tromeur-Dervout [22] considered the adaptive time-slices (i.e., the length of time-slices is not equal) to define $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$ and the derived parareal algorithm can be applied to strong stiff ODEs and differential algebraic equations (DAEs). Another example is the one investigated by Minion [11], where the authors defined the numerical propagators $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$ by spectral deferred correction based on the Gaussian quadrature spectral integration. It was shown that the iterative process based on spectral deferred correction rather than traditional methods within a parareal framework results in a significant decrease in the overall computational cost of the algorithm. Wu et al. [9] suggested that the propagator $\mathcal{F}_{\Delta t}$ was defined by local Richardson extrapolation, and it turns out that the Parareal-Richardson method has advantages of higher accuracy and better stability.

In some cases, the function f in (1) can be written into two parts

$$f = f_s + f_{ns}, \quad (2)$$

where f_{ns} is the nonstiff or mildly stiff part of f and f_s is the stiff part. A common example for such case arises from semidiscretization of reaction diffusion equations $u_t = du_{xx} + r(t, x, u)$. In this case the function f in (1) can be written as $f(t, y(t)) = Ay(t) + f_{ns}(t, y(t))$ with A being a tridiagonal matrix and stiff. In the case that f can be partitioned into stiff and nonstiff parts, it is more advisable to use the implicit-explicit (IMEX) RK methods (or additive RK methods called sometimes) to solve ODEs (1). For this type of RK methods, the stiff part is assigned with the implicit component of the IMEX RK method to satisfy stability requirement and the nonstiff or mildly stiff part is assigned with the explicit component to reduce computational cost. For more details about the RK methods and the IMEX RK methods, the interested reader may refer to [23–33] and references therein.

Therefore, under condition (2), we think that it is valuable to adopt the IMEX RK methods instead of the traditional RK methods to define $\mathcal{G}_{\Delta T}$ and/or $\mathcal{F}_{\Delta t}$ used in parareal framework. This is the main motivation of our work. In certain aspects, this paper should be viewed as an experimental one. While the stability of the derived algorithm we present is easily proven by the results given by Gander and Vandewalle [7], stability region of the parareal algorithm defined by the IMEX RK methods is numerically shown.

The structure of this paper is as follows. In Section 2, the parareal algorithm and the IMEX RK methods are revisited. Section 3 is concerned with the stability of the derived parareal algorithm. Several combinations of

different IMEX RK methods and the corresponding advantages/disadvantages are numerically shown. Finally, in Section 4, we test the Gray-Scott model arising in chemical reaction to illustrate our results.

2. The Parareal Algorithm and the IMEX RK Methods

In this section, we revisit the parareal algorithm and the IMEX RK methods.

2.1. The Parareal Algorithm. To introduce the parareal algorithm, let us first partition time interval $[0, T]$ into N subintervals $S_n = [T_n, T_{n+1}], n = 0, 1, \dots, N - 1$, with equal length ΔT and $T_n = n\Delta T$. We call T_n 's the coarse time-grids. We then use some finer step size $\Delta t = \Delta T/M$ ($M > 1$ is an integer) to partition each relative large interval S_n into M finer intervals $s_{n,m} = [T_{n+m/M}, T_{n+(m+1)/M}]$ with $T_{n+m/M} = T_n + m\Delta t$ and $m = 0, 1, \dots, M - 1$. Now, the parareal algorithm can be described as follows. We designate by symbol \ominus the time-sequential steps performed on the coarse time-grids and by symbol \oplus the parallel steps performed on the decomposed finer time-grids.

The Parareal Algorithm. Consider the following.

\ominus *Step 0 (initialization).* Perform sequential computation $Y_{n+1}^0 = \mathcal{G}_{\Delta T}(T_n, Y_n^0, \Delta T)$ with $Y_0^0 = y_0, n = 0, 1, \dots, N - 1$, for $k = 0, 1, \dots$

\oplus *Step 1.* Perform in subinterval $S_n = [T_n, T_{n+1}]$ the computation $\widehat{Y}_{n+(m+1)/M} = \mathcal{F}_{\Delta t}(T_{n+m/M}, \widehat{Y}_{n+m/M}, \Delta t)$ with initial value $Y_n^k, m = 0, 1, \dots, M - 1$.

\ominus *Step 2.* Perform sequential corrections $Y_{n+1}^{k+1} = \mathcal{G}_{\Delta T}(T_n, Y_n^{k+1}, \Delta T) + \widehat{Y}_{n+1} - \mathcal{G}_{\Delta T}(T_n, Y_n^k, \Delta T)$ with $Y_0^{k+1} = y_0, n = 0, 1, \dots, N - 1$.

\ominus *Step 3.* If for $n = 0, 1, \dots, N - 1$, Y_n^{k+1} satisfy some termination criteria, stop the iteration; else go to Step 1.

Here, the finer propagation operator $\mathcal{F}_{\Delta t}$ and the coarse propagation operator $\mathcal{G}_{\Delta T}$ relay on one-step numerical methods and if the underlying methods are implicit, in most cases (linear ODEs is an exception), nonlinear algebraic equations need to be solved.

The above algorithm can be written compactly as

$$\begin{aligned} Y_{n+1}^{k+1} = & \mathcal{G}_{\Delta T}(T_n, Y_n^{k+1}, \Delta T) + \mathcal{F}_{\Delta t}(T_n, Y_n^k, \Delta t) \\ & - \mathcal{G}_{\Delta T}(T_n, Y_n^k, \Delta T), \end{aligned} \quad (3)$$

where k is the iteration index. Note that for $k \rightarrow +\infty$ method (3) will generate a series of values Y_n which satisfy $Y_{n+1} = \mathcal{F}_{\Delta t}(T_n, Y_n, \Delta t)$. This implies that the converged solution Y_n will achieve the accuracy of the propagator $\mathcal{F}_{\Delta t}$.

2.2. The IMEX RK Methods. Let us consider a pair of two Runge-Kutta methods defined by the arrays

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & a_{33} & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & a_{ss} \\ \hline b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \quad (4)$$

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & \hat{a}_{21} & 0 & 0 & \cdots & 0 \\ c_3 & \hat{a}_{31} & \hat{a}_{32} & 0 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ c_s & \hat{a}_{s1} & \hat{a}_{s2} & \cdots & \hat{a}_{s,s-1} & 0 \\ \hline \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_{s-1} & \hat{b}_s \end{array}$$

with the same abscissae

$$c_i = \sum_{j=1}^i a_{ij} = \sum_{j=1}^{i-1} \hat{a}_{ij}, \quad i = 2, \dots, s. \quad (5)$$

The top formula of (4) determines a diagonally implicit (semi-implicit) RK method and the bottom formula is an explicit RK method. In addition, let $h > 0$ be a step size and define the step point $t_n = nh$ for integer n . Consider the following stiff/nonstiff partitioned ODEs:

$$\begin{aligned} y'(t) &= f_s(t, y(t)) + f_{ns}(t, y(t)), \quad t \in [0, T], \\ y(0) &= y_0, \end{aligned} \quad (6)$$

and by applying the top part of (4) to stiff component f_s and the bottom part to the nonstiff component, we obtain the following scheme:

$$\begin{aligned} K_i &= y_n + h \sum_{j=1}^i a_{ij} f_s(t_n + c_j h, K_j) \\ &\quad + h \sum_{j=1}^{i-1} \hat{a}_{ij} f_{ns}(t_n + c_j h, K_j), \quad i = 1, 2, \dots, s, \end{aligned}$$

$$\begin{aligned} y_{n+1} &= y_n + h \sum_{j=1}^s b_j f_s(t_n + c_j h, K_j) \\ &\quad + h \sum_{j=1}^s \hat{b}_j f_{ns}(t_n + c_j h, K_j), \end{aligned} \quad (7)$$

where $\sum_{j=1}^i (\cdot) = 0$ with $i < j$.

The order conditions for IMEX RK methods are very complicated and beyond the topic of this paper. For this aspect, we refer the interested reader to, for example, [26, 30]. At the moment, we introduce some frequently used IMEX methods with order $p = 1, 2, 3$. We will use the triplet $\text{IMEX}(s_{\text{im}}, s_{\text{ex}}, p)$ to identify scheme (4), where s_{im} is the number of stages of the implicit part, s_{ex} is the number of stages of the explicit part, and p is the order of the IMEX scheme.

