

2016

A Hybrid Algorithm Based on Optimal Quadratic Spline Collocation and Parareal Deferred Correction for Parabolic PDEs

Jun Liu

Yan Wang

Rongjian Li
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/computerscience_fac_pubs



Part of the [Computer Sciences Commons](#)

Repository Citation

Liu, Jun; Wang, Yan; and Li, Rongjian, "A Hybrid Algorithm Based on Optimal Quadratic Spline Collocation and Parareal Deferred Correction for Parabolic PDEs" (2016). *Computer Science Faculty Publications*. 51.
https://digitalcommons.odu.edu/computerscience_fac_pubs/51

Original Publication Citation

Liu, J., Wang, Y., & Li, R.J. (2016). A hybrid algorithm based on optimal quadratic spline collocation and parareal deferred correction for parabolic PDEs. *Mathematical Problems in Engineering*, 1-15. doi: 10.1155/2016/6943079

Research Article

A Hybrid Algorithm Based on Optimal Quadratic Spline Collocation and Parareal Deferred Correction for Parabolic PDEs

Jun Liu,¹ Yan Wang,¹ and Rongjian Li²

¹College of Science, China University of Petroleum (East China), Qingdao, Shandong 266580, China

²Department of Computer Science, Old Dominion University, Norfolk, VA 23508, USA

Correspondence should be addressed to Jun Liu; liujun@upc.edu.cn

Received 14 December 2015; Accepted 5 April 2016

Academic Editor: Yann Favennec

Copyright © 2016 Jun Liu et al. 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 is a kind of time parallel numerical methods for time-dependent systems. In this paper, we consider a general linear parabolic PDE, use optimal quadratic spline collocation (QSC) method for the space discretization, and proceed with the parareal technique on the time domain. Meanwhile, deferred correction technique is also used to improve the accuracy during the iterations. In fact, the optimal QSC method is a correction of general QSC method. Along the temporal direction we embed the iterations of deferred correction into parareal to construct a hybrid method, parareal deferred correction (PDC) method. The error estimation is presented and the stability is analyzed. To save computational cost, we find out a simple way to balance the two kinds of iterations as much as possible. We also argue that the hybrid algorithm has better system efficiency and costs less running time. Numerical experiments by multicore computers are attached to exhibit the effectiveness of the hybrid algorithm.

1. Introduction

The parareal algorithm was firstly proposed by Lions et al. [1] in 2001 as a parallel-in-time approach for solving time-dependent differential equations, mainly for ordinary differential equations (ODEs) and parabolic equations [2–4]. The significant advantage of the parareal algorithm is that it does not require the consecutive solving of subproblems along the temporal direction. As a purely parallel algorithm, the parareal algorithm has been applied to many practical problems, such as hyperbolic problems [5], fluid mechanics [6], quantum control [7, 8], and optimized control problem [9, 10]. Recently, some innovative variants of the parareal were proposed by integrating with other efficient methods, such as spectral deferred corrections [11], domain decomposition technique [12], the Krylov subspace method [13], and adaptive technique [14]. We have also proposed a kind of parareal waveform relaxation methods for ODEs [15] and parabolic PDEs [16], as well as parareal deferred correction methods for PDEs [17].

The parareal algorithm was initially derived from multiple shooting, time multigrid, and algebraic deferred

correction [18, 19] through borrowing the idea of domain decomposition on the time domain. In particular, it employs two operators on two time decompositions of different step sizes with the name of coarse and fine grids to propagate the values along the time direction. The initial guesses of subproblems defined on the coarse grid are given by the coarse propagator for the first iteration. Then, the values at the interface of two adjacent coarse-grid subproblems are updated using fine propagators in iterations, which is quite similar to the Schwarz methods for elliptic systems. Spectral deferred correction method, a kind of iterative methods for solving ODEs, is originally proposed in [20] and has been combined with parareal in [11]. It makes use of the previously computed solution on the same time subdomain. High order approximations can be achieved while the computation cost is reduced sharply.

In this paper, we apply the idea of correction to the problem of parabolic PDEs. In detail, we employ the optimal QSC method for the spatial variables and PDC method for the temporal variables, which will result in high accuracy approximations with low computational cost. The new method can

also be implemented in parallel, just like the same parallel mechanism as the classical parareal method.

The outline of our paper is as follows. In Section 2, we give a brief review about parareal algorithm and deferred correction. In Section 3, we employ the optimal QSC method for the space discretization of parabolic PDE, which results in an ODEs system. Then we present the PDC method and apply the method to the ODEs system. In Section 4, we give a proper error estimation for the PDC method. In Section 5, we analyze the stability of the algorithm and show the relationship between the stability and the propagators employed. In Section 6, we present a simple way to choose the time step of the fine propagator. The speedup and system efficiency of the new algorithm are analyzed in Section 7. Numerical experiments are attached in Section 8.

2. Materials and Methods

2.1. Parareal Algorithm. We briefly review the parareal algorithm on the following nonlinear ODEs:

$$\begin{aligned} \frac{du(t)}{dt} &= f(u(t), t), \quad t \in [0, T], \\ u(0) &= u_0, \end{aligned} \quad (1)$$

where the nonlinear function $f: \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^n$ is Lipschitz on \mathbf{R}^n and the function $u: \mathbf{R} \rightarrow \mathbf{R}^n$ is to be computed.

The parareal algorithm requires a decomposition of the time domain $[0, T]$ into N time subdomains $[T_n, T_{n+1}]$, $n = 0, 1, \dots, N-1$, with $0 = T_0 < T_1 < \dots < T_{N-1} < T_N = T$. For simplicity, we suppose that all time subdomains have the same length $\Delta T := T_{n+1} - T_n$. The approximations at these points $\{T_n \mid n = 1, 2, \dots, N\}$ can be obtained using the following scheme:

$$\begin{aligned} U_0^{k+1} &= u_0, \\ U_{n+1}^{k+1} &= G(T_{n+1}, T_n, U_n^{k+1}) + F(T_{n+1}, T_n, U_n^k) \\ &\quad - G(T_{n+1}, T_n, U_n^k), \end{aligned} \quad (2)$$

where $G(T_{n+1}, T_n, U_n)$ and $F(T_{n+1}, T_n, U_n)$ denote, respectively, the coarse and fine approximation at the point T_{n+1} with initial value U_n at the point T_n . The coarse propagator G is cheap and lowly accurate, and the fine propagator F is of high precision but also more expensive. The initialization of the iterative algorithm (2) is usually realized by the propagation of the coarse propagator along $\{T_n \mid n = 1, 2, \dots, N\}$ consecutively; that is, $U_{n+1}^0 = G(T_{n+1}, T_n, U_n^0)$.

The parareal algorithm has several well-known properties in the following: (a) the initial values for the fine propagator on each time subdomain are known in every iteration, so the fine propagators on different time subdomains can be implemented by different processors. (b) The final result of parareal algorithm will achieve the accuracy of serial implementation of the fine propagator on the whole time domain after N iterations, where N is the number of time subdomains. Note that the number of iterations is usually set to be much less

than the number of time subdomains in practice to avoid necessary computations, while achieving a competitive accuracy [13].

2.2. Deferred Correction Method. Deferred correction method is an iterative method for ODEs. We employ the scheme designed in [21] as one of the basic elements for our proposed method, in spite of many other variants of the original deferred corrections. We adopt the scheme in [21] because of its simplicity for implementation with high accuracy and also robustness for different problems.

We also take system (1) as our example to describe the execution details of deferred correction method. We firstly define a grid on $[0, T]$ with $(M+1)$ equispaced nodes, where $0 = t_0 < t_1 < \dots < t_M = T$, with time step $h = T/M$. An r th order method on this grid could bring a numerical solution $\{U_0, U_1, \dots, U_M\}$ with error $U_i = u(t_i) + \mathcal{O}(h^r)$. In addition, we can also generate a unique M th-degree polynomial $\tilde{u}(t)$ on $[0, T]$ with this grid using Lagrange interpolation. Then we can define an error function

$$\delta(t) = u(t) - \tilde{u}(t), \quad t \in [0, T], \quad (3)$$

which satisfies the differential equation

$$\begin{aligned} \frac{d\delta(t)}{dt} &= f(\delta(t) + \tilde{u}(t), t) - \frac{d\tilde{u}(t)}{dt}, \quad t \in [0, T], \\ \delta(0) &= 0. \end{aligned} \quad (4)$$

This equation is usually regarded as the neighbor problem of original system (1) in the regime of deferred correction methods. Meanwhile, if we use the same r th numerical method to solve this neighbor problem, we could get another group of numerical values $\{\Delta_0, \Delta_1, \dots, \Delta_M\}$ on the same grid with error $\Delta_i = \delta(t_i) + \mathcal{O}(h^r)$. The numerical solution $\{U_i + \Delta_i, i = 0, 1, \dots, M\}$ is a better approximation of the original system after one correction since we have $U_i + \Delta_i = u(t_i) + \mathcal{O}(h^{2r})$ [22–24]. If we iterate such correction J times, the order of accuracy of the iterated deferred correction is $\mathcal{O}(h^{\min[(J+1)r, M]})$. In practice, the degree M can not be very large.

3. QSC-PDC Methods for Parabolic PDEs

In this study, we consider the following linear PDE as our basic model to demonstrate our proposed method:

$$\begin{aligned} \frac{\partial}{\partial t} u(x, t) &= \frac{\partial^2}{\partial x^2} u(x, t) + f(x, t), \\ x &\in \Omega, \quad t \in [0, T], \end{aligned} \quad (5)$$

$$u(x, t) = 0, \quad x \in \partial\Omega, \quad t \in [0, T],$$

$$u(x, 0) = g(x), \quad x \in \Omega,$$

where $\Omega \equiv [a, b]$ is the spatial domain, $[0, T]$ is the time domain with $T > 0$, and $u(x, t)$ is the unknown function.