First Order IMEX RK Methods. The most popular first order IMEX RK method is the IMEX Euler method:

$$\begin{array}{c|cc|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ \hline 0 & 1 & & 1 & 0 \end{array} \quad \underbrace{\text{IMEX Euler}}_{(8)}$$

Second Order IMEX RK Methods. The second order IMEX RK methods considered here are the IMEX trapezoidal scheme [34]:

$$\begin{array}{c|cc|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ \hline 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{array} \quad \underbrace{\text{IMEX trapezoidal}}_{(9)}$$

The IMEX trapezoidal scheme is a combination of the trapezoidal rule and Heun's second order method (the explicit trapezoidal rule).

Third Order IMEX RK Methods. For third order IMEX RK method, we consider the one constructed by Ascher et al. [23] (this scheme is denoted by $\text{IMEX}(4, 4, 3)$):

$$\begin{array}{c|cc|cc} 0 & 0 & 0 & 0 & 0 \\ a & 0 & a & 0 & 0 \\ 0.7179332607 & 0 & 0.2820667392 & a & 0 \\ 1 & 0 & 1.2084966490 & -0.644363171 & a \\ \hline 0 & 1.2084966490 & -0.644363171 & a & \end{array} \quad \begin{array}{c|cc|cc} 0 & 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 & 0 \\ 0.3212788860 & 0.3966543747 & 0 & 0 & 0 \\ -0.1058858296 & b & b & 0 & 0 \\ \hline 0 & 1.2084966490 & -0.644363171 & a & \end{array} \quad \underbrace{\text{IMEX}(4, 4, 3)}_{(10)}$$

where $a = 0.4358665215$ and $b = 0.552929147$. Both the implicit and explicit schemes of $\text{IMEX}(4, 4, 3)$ are third order RK methods, and the implicit scheme is L -stable.

To finish this section, we introduce the concept of stability of the IMEX RK methods which will be used in the stability analysis of the parareal algorithm.

The asymptotic behavior of the IMEX RK methods applied to (6) is analyzed on the basis of the scalar model equation:

$$y'(t) = \lambda y + \mu y, \quad \lambda, \mu \in \mathbb{C}, \quad (11)$$

which was proposed by Frank et al. [35] and Verwer and Sommeijer [36] and adopted by Koto [28, 37]. For model equation (11), from (7) we arrive at

$$\begin{aligned} K &= \mathbb{E} y_n + h\lambda AK + h\mu \widehat{A}K, \\ y_{n+1} &= y_n + h\lambda \mathbf{b}K + h\mu \widehat{\mathbf{b}}K, \end{aligned} \quad (12)$$

where we have used vector and matrix notations to denote the values shown in the Butcher tableau (4); that is, $A = (a_{ij})$, $\widehat{A} = (\widehat{A}_{ij})$, $\mathbf{b} = (b_1, b_2, \dots, b_s)$, and $\widehat{\mathbf{b}} = (\widehat{b}_1, \widehat{b}_2, \dots, \widehat{b}_s)$. Moreover, $\mathbb{E} = (1, 1, \dots, 1)^T$.

Let $\alpha = h\lambda$, $\beta = h\mu$. We then have

$$y_{n+1} = R(\alpha, \beta) y_n, \quad (13)$$

where $R(\alpha, \beta)$ is called *stability function* of the IMEX RK method which is defined by

$$R(\alpha, \beta) = 1 + (\alpha \mathbf{b} + \beta \widehat{\mathbf{b}}) (I - \alpha A - \beta \widehat{A})^{-1} \mathbb{E}, \quad (14)$$

where I is an identity matrix of size s .

3. Stability of the Parareal Algorithm

In our analysis of the stability of the parareal algorithm implemented with IMEX RK methods, an important role is a strictly lower triangular Toeplitz matrix $\mathcal{M} := \mathcal{M}(\sigma)$ of size N . Its elements are defined by

$$\mathcal{M}_{ij} = \begin{cases} 0, & \text{if } i \leq j, \\ \sigma^{i-j-1}, & \text{if } i > j. \end{cases} \quad (15)$$

The next necessity for our analysis is the estimation of $\|\mathcal{M}^k\|_\infty$, which is proved by Gander and Vandewalle [7].

Lemma 1 (see [7]). *Let $k > 0$ be an integer. Then the infinite norm of \mathcal{M}^k can be estimated by*

$$\begin{aligned} &\|\mathcal{M}^k(\sigma)\|_\infty \\ &\leq \begin{cases} \min \left\{ \left(\frac{1 - |\sigma|^{N-1}}{1 - |\sigma|} \right)^k, \binom{N-1}{k} \right\}, & \text{if } |\sigma| < 1, \\ |\sigma|^{N-k-1} \binom{N-1}{k}, & \text{if } |\sigma| \geq 1. \end{cases} \end{aligned} \quad (16)$$

Assume that the stability functions of the underlying numerical methods used to define the two propagators $\mathcal{G}_{\Delta T}$

and $\mathcal{F}_{\Delta t}$ are $R_{\mathcal{F}}(\alpha, \beta)$ and $R_{\mathcal{G}}(\alpha, \beta)$, respectively. Then it follows by applying the parareal algorithm to the model equation (11) that

$$\begin{aligned} \mathcal{G}_{\Delta T}(T_n, Y_n^k, \Delta T) &= R_{\mathcal{G}}(\alpha, \beta) Y_n^k, \\ \mathcal{F}_{\Delta t}(T_n, Y_n^k, \Delta T) &= R_{\mathcal{F}}^M \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) Y_n^k, \quad k = 1, 2, \dots, \end{aligned} \quad (17)$$

where we recall $M = \Delta T / \Delta t$.

Theorem 2. *Let ΔT be given and $T_n = n\Delta T$ for $n = 0, 1, \dots$. Let the underlying numerical method used to define the propagator $\mathcal{G}_{\Delta T}$ running on the coarse time-grids be in its region of absolute stability; that is, $|R_{\mathcal{G}}| < 1$. Then one has the following estimates:*

$$\begin{aligned} &\sup_{n>0} |y(t_n) - Y_n^k| \\ &\leq \left(\frac{|R_{\mathcal{F}}^M(\alpha/M, \beta/M) - R_{\mathcal{G}}(\alpha, \beta)|}{1 - |R_{\mathcal{G}}(\alpha, \beta)|} \right)^k \sup_{n>0} |y(t_n) - Y_n^0|, \\ &\quad k \geq 1. \end{aligned} \quad (18)$$

Proof. We denote by e_n^k the error at iteration k of the parareal algorithm at coarse time point T_n ; that is, $e_n^k = y(t_n) - Y_n^k$. Then with an induction argument on n , it follows by applying the iterative process (3) to the model equation (11) that this error satisfies

$$\begin{aligned} e_{n+1}^{k+1} &= R_{\mathcal{G}}(\alpha, \beta) e_n^{k+1} + \left[R_{\mathcal{F}} \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) - R_{\mathcal{G}}(\alpha, \beta) \right] e_n^k \\ &= \left[R_{\mathcal{F}} \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) - R_{\mathcal{G}}(\alpha, \beta) \right] \sum_{j=1}^n R_{\mathcal{G}}^{n-j}(\alpha, \beta) e_j^k, \end{aligned} \quad (19)$$

where we have used the fact that $e_0^k = 0$ for any $k \geq 0$. Set $E^k = (e_1^k, e_2^k, \dots, e_N^k)^T$. Then relation (19) can be written in matrix form as

$$E^{k+1} = \left[R_{\mathcal{F}} \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) - R_{\mathcal{G}}(\alpha, \beta) \right] \mathcal{M}(R_{\mathcal{G}}(\alpha, \beta)) E^k, \quad (20)$$

where the matrix \mathcal{M} is defined by (15) with $\sigma = R_{\mathcal{G}}(\alpha, \beta)$. This implies

$$E^k = \left[R_{\mathcal{F}} \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) - R_{\mathcal{G}}(\alpha, \beta) \right]^k \mathcal{M}^k(R_{\mathcal{G}}(\alpha, \beta)) E^0. \quad (21)$$

Therefore, from Lemma 1 and the assumption $|R_{\mathcal{G}}| < 1$, we have

$$\begin{aligned} \|E^k\|_\infty &\leq \left| R_{\mathcal{F}} \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) - R_{\mathcal{G}}(\alpha, \beta) \right|^k \\ &\quad \times \|\mathcal{M}^k(R_{\mathcal{G}}(\alpha, \beta))\| \|E^0\|_\infty \\ &\leq \left| R_{\mathcal{F}} \left(\frac{\alpha}{M}, \frac{\beta}{M} \right) - R_{\mathcal{G}}(\alpha, \beta) \right|^k \\ &\quad \times \left(\frac{1 - |R_{\mathcal{G}}(\alpha, \beta)|^{N-1}}{1 - |R_{\mathcal{G}}(\alpha, \beta)|} \right)^k \|E^0\|_\infty. \end{aligned} \quad (22)$$

Hence, the proof of this theorem is completed by letting $N \rightarrow +\infty$ in the second inequality of (22). \square

The factor $\rho(\alpha, \beta) := |R_{\mathcal{F}}^M(\alpha/M, \beta/M) - R_{\mathcal{G}}(\alpha, \beta)|/(1 - |R_{\mathcal{G}}(\alpha, \beta)|)$ is called by Gander and Vandewalle [7] the *linear convergence factor* of the parareal algorithm performed on unbounded time intervals. According to the concepts introduced by Bal [6] and Staff and Rønquist [8], it is also called the stability function of the parareal algorithm. In this paper, we use the latter name.