We take the time domain decomposition for system (5) and design the generic iteration scheme:

$$\begin{aligned} U_0^{k+1}(x) &= g(x), \\ U_{n+1}^{k+1}(x) &= G(T_{n+1}, T_n, U_n^{k+1}(x)) \\ &\quad + F(T_{n+1}, T_n, U_n^k(x)) \\ &\quad - G(T_{n+1}, T_n, U_n^k(x)). \end{aligned} \quad (6)$$

The essential efforts of this paper are concentrated on the design of techniques, which could bring execution cost and high accuracy for scheme (6).

3.1. QSC Methods for Spatial Variables. We firstly consider reducing the computational cost in spatial domain. Note that, for every iteration k , we have to use the propagators G and F to solve a linear homogeneous PDE defined on a fixed spatial domain the same as that of the original system for every time subdomain. The discretization of the spatial and temporal variables will result in algebraic equations for subproblems. The approximate order of the discretization procedure is crucial to the scale of the algebraic equation. In this paper, we use the optimal QSC method for the spatial variable, which was originally used for two-point boundary value problem [25], then elliptic PDEs [26], and parabolic PDEs [27, 28].

Similarly, we consider a uniform partition of the spatial domain, $\{a = x_0 < x_1 < \dots < x_I = b\}$ with mesh size $\Delta x = 1/I$. We denote $\mathcal{P}_{2,\Delta}$ as the space of piecewise quadratic polynomials and $\mathcal{S}_{2,\Delta} = \mathcal{P}_{2,\Delta} \cap \mathcal{E}^1([a, b])$ as the space of quadratic splines. Remember that the dimension of the space $\mathcal{S}_{2,\Delta}$ is $I + 2$. We let $\{\phi_0, \phi_1, \dots, \phi_{I+1}\}$ be a set of piecewise polynomial basis functions of the space $\mathcal{S}_{2,\Delta}$.

For a fixed t , system (5) is a two-point boundary value problem, whose quadratic spline approximate solution can be written as

$$p(x) = \sum_{i=0}^{I+1} c_i \phi_i(x). \quad (7)$$

Therefore, we can write the approximation of system (5) as

$$u_\Delta(x, t) = \sum_{i=0}^{I+1} c_i(t) \phi_i(x), \quad (8)$$

where $c_i(t)$, $i = 0, 1, \dots, I$, are degrees of freedom (DOFs). Taking the set $\{\tau_0 = a, \tau_i = (x_{i-1} + x_i)/2; 1 \leq i \leq I, \tau_{I+1} = b\}$ as the collocation points, we can obtain ODEs systems with respect to the DOFs.

Starting from the quadratic polynomial

$$\phi(x) = \frac{1}{2} \begin{cases} x^2, & 0 \leq x \leq 1, \\ -2(x-1)^2 + 2(x-1) + 1, & 1 \leq x \leq 2, \\ (3-x)^2, & 2 \leq x \leq 3, \\ 0, & \text{elsewhere,} \end{cases} \quad (9)$$

we choose a set of quadratic B-splines

$$\phi_i(x) = \phi\left(\frac{x-a}{\Delta x} - i + 2\right), \quad i = 0, 1, \dots, I + 1 \quad (10)$$

as the basis functions of the space $\mathcal{S}_{2,\Delta}$. Following [27], we denote $\widetilde{\mathcal{S}}_{2,\Delta}$ as the space of quadratic splines satisfying homogeneous Dirichlet boundary conditions. The dimension of the space $\widetilde{\mathcal{S}}_{2,\Delta}$ is I , and a set of basis functions of such space is $\{\tilde{\phi}_1 = \phi_1 - \phi_0, \tilde{\phi}_i = \phi_i, i = 2, \dots, I-1, \tilde{\phi}_I = \phi_I - \phi_{I+1}\}$. Then the quadratic spline approximate solution of system (5) can be written as

$$u_\Delta(x, t) = \sum_{i=1}^I c_i(t) \tilde{\phi}_i(x). \quad (11)$$

With the values of the basis functions on collocation points $\{\tau_1, \dots, \tau_I\}$, we can obtain the following ODEs systems on the DOFs $c_i(t)$ of the optimal one-step QSC methods:

$$Q_0 \frac{dc(t)}{dt} = \frac{1}{\Delta x^2} \left(Q_2 + \frac{1}{24} Q_{xx} \right) c(t) + \vec{f}(t), \quad t \in [0, T], \quad (12)$$

$$c(0) = c_0,$$

where $c(t) = [c_1(t), \dots, c_I(t)]^T$, the coefficient matrices are

$$\begin{aligned} Q_0 &= \frac{1}{8} \begin{pmatrix} 5 & 1 & 0 & \dots & 0 \\ 1 & 6 & 1 & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \dots & 1 & 6 & 1 \\ 0 & \dots & 0 & 1 & 5 \end{pmatrix}, \\ Q_2 &= \begin{pmatrix} -3 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \dots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -3 \end{pmatrix}, \\ Q_{xx} &= \begin{pmatrix} -11 & 16 & -14 & 6 & -1 & 0 & \dots & 0 \\ -5 & 6 & -4 & 1 & 0 & 0 & \dots & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots & 0 \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & \dots & 0 & 0 & 1 & -4 & 6 & -5 \\ 0 & \dots & 0 & -1 & 6 & -14 & 16 & -11 \end{pmatrix}, \end{aligned} \quad (13)$$

and $\vec{f}(t)$ is the interpolating vector of $f(x, t)$ at all the collocation points. The vector $c_0 \in R^I$ satisfies $Q_0 c_0 = g_0$, with g_0 the interpolation of $g(x)$ at collocation points.

The discretization error of the spatial variable in the uniform norm is $\mathcal{O}(\Delta x^3)$ globally and $\mathcal{O}(\Delta x^4)$ at the nodes

of the uniform partition and the collocation points [25]. In contrast, the classical finite difference discretization could only generate approximations with error $\mathcal{O}(\Delta x^2)$ locally, which demonstrates the great benefits of the optimal QSC.

3.2. PDC Method for the Temporal Variables. The parareal algorithm has a very important property. The approximation at all the endpoints of time subdomains will achieve the accuracy of the F -propagator after N iterations [8, 18], where N is the number of time subdomains. In fact, for any $n < N$, the approximation at the time point T_n will be updated n times and reach the final approximation which is exactly the result by the F -propagator at this time point.

In order to carry out the PDC method on ODEs system (12) with respect to DOFs, we discretize each time interval $I_n = [T_n, T_{n+1}]$ into smaller equispaced grids $T_n = t_0 < t_1 < \dots < t_m < \dots < t_M = T_{n+1}$ and denote C_n^k the PDC approximation at the point T_n in k th parareal iteration. For simplicity, we only arrange one correction in each parareal iteration. The details of the hybrid algorithm are described as follows step by step:

- (i) Use the coarse propagator G to obtain rough numerical approximations at all the time interval points T_n , that is, C_n^0 , $n = 0, \dots, N$.
- (ii) Use the prediction step of the new propagator F_{DC} on each time interval I_n , usually an (r_0) th-order numerical method, to obtain fine approximations at the grid points on I_n with the initial value C_n^0 at T_n . We suppose that the numerical solution is $\vec{\eta}_n^{[1]} = (\eta_{n,0}^{[1]}, \eta_{n,1}^{[1]}, \dots, \eta_{n,m}^{[1]}, \dots, \eta_{n,M}^{[1]})$, and $F_{\text{DC}}(T_{n+1}, T_n, C_n^0) = \eta_{n,M}^{[1]}$, where $\eta_{n,0}^{[1]} = C_n^0$.
- (iii) Use the inductive formulae of parareal algorithm to obtain improved approximations C_n^1 at T_n , $n = 0, \dots, N$; that is,

$$\begin{aligned} C_0^1 &= c_0, \\ C_{n+1}^1 &= G(T_{n+1}, T_n, C_n^1) + F_{\text{DC}}(T_{n+1}, T_n, C_n^0) \\ &\quad - G(T_{n+1}, T_n, C_n^0). \end{aligned} \quad (14)$$

- (iv) For $k = 2, \dots, k_0$, (k_0 is the number of parareal iterations)

- (1) Solve the following neighbor problem of the original system:

$$\begin{aligned} Q_0 \frac{de_n^{(k-1)}(t)}{dt} &= Q_0 \frac{dc_n(t)}{dt} - Q_0 \frac{d\eta_n^{(k-1)}(t)}{dt} \\ &= \frac{1}{\Delta x^2} \left(Q_2 + \frac{1}{24} Q_{xx} \right) \left(e_n^{(k-1)}(t) + \eta_n^{(k-1)}(t) \right) \\ &\quad + \vec{f}(t) - Q_0 \frac{d\eta_n^{(k-1)}(t)}{dt}, \quad t \in I_n, \end{aligned} \quad (15)$$

$$e_n^{(k-1)}(T_n) = C_0^k - C_0^{k-1},$$

where $e_n^{(k-1)}(t) = c_n(t) - \eta_n^{(k-1)}(t)$ is the error function on each time interval I_n from the previous step, $c_n(t)$ is the true solution of (12) restricted on I_n , and $\eta_n^{(k-1)}(t)$ is an M th-order polynomial interpolating $\vec{\eta}_n^{[k-1]}$. We denote the numerical solution of (15) as $\vec{\delta}_n^{[k-1]} = (\delta_{n,0}^{[k-1]}, \delta_{n,1}^{[k-1]}, \dots, \delta_{n,m}^{[k-1]}, \dots, \delta_{n,M}^{[k-1]})$.