Define

$$\mathcal{D} = \{(\alpha, \beta) \mid \rho(\alpha, \beta) < 1, \alpha, \beta \in \mathbb{C}\}. \quad (23)$$

Then if $(\alpha, \beta) \in \mathcal{D}$, the parareal algorithm is convergent on unbounded time intervals. In what follows of this paper, the set \mathcal{D} is called stability region of the parareal algorithm.

Remark 3. If $\mathbf{b} = \hat{\mathbf{b}}$ and $A = \hat{A}$ (i.e., the IMEX RK methods reduce to the traditional RK methods), it follows from (14) that $R(\alpha, \beta) = \mathcal{R}(z) = 1 + z\mathbf{b}(I - zA)^{-1}\mathbb{E}$ with $z = \alpha + \beta$. Therefore, if both the propagators $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$ are defined by the traditional RK methods, the stability function $\rho(\alpha, \beta)$ can be rewritten as $\tilde{\rho}(z) = |\mathcal{R}_{\mathcal{F}}^M(z/M) - \mathcal{R}_{\mathcal{G}}(z)|/(1 - |\mathcal{R}_{\mathcal{G}}(z)|)$, which is the one obtained by Gander and Vandewalle in [7].

From Remark 3, we see that the underlying numerical method used to define propagator $\mathcal{G}_{\Delta T}$ that is in its absolute stability region is necessary for the stability of the parareal algorithm. In fact, in the previous analysis and application of the parareal algorithm (i.e., both the propagators $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$ are defined by the traditional RK methods), it has been proposed by Farhat et al. [4, 5, 14] that the underlying RK method for $\mathcal{G}_{\Delta T}$ should be implicit and the one for the propagator $\mathcal{F}_{\Delta t}$ should be explicit. The reason is obvious—the implicit RK method is used to guarantee stability and the explicit RK method is used to reduce computation cost as much as possible.

The greatest advantage of using an IMEX RK method is that the stability of the combined scheme approaches to that of its implicit part while the computational cost can bear comparison with simply using the explicit part. Hence, in the parareal framework, it is naturally desired that the adoption of the IMEX RK methods will provide a far more stable parareal algorithm with computational cost sharply reduced.

To see whether our desirability can be carried through, for two IMEX RK methods, $\text{IMEX}_{\mathcal{G}}$ and $\text{IMEX}_{\mathcal{F}}$, we consider here $M = 50$ and the following four special combinations which leads to four parareal algorithms:

P_{IM-EX}: $\mathcal{G}_{\Delta T}$ is defined by the implicit part of $\text{IMEX}_{\mathcal{G}}$ and $\mathcal{F}_{\Delta t}$ is defined by the explicit part $\text{IMEX}_{\mathcal{F}}$;

P_{IM-IMEX}: $\mathcal{G}_{\Delta T}$ is defined by the implicit component of $\text{IMEX}_{\mathcal{G}}$ and $\mathcal{F}_{\Delta t}$ is defined by $\text{IMEX}_{\mathcal{F}}$;

P_{IMEX-EX}: $\mathcal{G}_{\Delta T}$ is defined by $\text{IMEX}_{\mathcal{G}}$ and $\mathcal{F}_{\Delta t}$ is defined by the explicit component of $\text{IMEX}_{\mathcal{F}}$;

P_{IMEX-IMEX}: $\mathcal{G}_{\Delta T}$ is defined by $\text{IMEX}_{\mathcal{G}}$ and $\mathcal{F}_{\Delta t}$ is defined by $\text{IMEX}_{\mathcal{F}}$.

For the convenience of plotting the stability region of the parareal algorithm, we consider the case $(\alpha, \beta) = (x, iy)$, $x, y \in \mathbb{R}$. The stability functions of the above four parareal algorithms are denoted by $\mathcal{R}_{\text{IM-EX}}(x, y)$, $\mathcal{R}_{\text{IM-IMEX}}(x, y)$, $\mathcal{R}_{\text{IMEX-EX}}(x, y)$, and $\mathcal{R}_{\text{IMEX-IMEX}}(x, y)$, respectively. The stability regions are denoted by $\mathcal{D}_{\text{IM-EX}}$, $\mathcal{D}_{\text{IM-IMEX}}$, $\mathcal{D}_{\text{IMEX-EX}}$, and $\mathcal{D}_{\text{IMEX-IMEX}}$, respectively. We remark that the **P_{IM-EX}** algorithm is a conventional parareal algorithm and we will compare its stability region with the ones of the other three algorithms.

We first illustrate the stability regions of the parareal algorithms when $\text{IMEX}_{\mathcal{G}}$ is chosen as the IMEX Euler method (see (8)). For $\text{IMEX}_{\mathcal{F}} = \text{IMEX Euler}$ and $\text{IMEX}_{\mathcal{F}} = \text{IMEX Trapezoidal}$ and $\text{IMEX}_{\mathcal{F}} = \text{IMEX}(4, 4, 3)$, the stability region of the algorithms **P_{IM-EX}**, **P_{IM-IMEX}**, **P_{IMEX-EX}**, and **P_{IMEX-IMEX}** is plotted in Figures 1(a), 1(b), and 1(c), respectively. For $\text{IMEX}_{\mathcal{G}} = \text{IMEX Euler}$, the stability functions of the parareal algorithms shown in Figures 1(a) and 1(b) are

Figure 1(a):

$$\mathcal{R}_{\text{IM-EX}}(x, y) = \frac{|1/(x + iy - 1) - (1 + (x + iy)/M)^M|}{1 - |1/(x + iy - 1)|},$$

$$\mathcal{R}_{\text{IM-IMEX}}(x, y) = \frac{|1/(x + iy - 1) - ((M + iy)/(M - x))^M|}{1 - |1/(x + iy - 1)|},$$