- (2) Update the numerical solution $\vec{\eta}_n^{[k]} = \vec{\eta}_n^{[k-1]} + \vec{\delta}_n^{[k-1]}$, and let $F_{\text{DC}}(T_{n+1}, T_n, C_n^{k-1}) = \eta_{n,M}^{[k]}$.
- (3) Perform the following iterative formulae to get improved approximations:

$$\begin{aligned} C_{n+1}^k &= G(T_{n+1}, T_n, C_n^k) + F_{\text{DC}}(T_{n+1}, T_n, C_n^{k-1}) \\ &\quad - G(T_{n+1}, T_n, C_n^{k-1}), \end{aligned} \quad (16)$$

where $n = 0, 1, \dots, N-1$.

With these final DOFs $C_n^{k_0}$, we can obtain the approximations

$$[U_0^{k_0}, U_1^{k_0}, \dots, U_N^{k_0}] = Q_0 [C_0^{k_0}, C_1^{k_0}, \dots, C_N^{k_0}] \quad (17)$$

of original system (5) on the grid consisting in spatial collocation points and temporal points.

Remark 1. When solving neighbor problem (15), we can use some simple numerical integrators, such as the backward Euler method. An alternative formulation is to compute the approximate solution in two steps, and the order of accuracy $\mathcal{O}(\Delta x^4)$ is also preserved [25, 27, 28]. The realization process of such strategy based on backward Euler method is as follows:

- (i) first compute a second order approximation $\{\xi_{n,0}^{[k-1]}, \dots, \xi_{n,m}^{[k-1]}, \dots, \xi_{n,M+1}^{[k-1]}\}$ by
$$\begin{aligned} &\left(Q_0 - \frac{\Delta t}{\Delta x^2} Q_2 \right) \xi_{n,m+1}^{[k-1]} \\ &= Q_0 \xi_{n,m}^{[k-1]} + \frac{\Delta t}{\Delta x^2} Q_2 \eta_{n,m+1}^{[k-1]} + \Delta t \vec{f}(t_{n,m+1}) \\ &\quad - \Delta t Q_0 \frac{d}{dt} \eta_n^{(k-1)}(t_{n,m+1}), \end{aligned} \quad (18)$$

- (ii) then compute an optimal order approximation $\{\delta_{n,0}^{[k-1]}, \dots, \delta_{n,m}^{[k-1]}, \dots, \delta_{n,M+1}^{[k-1]}\}$ by

$$\begin{aligned} &\left(Q_0 - \frac{\Delta t}{\Delta x^2} Q_2 \right) \delta_{n,m+1}^{[k-1]} \\ &= Q_0 \delta_{n,m}^{[k-1]} + \frac{\Delta t}{\Delta x^2} \frac{1}{24} Q_{xx} \xi_{n,m+1}^{[k-1]} \\ &\quad + \frac{\Delta t}{\Delta x^2} \left(Q_2 + \frac{1}{24} Q_{xx} \right) \eta_{n,m+1}^{[k-1]} + \Delta t \vec{f}(t_{n,m+1}) \\ &\quad - \Delta t Q_0 \frac{d}{dt} \eta_n^{(k-1)}(t_{n,m+1}). \end{aligned} \quad (19)$$

The two-step version involves two linear systems over every time step, and both the two coefficient matrices of the above two systems are tridiagonal. So the total computational cost will be smaller than that of one-step integrator (15), in which the bandwidth of the coefficient matrix is 4, because of the matrix Q_{xx} .

4. The Convergence of the PDC Method

The convergence analysis of the optimal QSC method can be found in the literature [25], and we just need to present the convergence of the PDC algorithm in the temporal domain. Bal [2], Gander, and Hairer [29] gave a convergence analysis on the classical parareal algorithm. Based on their work, we present the convergence analysis on the PDC method.

We also assume that the coarse propagator G satisfies error expansion

$$\begin{aligned} S(T_{n+1}, T_n, U) - G(T_{n+1}, T_n, U) \\ = c_{p+1}(U) \Delta T^{p+1} + c_{p+2}(U) \Delta T^{p+2} + \dots, \end{aligned} \quad (20)$$

for ΔT small and the Lipschitz condition

$$\begin{aligned} \|G(t + \Delta T, t, U) - G(t + \Delta T, t, V)\| \\ \leq (1 + C_2 \Delta T) \|U - V\|, \end{aligned} \quad (21)$$

where $S(T_{n+1}, T_n, U)$ means the true solution at point T_{n+1} of the original system with the value U at the time point T_n .