$$\mathcal{R}_{\text{IMEX-EX}}(x, y)$$

$$= \frac{|(1 + iy)/(1 - x) - (1 + (x + iy)/M)^M|}{1 - |(1 + iy)/(1 - x)|},$$

$$\mathcal{R}_{\text{IMEX-IMEX}}(x, y)$$

$$= \frac{|(1 + iy)/(1 - x) - ((M + iy)/(M - x))^M|}{1 - |(1 + iy)/(1 - x)|},$$

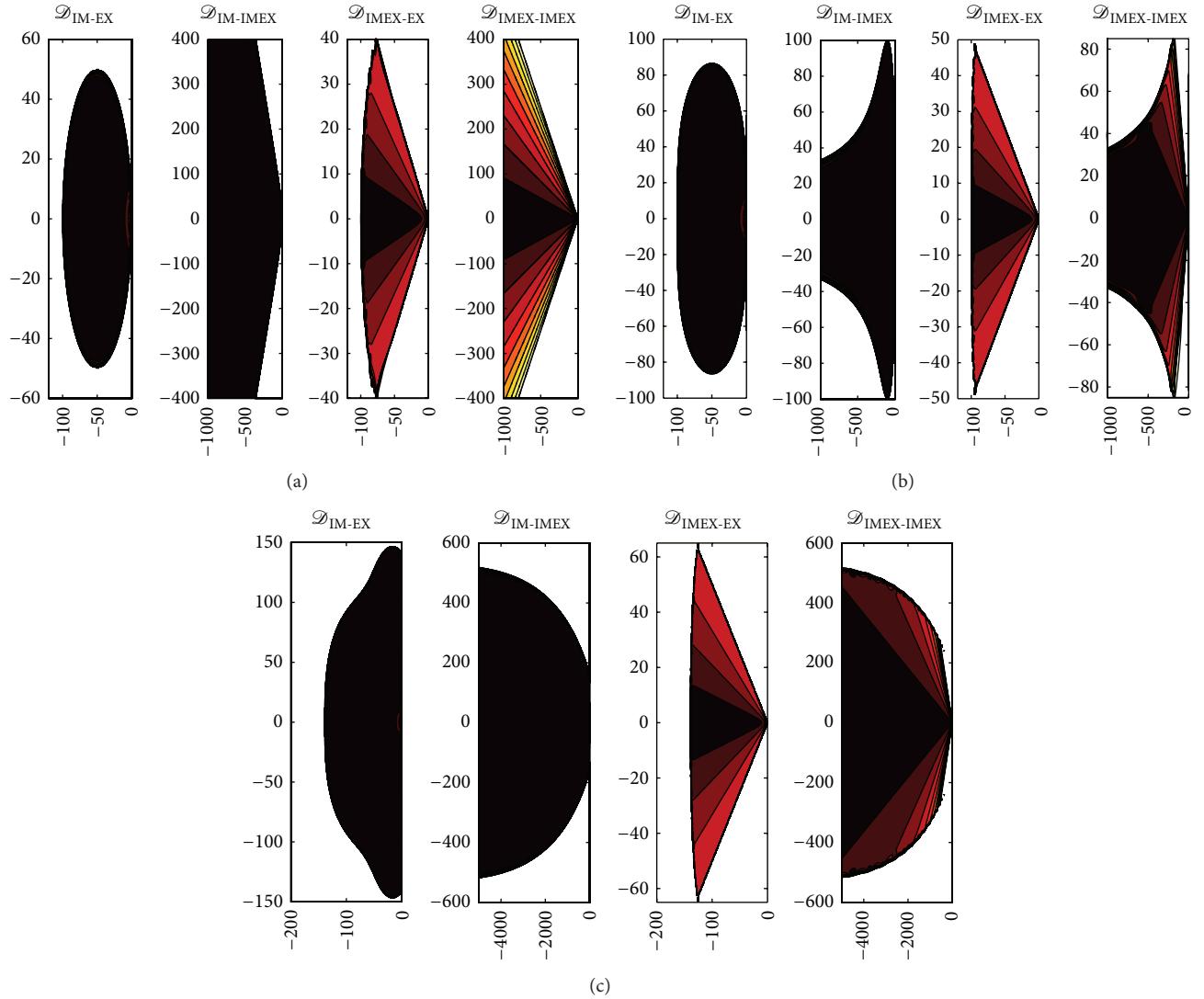


FIGURE 1: (a) $(\text{IMEX}_{\mathcal{G}}, \text{IMEX}_{\mathcal{F}}) = (\text{IMEX Euler}, \text{IMEX Euler})$. (b) $(\text{IMEX}_{\mathcal{G}}, \text{IMEX}_{\mathcal{F}}) = (\text{IMEX Euler}, \text{IMEX Trapezoidal})$. (c) $(\text{IMEX}_{\mathcal{G}}, \text{IMEX}_{\mathcal{F}}) = (\text{IMEX Euler}, \text{IMEX}(4, 4, 3))$.

Figure 1(b):

$$\begin{aligned} & \mathcal{R}_{\text{IM-EX}}(x, y) \\ &= \left| \frac{1}{x + iy - 1} - \left(1 + \frac{x + iy}{M} + \frac{(x + iy)^2}{2M^2} \right)^M \right| \\ & \times \left(1 - \left| \frac{1}{x + iy - 1} \right| \right)^{-1}, \end{aligned}$$

$$\begin{aligned} & \mathcal{R}_{\text{IM-IMEX}}(x, y) \\ &= \left| \frac{1}{x + iy - 1} - \left(1 + \frac{(2 + i(y/M))(x + iy)}{2M - x} \right)^M \right| \\ & \times \left(1 - \left| \frac{1}{x + iy - 1} \right| \right)^{-1}, \end{aligned}$$

$$\begin{aligned} & \mathcal{R}_{\text{IMEX-EX}}(x, y) \\ &= \left| \frac{1 + iy}{1 - x} - \left(1 + \frac{x + iy}{M} + \frac{(x + iy)^2}{2M^2} \right)^M \right| \\ & \times \left(1 - \left| \frac{1 + iy}{1 - x} \right| \right)^{-1}, \\ & \mathcal{R}_{\text{IMEX-IMEX}}(x, y) \\ &= \left| \frac{1 + iy}{1 - x} - \left(1 + \frac{(2 + i(y/M))(x + iy)}{2M - x} \right)^M \right| \\ & \times \left(1 - \left| \frac{1 + iy}{1 - x} \right| \right)^{-1}, \end{aligned} \tag{24}$$

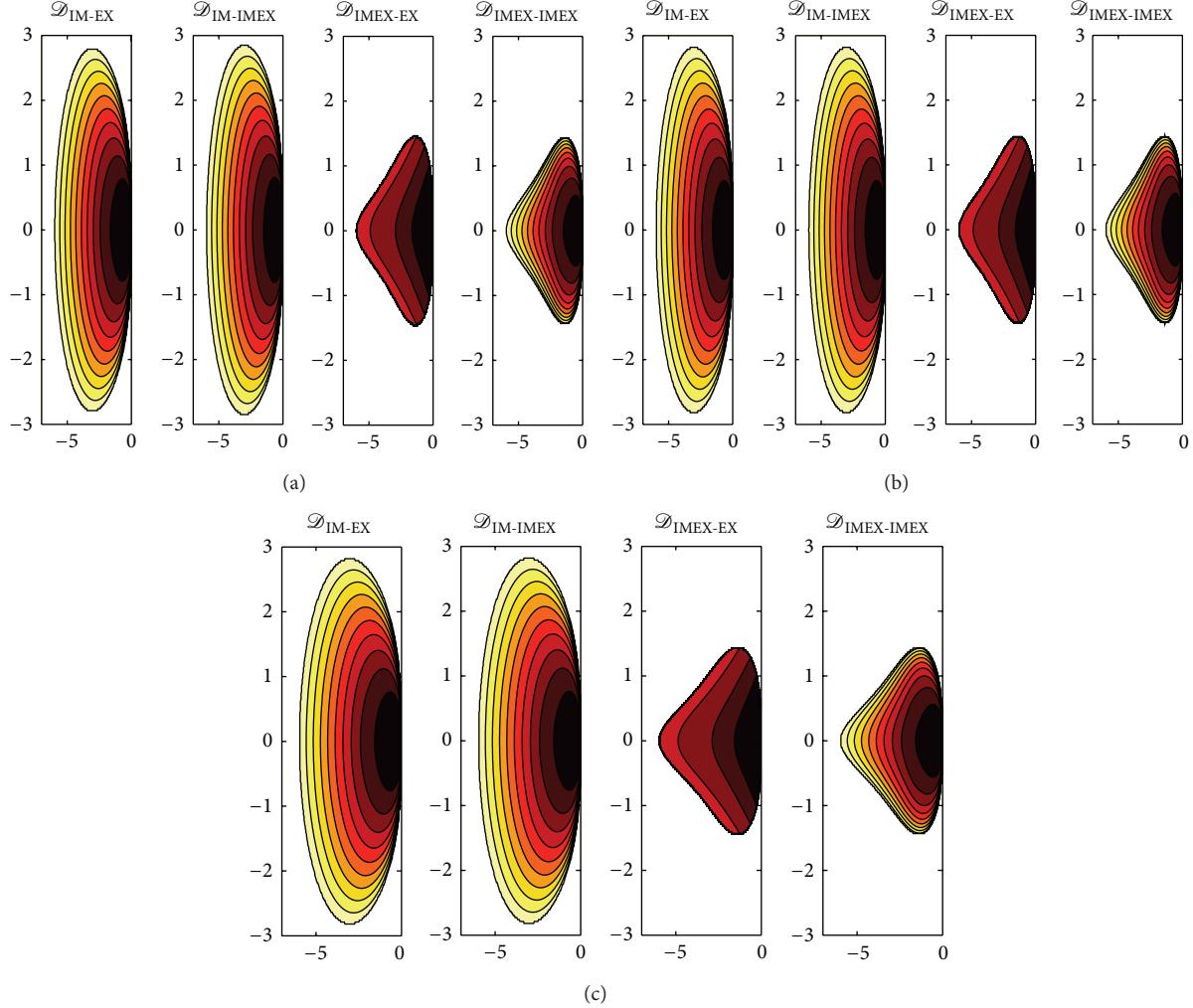


FIGURE 2: (a) $(\text{IMEX}_G, \text{IMEX}_F) = (\text{IMEX Trapezoidal}, \text{IMEX Euler})$. (b) $(\text{IMEX}_G, \text{IMEX}_F) = (\text{IMEX Trapezoidal}, \text{IMEX Trapezoidal})$, $(\text{IMEX}_G, \text{IMEX}_F) = (\text{IMEX Trapezoidal}, \text{IMEX}(4, 4, 3))$.

where $M = 50$. For the case $\text{IMEX}_F = \text{IMEX}(4, 4, 3)$, the stability functions of the four parareal algorithms are very complex and the presentations are omitted.