Within every time subinterval $[T_n, T_{n+1}]$, the difference between the true solution and the approximation by the PDC method can be written as

$$\begin{aligned} u(T_{n+1}) - U_{n+1}^{k+1} &= S(T_{n+1}, T_n, u(T_n)) \\ &- G(T_{n+1}, T_n, U_n^{k+1}) - F_{DC}(T_{n+1}, T_n, U_n^k) \\ &+ G(T_{n+1}, T_n, U_n^k) = [S(T_{n+1}, T_n, u(T_n)) \\ &- G(T_{n+1}, T_n, u(T_n)) + G(T_{n+1}, T_n, U_n^k) \\ &- S(T_{n+1}, T_n, U_n^k)] + [S(T_{n+1}, T_n, U_n^k) \\ &- F_{DC}(T_{n+1}, T_n, U_n^k)] + [G(T_{n+1}, T_n, u(T_n)) \\ &- G(T_{n+1}, T_n, U_n^{k+1})]. \end{aligned} \quad (22)$$

Based on this relationship and also the conditions (20) and (21), we could get

$$\begin{aligned} \|u(T_{n+1}) - U_{n+1}^{k+1}\| \\ \leq C_1 \Delta T^{p+1} \|u(T_n) - U_n^k\| \\ + (1 + C_2 \Delta T) \|u(T_n) - U_n^{k+1}\| \\ + \|S(T_{n+1}, T_n, U_n^k) - F_{DC}(T_{n+1}, T_n, U_n^k)\|. \end{aligned} \quad (23)$$

We first look at the last term $\|S(T_{n+1}, T_n, U_n^k) - F_{DC}(T_{n+1}, T_n, U_n^k)\|$, which is an upper bound of the error

caused by the deferred correction. For simplicity, we assume the solution of the original system to be smooth enough and solve the original equation in the prediction steps and the neighbor equations in the correction steps by the forward Euler integrators.

The prediction step in deferred correction includes a numerical integration with initial value U_n^0 on every time interval I_n . Let $\vec{\eta}_n^{[1]} = (\eta_{n,0}^{[1]}, \eta_{n,1}^{[1]}, \dots, \eta_{n,M}^{[1]})$ be the numerical solution from the forward Euler integrators on $(M + 1)$ uniformly distributed nodes in I_n with time step h . We know from [30] that the error satisfies $\|u_n(Mh) - \eta_{n,M}^{[1]}\|_\infty \sim \mathcal{O}(h)$, where $u_n(Mh)$ is the true solution at T_{n+1} with the same initial value U_n^0 at T_n . In fact, the error at T_{n+1} is related to the length of I_n , so it is reasonable to hypothesize $\|u_n(Mh) - \eta_{n,M}^{[1]}\|_\infty \leq C \Delta T h$, where C is a constant.

The correction steps in the hybrid algorithm are different from these steps of the pure deferred correction. The initial values of neighbor problems are always 0 for pure deferred correction. Meanwhile, for the correction steps which are embedded in parareal process, the initial values of the neighbor problems are kept being updated in parareal iterations. In other words, the initial value of problem (15) on time subinterval I_n is $U_0^k - U_0^{k-1}$. However, this formula of the initial value does not affect the iterative improvement property of the deferred correction. For the k th parareal iteration, we have

$$\begin{aligned} u(t_{n,m}) - \eta_{n,m}^{[k]} &= u(t_{n,m}) - \eta_{n,m}^{[k-1]} - \delta_{n,m}^{[k-1]} \\ &= e_n^{(k-1)}(t_{n,m}) - \delta_{n,m}^{[k-1]}, \end{aligned} \quad (24)$$

where $t_{n,m} = (n\Delta T + mh)$. We can see from (24) that the error of deferred correction method equals the error of the corresponding neighbor problem. Similar to the inductive convergence proof of spectral deferred correction in [30], we can obtain that $\|S(T_{n+1}, T_n, U_n^k) - F_{DC}(T_{n+1}, T_n, U_n^k)\| \sim \mathcal{O}(h^{k+1})$, considering the solution of the original system to be smooth enough.

Based on these results, we have the convergence of the PDC method.

Theorem 2. *Let the solution of the original system be smooth enough. If the coarse propagator G satisfies condition (20) and condition (21), where c_j , $j = p + 1, p + 2, \dots$, are continuously differentiable, then the error at time point T_n after k PDC iterations is bounded:*

$$\begin{aligned} \|u(T_n) - U_n^k\| \\ \leq \frac{C_3 (C_1 T_n \Delta T^p)^{k+1}}{C_1 (k+1)!} e^{C_2(T_n - T_{k+1})} \\ + \sum_{i=0}^{k-1} \frac{C_1 \Delta T^{p_i} (T_n)^i}{(i+1)!} T_n e^{C_2(T_n - T_{i+1})} C_T h^{k-i}. \end{aligned} \quad (25)$$

Obviously, when the time step h approaches 0, such error bound will go to 0 with enough iterations.