From Figure 1, we see that the stability region of the $P_{\text{IMEX-IMEX}}$ algorithm seems only a little smaller than that of the $P_{\text{IM-IMEX}}$ algorithm, and both are significantly larger than that of the $P_{\text{IM-EX}}$ algorithm. We note that the computation cost of $P_{\text{IM-IMEX}}$ is obviously more expensive than that of $P_{\text{IMEX-IMEX}}$, since in some cases solving nonlinear equations is not involved in the $P_{\text{IMEX-IMEX}}$ algorithm. Therefore, we get the conclusion that, for $\text{IMEX}_G = \text{IMEX Euler}$, it is better to use the $P_{\text{IMEX-IMEX}}$ algorithm instead of $P_{\text{IM-IMEX}}$ (in the sense of computational cost) and $P_{\text{IM-EX}}$ (in the sense of stability). Moreover, it is clear that the computational cost of the $P_{\text{IMEX-EX}}$ algorithm is the least one among the four algorithms, while from Figure 1 we see that the stability region of this algorithm is the smallest one, and particularly it is smaller than that of the $P_{\text{IM-EX}}$ algorithm. However, from the point of computational cost of view, this algorithm still stands comparison with $P_{\text{IM-EX}}$.

We next consider the case $\text{IMEX}_G = \text{IMEX Trapezoidal}$ scheme (see the left scheme of (9)). For $\text{IMEX}_F = \text{IMEX Euler}$ and $\text{IMEX}_F = \text{IMEX Trapezoidal}$ and $\text{IMEX}_F = \text{IMEX}(4, 4, 3)$, the stability regions of the four algorithms $P_{\text{IM-EX}}$, $P_{\text{IM-IMEX}}$, $P_{\text{IMEX-EX}}$, and $P_{\text{IMEX-IMEX}}$ are plotted in Figures 2(a), 2(b), and 2(c), respectively. For $\text{IMEX}_G = \text{IMEX Trapezoidal}$, the stability functions of the parareal algorithms shown in Figures 2(a) and 2(b) are

Figure 2(a):

$$\mathcal{R}_{\text{IM-EX}}(x, y)$$

$$\begin{aligned}
 &= \left| \frac{2+x+iy}{2-x-iy} - \left(1 + \frac{x+iy}{M} \right)^M \right| \\
 &\times \left(1 - \left| \frac{2+x+iy}{2-x-iy} \right| \right)^{-1},
 \end{aligned}$$

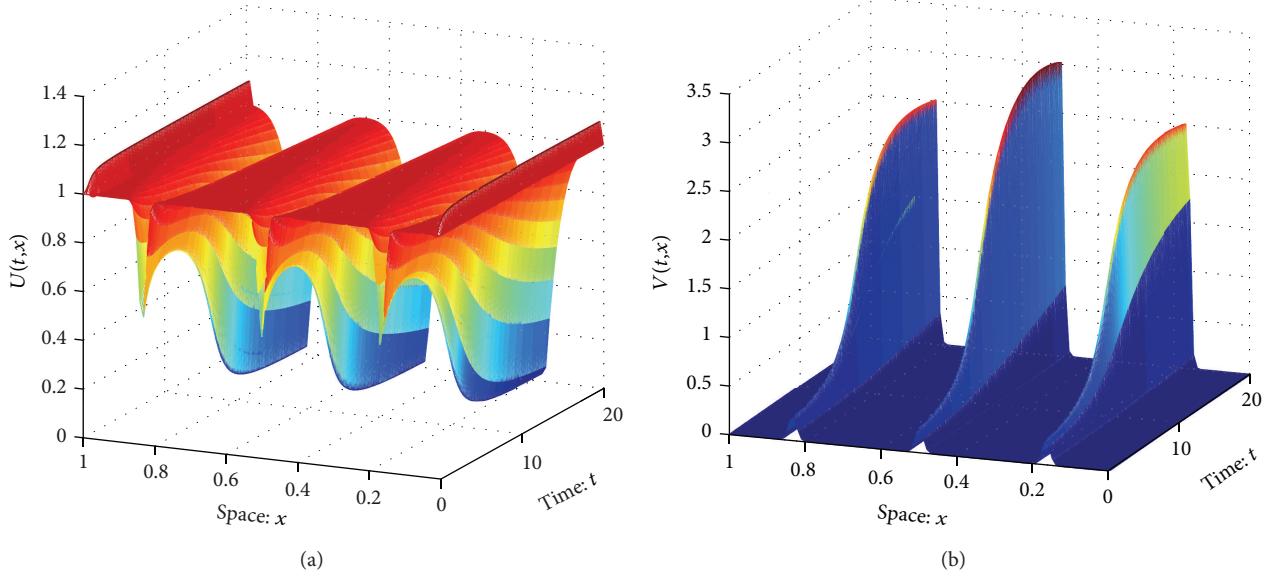


FIGURE 3: Behavior of the solutions of the Gray-Scott model (26)-(27).

$$\mathcal{R}_{\text{IM-IMEX}}(x, y)$$

$$= \left| \frac{2+x+iy}{2-x-iy} - \left(\frac{M+iy}{M-x} \right)^M \right| \\ \times \left(1 - \left| \frac{2+x+iy}{2-x-iy} \right| \right)^{-1},$$

$$\mathcal{R}_{\text{IMEX-EX}}(x, y)$$

$$= \left| 1 + \frac{(2+iy)(x+iy)}{2-x} - \left(1 + \frac{x+iy}{M} \right)^M \right| \\ \times \left(1 - \left| 1 + \frac{(2+iy)(x+iy)}{2-x} \right| \right)^{-1},$$

$$\mathcal{R}_{\text{IMEX-IMEX}}(x, y)$$

$$= \left| 1 + \frac{(2+iy)(x+iy)}{2-x} - \left(\frac{M+iy}{M-x} \right)^M \right| \\ \times \left(1 - \left| 1 + \frac{(2+iy)(x+iy)}{2-x} \right| \right)^{-1},$$

Figure 2(b):

$$\mathcal{R}_{\text{IM-EX}}(x, y)$$

$$= \left| \frac{2+x+iy}{2-x-iy} - \left(1 + \frac{x+iy}{M} + \frac{(x+iy)^2}{2M^2} \right)^M \right| \\ \times \left(1 - \left| \frac{2+x+iy}{2-x-iy} \right| \right)^{-1},$$

$$\mathcal{R}_{\text{IM-IMEX}}(x, y)$$

$$= \left| \frac{2+x+iy}{2-x-iy} - \left(1 + \frac{(2+i(y/M))(x+iy)}{2M-x} \right)^M \right| \\ \times \left(1 - \left| \frac{2+x+iy}{2-x-iy} \right| \right)^{-1},$$