Proof. We suppose that a constant C_T exists such that $\|S(T_{n+1}, T_n, U_n^k) - F_{DC}(T_{n+1}, T_n, U_n^k)\| \leq C_T \Delta T h^{k+1}$ holds for any $n = 0, 1, \dots, N-1$. Inequality (23) can be bounded by

$$\begin{aligned} \|u(T_{n+1}) - U_{n+1}^{k+1}\| &\leq C_1 \Delta T^{p+1} \|u(T_n) - U_n^k\| \\ &\quad + (1 + C_2 \Delta T) \|u(T_n) - U_n^{k+1}\| \\ &\quad + C_T \Delta T h^{k+1}. \end{aligned} \quad (26)$$

Considering such inequality, we study the recurrence relation

$$\begin{aligned} e_{n+1}^{k+1} &= \alpha e_n^k + \beta e_n^{k+1} + C_T \Delta T h^{k+1}, \\ e_{n+1}^0 &= \gamma + \beta e_n^0, \end{aligned} \quad (27)$$

where $\alpha = C_1 \Delta T^{p+1}$, $\beta = 1 + C_2 \Delta T$, and $\gamma = C_3 \Delta T^{p+1}$. It is easy to see that e_{n+1}^{k+1} is an upper bound on $\|u(T_{n+1}) - U_{n+1}^{k+1}\|$, and we can choose $e_0^k = 0$, $k = 0, 1, \dots$. Multiplying (27) by ζ^{n+1} and summing over n , we obtain

$$\begin{aligned} \sum_{n \geq 0} e_{n+1}^{k+1} \zeta^{n+1} &= \sum_{n \geq 0} \alpha e_n^k \zeta^{n+1} + \sum_{n \geq 0} \beta e_n^{k+1} \zeta^{n+1} \\ &\quad + \sum_{n \geq 0} C_T \Delta T h^{k+1} \zeta^{n+1}, \\ \sum_{n \geq 0} e_{n+1}^0 \zeta^{n+1} &= \sum_{n \geq 0} \gamma \zeta^{n+1} + \sum_{n \geq 0} \beta e_n^0 \zeta^{n+1}, \end{aligned} \quad (28)$$

where $-1 < \zeta < 1/\beta < 1$.

We denote $\rho_k(\zeta) := \sum_{n \geq 0} e_{n+1}^k \zeta^{n+1}$. This function satisfies the following recurrence relation:

$$\begin{aligned} \rho_{k+1}(\zeta) &= \alpha \zeta \rho_k(\zeta) + \beta \zeta \rho_{k+1}(\zeta) + C_T \Delta T h^{k+1} \frac{\zeta}{1-\zeta}, \\ \rho_0(\zeta) &= \gamma \frac{\zeta}{1-\zeta} + \beta \zeta \rho_0(\zeta). \end{aligned} \quad (29)$$

Then, we have

$$\begin{aligned} \rho_k(\zeta) &= \frac{\alpha \zeta}{1-\beta \zeta} \rho_{k-1}(\zeta) + \frac{C_T \Delta T h^k \zeta}{(1-\zeta)(1-\beta \zeta)}, \\ \rho_0(\zeta) &= \frac{\gamma \zeta}{(1-\zeta)(1-\beta \zeta)}. \end{aligned} \quad (30)$$

Solving for $\rho_k(\zeta)$, we obtain after induction

$$\begin{aligned} \rho_k(\zeta) &= \left(\frac{\alpha \zeta}{1-\beta \zeta} \right)^k \rho_0(\zeta) \\ &\quad + \frac{C_T \Delta T \zeta}{(1-\zeta)(1-\beta \zeta)} (h^k + ah^{k-1} + \dots + a^{k-1}h) \\ &= \frac{\gamma \alpha^k \zeta^{k+1}}{(1-\beta \zeta)^{k+1} (1-\zeta)} \\ &\quad + \frac{C_T \Delta T \zeta}{(1-\zeta)(1-\beta \zeta)} (h^k + ah^{k-1} + \dots + a^{k-1}h), \end{aligned} \quad (31)$$

where $a = \alpha \zeta / (1-\beta \zeta)$. We know that replacing the factor $1-\zeta$ in the denominator by $1-\beta \zeta$ only will increase the coefficients in the power series of $\rho_k(\zeta)$. It means the coefficients in the power series of $\rho_k(\zeta)$ are smaller than the corresponding coefficients in the power series of $\tilde{\rho}_k(\zeta)$, where

$$\begin{aligned} \tilde{\rho}_k(\zeta) &= \frac{\gamma \alpha^k \zeta^{k+1}}{(1-\beta \zeta)^{k+2}} \\ &\quad + \frac{C_T \Delta T \zeta}{(1-\beta \zeta)^2} (h^k + ah^{k-1} + \dots + a^{k-1}h). \end{aligned} \quad (32)$$