$$\mathcal{R}_{\text{IMEX-EX}}(x, y)$$

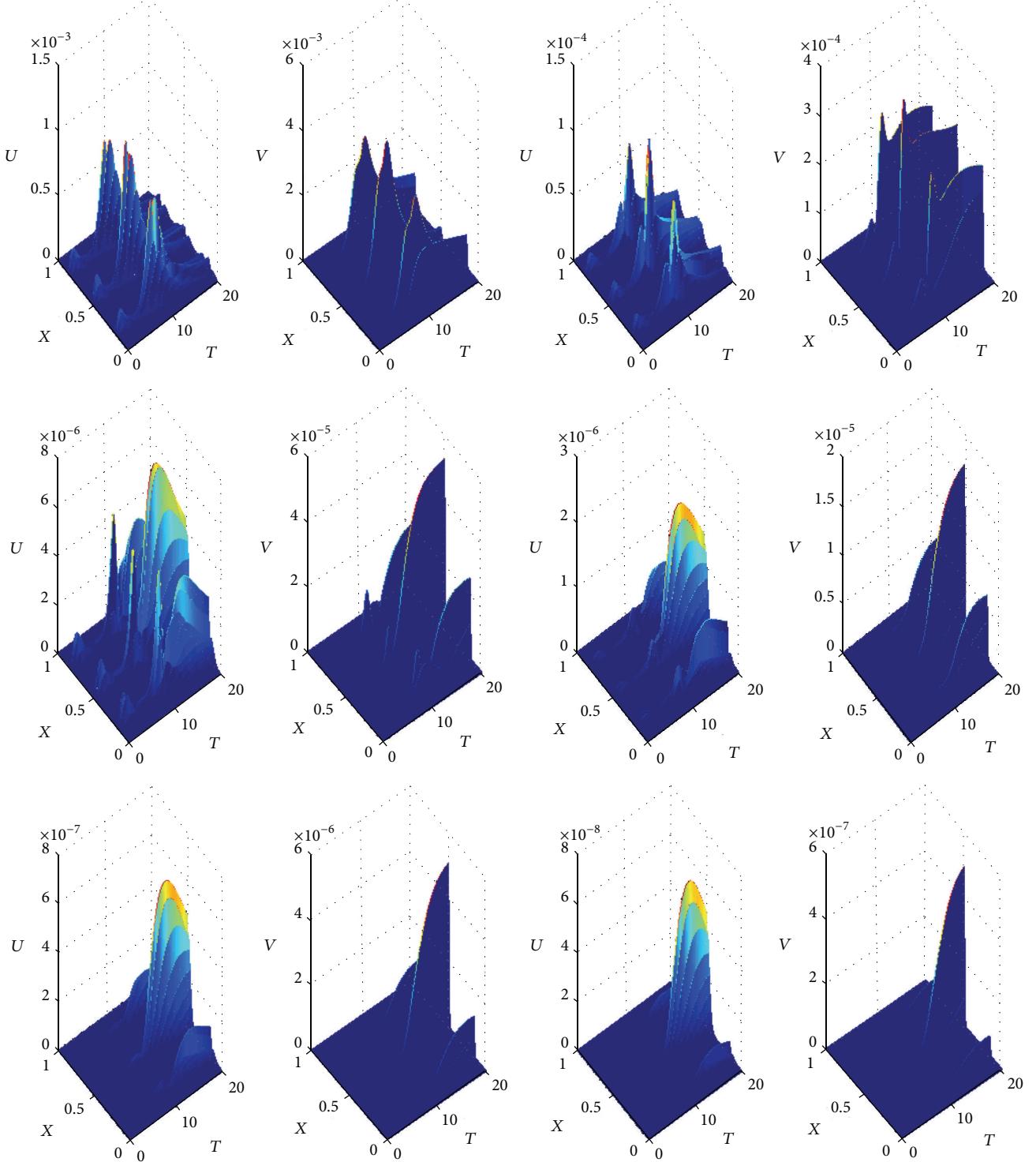
$$= \left| 1 + \frac{(2+iy)(x+iy)}{2-x} - \left(1 + \frac{x+iy}{M} + \frac{(x+iy)^2}{2M^2} \right)^M \right| \\ \times \left(1 - \left| 1 + \frac{(2+iy)(x+iy)}{2-x} \right| \right)^{-1},$$

$$\mathcal{R}_{\text{IMEX-IMEX}}(x, y)$$

$$= \left| 1 + \frac{(2+iy)(x+iy)}{2-x} - \left(1 + \frac{(2+i(y/M))(x+iy)}{2M-x} \right)^M \right| \\ \times \left(1 - \left| 1 + \frac{(2+iy)(x+iy)}{2-x} \right| \right)^{-1},$$

(25)

where $M = 50$. Again, for the case $\text{IMEX}_{\mathcal{F}} = \text{IMEX}(4, 4, 3)$ we omit the presentations of the four stability functions because of high complexity.

FIGURE 4: Measured error diminishing of the solutions u and v when the $\mathbf{P}_{\text{IMEX-EX}}$ algorithm is used.

For $\text{IMEX}_{\mathcal{G}} = \text{IMEX Trapezoidal}$, from Figure 2 we see that the results are different from the previous case $\text{IMEX}_{\mathcal{G}} = \text{IMEX Euler}$ and are more interesting: (a) the stability region of each algorithm shrinks to a smaller region; (b) for each 1 algorithm, the stability regions for different $\text{IMEX}_{\mathcal{G}}$ are very similar; (c) it seems that there is no difference between $\mathbf{P}_{\text{IM-EX}}$ ($\mathbf{P}_{\text{IMEX-EX}}$) and $\mathbf{P}_{\text{IM-IMEX}}$ ($\mathbf{P}_{\text{IMEX-IMEX}}$).

Based on the results shown in Figures 1 and 4, we get the following two conclusions:

- (1) it is more advisable to use IMEX Euler method as the underlying numerical method for the $\mathcal{G}_{\Delta T}$ propagator instead of using the IMEX Trapezoidal method;

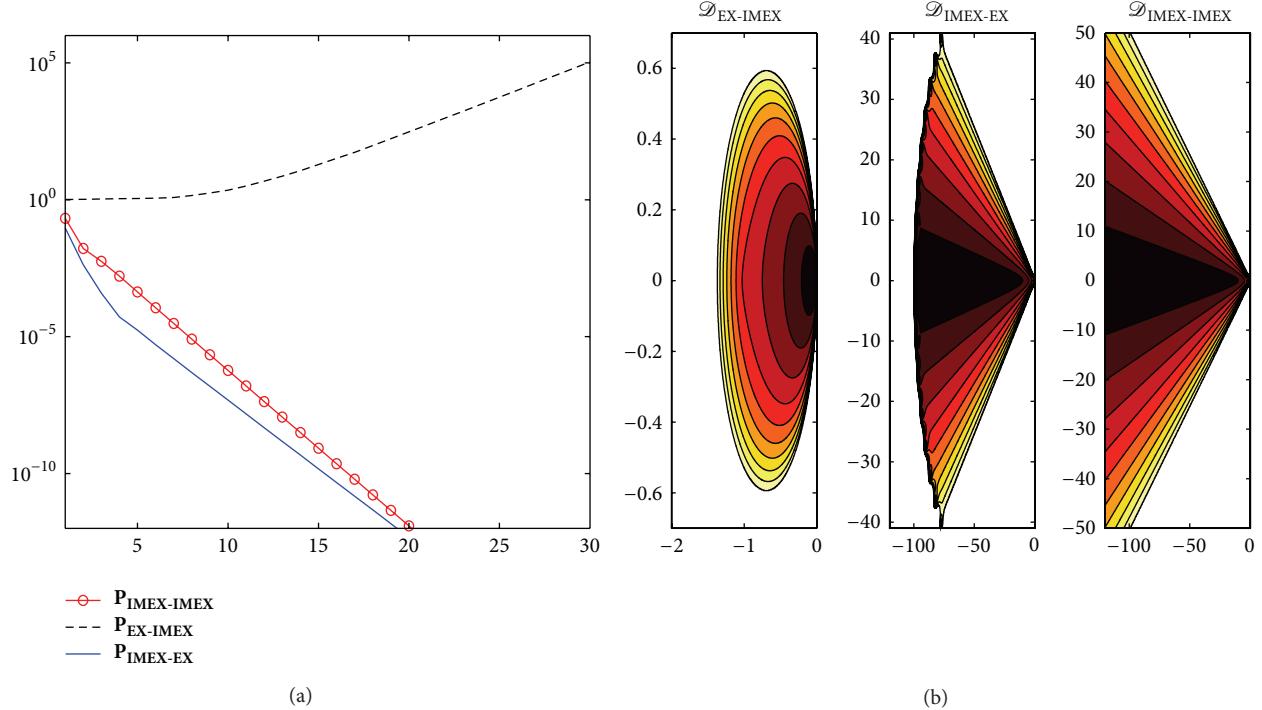


FIGURE 5: Measured error diminishing (a) and stability region (b) of the algorithms $P_{\text{EX-IMEX}}$, $P_{\text{IMEX-EX}}$, and $P_{\text{IMEX-IMEX}}$.

- (2) if $\text{IMEX}_{\mathcal{G}} = \text{IMEX}$ Trapezoidal, it seems that the $P_{\text{IM-EX}}$ algorithm is the best one, since it outperforms $P_{\text{IMEX-EX}}$ and $P_{\text{IMEX-IMEX}}$ in the sense of stability and outperforms $P_{\text{IM-IMEX}}$ in the sense of computational cost.

4. Numerical Results

In this section, we show some numerical results to illustrate the good performance of the parareal algorithm implemented with the IMEX RK method. We test the well-known Gray-Scott model arising from chemical reaction (see [38, 39]):

$$\begin{aligned} u_t &= \epsilon_1 u_{xx} - uv^2 + \eta(1-u), \\ v_t &= \epsilon_2 v_{xx} + uv^2 - (\eta + \theta)v, \end{aligned} \quad (26)$$

where $t \in [0, 20]$, $x \in [0, 1]$ and $\epsilon_1 = 10^{-4}$, $\epsilon_2 = 10^{-6}$, $\eta = 0.024$, and $\theta = 0.06$. The initial-boundary condition for (26) is chosen as

$$\begin{aligned} u(0, x) &= 1 - \frac{1}{2}(\sin(3\pi x))^{100}, \\ v(0, x) &= \frac{1}{4}(\sin(3\pi x))^{100}, \\ u(t, 1) &= u(t, 0) = 1, \\ v(t, 1) &= v(t, 0) = 0. \end{aligned} \quad (27)$$

With these conditions, we expect three pulses in the solutions of (26). Figure 3 shows a typical behaviour of the solutions.

The PDE system (26) is first discretized spatially as $u_{xx} \approx (u_{j+1} - 2u_j + u_{j-1})/\Delta x^2$ and $v_{xx} \approx (v_{j+1} - 2v_j + v_{j-1})/\Delta x^2$ with

$\Delta x = 0.01$, $j = 1, 2, \dots, 100$, and then a nonlinear ordinary differential system that consists of 200 ODEs is solved by the parareal algorithm. We use the IMEX Euler method to define both the $\mathcal{G}_{\Delta T}$ propagator and $\mathcal{F}_{\Delta t}$ propagator. We consider here $M = 50$, $\Delta T = 1/10$, and $\Delta t = \Delta T/M$. In Figure 4, we show the first 6 iterations of the parareal $P_{\text{IMEX-EX}}$ algorithm, where we see that the error diminishing for the solutions u and v is very rapid.

We also test the convergence speed of the parareal $P_{\text{IMEX-EX}}$ algorithm and the parareal algorithm which is denoted by $P_{\text{EX-IMEX}}$ with propagators $\mathcal{G}_{\Delta T}$ and $\mathcal{F}_{\Delta t}$ being defined by the explicit Euler and the IMEX Euler methods, respectively. The convergence curves corresponding to these three parareal algorithms are shown in Figure 5(a). We see that the $P_{\text{EX-IMEX}}$ algorithm is not convergent and both the other two algorithms converge rapidly. This observation can be explained by the stability region shown in Figure 5(b), where we see that the stability region of $P_{\text{EX-IMEX}}$ is significantly smaller than that of the other two algorithms. Moreover, as has been shown in Figure 1, we see that the stability region of the $P_{\text{IMEX-EX}}$ algorithm is nearly contained in the one of $P_{\text{IMEX-IMEX}}$. This means that, for the Gray-Scott model (26)-(27), if the $P_{\text{IMEX-EX}}$ algorithm converges, so does the $P_{\text{IMEX-IMEX}}$ algorithm. Moreover, it is interesting that the $P_{\text{IMEX-EX}}$ algorithm converges a little sharper than $P_{\text{IMEX-IMEX}}$; see Figure 5(a).

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are very grateful to the anonymous referees for the careful reading of a preliminary version of the paper and their valuable suggestions and comments, which really improve the quality of this paper. This work was supported by Science & Technology Bureau of Sichuan Province (2014JQ0035), Non-Destructive Inspection of Sichuan Institutes of Higher Education (2013QZY01), the project sponsored by OATF-UESTC, and the NSF of China (11301058, 11301362, 11371157, and 91130003).

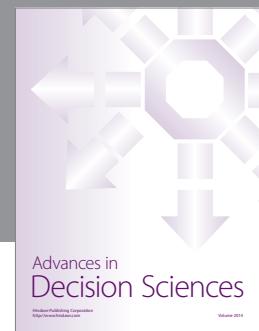
References

- [1] J. Lions, Y. Maday, and G. Turinici, “A “parareal” in time discretization of PDE’s,” *Comptes Rendus de l’Academie des Sciences Series I Mathematics*, vol. 332, no. 1, pp. 661–668, 2001.
- [2] G. Bal and Y. Maday, “A “parareal” time discretization for non-linear PDE’s with application to the pricing of an American put,” in *Recent Developments in Domain Decomposition Methods*, L. F. Pavarino and A. Toselli, Eds., vol. 23 of *Lecture Notes in Computational Science and Engineering*, pp. 189–202, 2002.
- [3] M. J. Gander and M. Petcu, “Analysis of a Krylov subspace enhanced parareal algorithm,” *ESAIM: Proceedings*, vol. 29, no. 2, pp. 556–578, 2007.
- [4] J. Cortial and C. Farhat, “A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems,” *International Journal for Numerical Methods in Engineering*, vol. 77, no. 4, pp. 451–470, 2009.
- [5] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello, “Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses,” *International Journal for Numerical Methods in Engineering*, vol. 67, no. 5, pp. 697–724, 2006.
- [6] G. Bal, “On the convergence and the stability of the parareal algorithm to solve partial differential equations,” in *Domain Decomposition Methods in Science and Engineering*, vol. 40 of *Lecture Notes in Computational Science and Engineering*, pp. 425–432, Springer, Berlin, Germany, 2005.
- [7] M. J. Gander and S. Vandewalle, “Analysis of the parareal time-parallel time-integration method,” *SIAM Journal on Scientific Computing*, vol. 29, no. 2, pp. 556–578, 2007.
- [8] G. A. Staff and E. M. Rønquist, “Stability of the parareal algorithm,” in *Domain Decomposition Methods in Science and Engineering*, vol. 40 of *Lecture Notes in Computational Science and Engineering*, pp. 449–456, Springer, Berlin, Germany, 2005.
- [9] S. Wu, B. Shi, and C. Huang, “Parareal-Richardson algorithm for solving nonlinear ODEs and PDEs,” *Communications in Computational Physics*, vol. 6, no. 4, pp. 883–902, 2009.
- [10] M. Emmett and M. L. Minion, “Toward an efficient parallel in time method for partial differential equations,” *Communications in Applied Mathematics and Computational Science*, vol. 7, no. 1, pp. 105–132, 2012.
- [11] M. L. Minion, “A hybrid parareal spectral deferred corrections method,” *Communications in Applied Mathematics and Computational Science*, vol. 5, no. 2, pp. 265–301, 2010.
- [12] X. Dai and Y. Maday, “Stable parareal in time method for first- and second-order hyperbolic systems,” *SIAM Journal on Scientific Computing*, vol. 35, no. 1, pp. A52–A78, 2013.
- [13] L. P. He and M. X. He, “Parareal in time simulation of morphological transformation in cubic alloys with spatially dependent composition,” *Communications in Computational Physics*, vol. 11, no. 5, pp. 1697–1717, 2012.
- [14] C. Farhat and M. Chandesris, “Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications,” *International Journal for Numerical Methods in Engineering*, vol. 58, no. 9, pp. 1397–1434, 2003.
- [15] Y. Maday, J. Salomon, and G. Turinici, “Monotonic parareal control for quantum systems,” *SIAM Journal on Numerical Analysis*, vol. 45, no. 6, pp. 2468–2482, 2007.
- [16] T. P. Mathew, M. Sarkis, and C. E. Schaerer, “Analysis of block parareal preconditioners for parabolic optimal control problems,” *SIAM Journal on Scientific Computing*, vol. 32, no. 3, pp. 1180–1200, 2010.
- [17] X. Dai, C. le Bris, F. Legoll, and Y. Maday, “Symmetric parareal algorithms for Hamiltonian systems,” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 47, no. 3, pp. 717–742, 2013.
- [18] M. J. Gander and E. Hairer, “Analysis for parareal algorithms applied to Hamiltonian differential equations,” *Journal of Computational and Applied Mathematics*, vol. 259, no. part A, pp. 2–13, 2014.
- [19] J. M. Reynolds-Barredo, D. E. Newman, R. Sanchez, D. Samadar, L. A. Berry, and W. R. Elwasif, “Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations,” *Journal of Computational Physics*, vol. 231, no. 23, pp. 7851–7867, 2012.
- [20] J. M. Reynolds-Barredo, D. E. Newman, and R. Sanchez, “An analytic model for the convergence of turbulent simulations time-parallelized via the parareal algorithm,” *Journal of Computational Physics*, vol. 255, pp. 293–315, 2013.
- [21] X. Li, T. Tang, and C. Xu, “Parallel in time algorithm with spectral-subdomain enhancement for Volterra integral equations,” *SIAM Journal on Numerical Analysis*, vol. 51, no. 3, pp. 1735–1756, 2013.
- [22] D. Guibert and D. Tromeur-Dervout, “Parallel adaptive time domain decomposition for stiff systems of ODEs/DAEs,” *Computers & Structures*, vol. 85, no. 9, pp. 553–562, 2007.
- [23] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, “Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations,” *Applied Numerical Mathematics. An IMACS Journal*, vol. 25, no. 2-3, pp. 151–167, 1997.
- [24] S. Sekar, “Analysis of linear and nonlinear stiff problems using the RK-Butcher algorithm,” *Mathematical Problems in Engineering*, vol. 2006, Article ID 39246, 15 pages, 2006.
- [25] U. M. Ascher, S. J. Ruuth, and B. T. Wetton, “Implicit-explicit methods for time-dependent partial differential equations,” *SIAM Journal on Numerical Analysis*, vol. 32, no. 3, pp. 797–823, 1995.
- [26] G. J. Cooper and A. Sayfy, “Additive Runge-Kutta methods for stiff ordinary differential equations,” *Mathematics of Computation*, vol. 40, no. 161, pp. 207–218, 1983.
- [27] M. Mechee, N. Senu, F. Ismail, B. Nikouravan, and Z. Siri, “A three-stage fifth-order Runge-Kutta method for directly solving special third-order differential equation with application to thin film flow problem,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 795397, 7 pages, 2013.
- [28] T. Koto, “IMEX Runge-Kutta schemes for reaction-diffusion equations,” *Journal of Computational and Applied Mathematics*, vol. 215, no. 1, pp. 182–195, 2008.

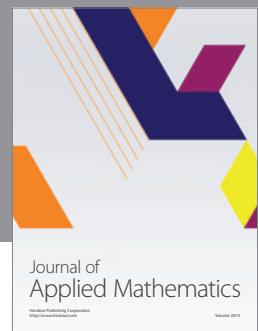
- [29] Q. Ming, Y. Yang, and Y. Fang, "An optimized Runge-Kutta method for the numerical solution of the radial Schrödinger equation," *Mathematical Problems in Engineering*, vol. 2012, Article ID 867948, 12 pages, 2012.
- [30] H. Liu and J. Zou, "Some new additive Runge-Kutta methods and their applications," *Journal of Computational and Applied Mathematics*, vol. 190, no. 1-2, pp. 74–98, 2006.
- [31] X. Duan, J. Leng, C. Cattani, and C. Li, "A Shannon-Runge-Kutta-Gill method for convection-diffusion equations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 163734, 5 pages, 2013.
- [32] L. Pareschi and G. Russo, "Implicit-Explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation," *Journal of Scientific Computing*, vol. 25, no. 1-2, pp. 129–155, 2005.
- [33] H. Yuan, C. Song, and P. Wang, "Nonlinear stability and convergence of two-step Runge-Kutta methods for neutral delay differential equations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 683137, 14 pages, 2013.
- [34] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer, Berlin, Germany, 2003.
- [35] J. Frank, W. Hundsdorfer, and J. G. Verwer, "On the stability of implicit-explicit linear multistep methods," *Applied Numerical Mathematics*, vol. 25, no. 2-3, pp. 193–205, 1997.
- [36] J. G. Verwer and B. P. Sommeijer, "An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations," *SIAM Journal on Scientific Computing*, vol. 25, no. 5, pp. 1824–1835, 2004.
- [37] T. Koto, "Stability of IMEX Runge-Kutta methods for delay differential equations," *Journal of Computational and Applied Mathematics*, vol. 211, no. 2, pp. 201–212, 2008.
- [38] J. E. Pearson, "Complex patterns in a simple system," *Science*, vol. 261, no. 5118, pp. 189–192, 1993.
- [39] K. J. Lee, W. D. McCormick, Q. Ouyang, and H. L. Swinney, "Pattern formation by interacting chemical fronts," *Science*, vol. 261, no. 5118, pp. 192–194, 1993.



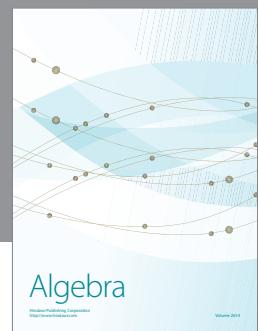
Advances in
Operations Research



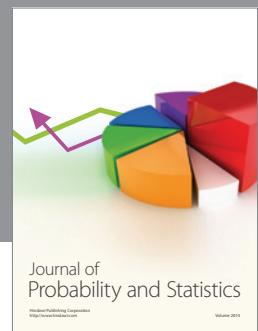
Advances in
Decision Sciences



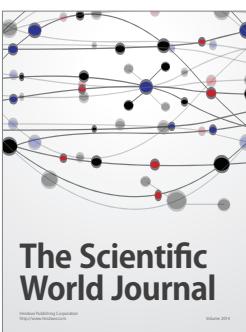
Journal of
Applied Mathematics



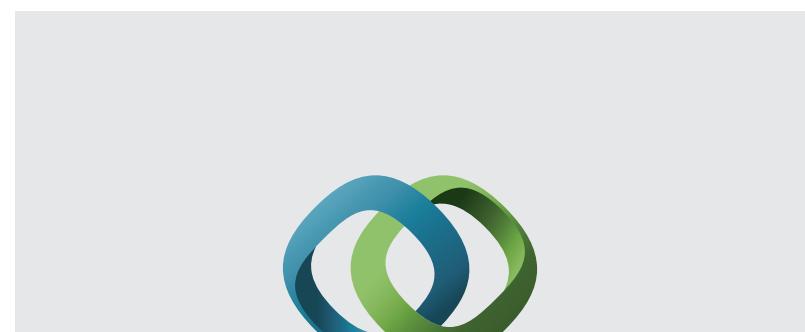
Algebra



Journal of
Probability and Statistics

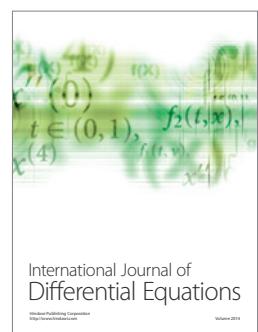


The Scientific
World Journal



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>



International Journal of
Differential Equations



International Journal of
Combinatorics



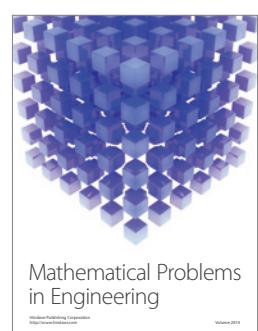
Advances in
Mathematical Physics



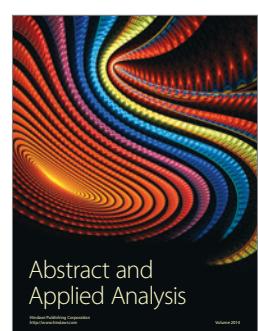
Journal of
Complex Analysis



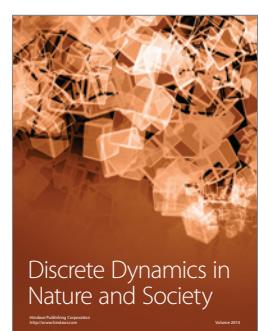
Journal of
Mathematics



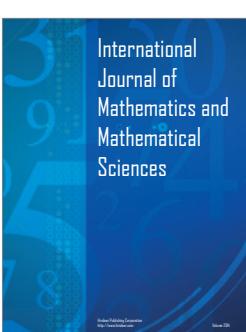
Mathematical Problems
in Engineering



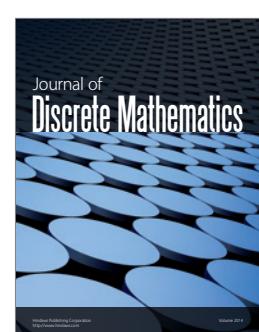
Abstract and
Applied Analysis



Discrete Dynamics in
Nature and Society



International
Journal of
Mathematics and
Mathematical
Sciences



Journal of
Discrete Mathematics



Journal of
Function Spaces



International Journal of
Stochastic Analysis



Journal of
Optimization