Using the binomial series expansion

$$\frac{1}{(1-\beta \zeta)^{k+2}} = \sum_{j \geq 0} \binom{k+1+j}{j} \beta^j \zeta^j, \quad (33)$$

it is easy to find that the coefficient of the term ζ^n in the expression $\tilde{\rho}_k(\zeta)$ is

$$\begin{aligned} c_n^k &= \gamma \alpha^k \binom{n}{k+1} \beta^{n-k-1} \\ &\quad + \sum_{i=0}^{k-1} \binom{n}{i+1} \alpha^i \beta^{n-i-1} C_T \Delta T h^{k-i} \\ &= C_3 \Delta T^{p+1} C_1^k (\Delta T^{p+1})^k \frac{n!}{(n-k-1)!(k+1)!} (1 \\ &\quad + C_2 \Delta T)^{n-k-1} \\ &\quad + \sum_{i=0}^{k-1} \frac{n!}{(n-i-1)!(i+1)!} C_1^i (\Delta T^{p+1})^i \\ &\quad \cdot (1 + C_2 \Delta T)^{n-i-1} C_T \Delta T h^{k-i}. \end{aligned} \quad (34)$$

For any $x > 0$, we have $1+x \leq e^x$. So, it follows

$$\begin{aligned} c_n^k &\leq \frac{C_3}{C_1} \frac{(C_1 \Delta T^{p+1})^{k+1}}{(k+1)!} (1 + C_2 \Delta T)^{n-k-1} \prod_{j=0}^k (n-j) \\ &\quad + \sum_{i=0}^{k-1} \frac{C_1^i (\Delta T^{p+1})^i}{(i+1)!} (1 + C_2 \Delta T)^{n-i-1} C_T \Delta T h^{k-i} \\ &\quad \cdot \prod_{j=0}^i (n-j) \leq \frac{C_3}{C_1} \frac{(C_1 T_n \Delta T^p)^{k+1}}{(k+1)!} e^{C_2(T_n - T_{k+1})} \\ &\quad + \sum_{i=0}^{k-1} \frac{C_1^i \Delta T^{p_i} (T_n)^i}{(i+1)!} T_n e^{C_2(T_n - T_{i+1})} C_T h^{k-i}. \end{aligned} \quad (35)$$

□

Remark 3. In practice, we usually take ΔT very small and h much smaller. We notice that the first term in (25) decreases to 0 superlinearly as k increases, and such decrease is very quick with the help of small ΔT . For another hand, the increase of k also takes the summation in (25) much smaller despite more terms. Therefore, a couple of iterations are enough to pull the error down below some tolerance.

5. Stability Analysis of the PDC Method

To describe the stability of the PDC method, we apply such numerical method to the equation

$$\begin{aligned} \frac{d\varphi(t)}{dt} &= \lambda\varphi(t), \quad t \in [0, 1], \\ \varphi(0) &= 1. \end{aligned} \quad (36)$$

We follow the definition of stability in [20] and define the amplification factor $\text{Am}(\lambda\Delta T)$ for $\lambda \in \mathbb{C}$ as

$$\text{Am}(\lambda\Delta T) = \tilde{\varphi}(1), \quad (37)$$

where $\tilde{\varphi}(1)$ denotes the numerical approximation at the point $t = 1$. The numerical method is stable for some given value of λ , if

$$|\text{Am}(\lambda\Delta T)| \leq 1. \quad (38)$$

For simplicity, we adopt forward and backward Euler methods in the analysis. The time domain $[0, 1]$ is divided into N subdomains. The coarse propagator is forced to take large time steps, and implicit Euler is better than explicit Euler for the coarse propagator [31]. Therefore, we choose one-step backward Euler method to be the coarse propagator on every subdomain, while the fine propagator is chosen as s steps of Euler method on every subdomain for the original equations and corresponding neighbor problems in the process of deferred correction.

We notice that the approximation $\tilde{\varphi}(1)$ is related to the iteration number k of the PDC method. We denote by Φ_n^k the approximation of system (36) at n th subdomain point and on the k th iteration, where $n = 0, 1, \dots, N$ and $k = 0, 1, \dots$. The approximations satisfy

$$\begin{aligned} \Phi_{n+1}^{k+1} &= G(T_{n+1}, T_n, \Phi_n^{k+1}) + F(T_{n+1}, T_n, \Phi_n^k) \\ &\quad - G(T_{n+1}, T_n, \Phi_n^k). \end{aligned} \quad (39)$$

Because of the one-step backward Euler method for the coarse propagator, we have

$$G(T_{n+1}, T_n, \Phi_n^k) = \frac{1}{1 - \lambda\Delta T} \Phi_n^k. \quad (40)$$

We denote $R = 1/(1 - \lambda\Delta T)$ as the amplification factor of the coarse propagator and let r be the amplification factor of the fine propagator, which is from the deferred correction method. In fact, the first iteration of the PDC method is exactly the classical deferred correction, but with smaller numerical steps on each time subdomain. Therefore, $r = (1/(1 - \lambda\Delta T/s))^s$ for s steps of backward Euler integrators, and $r = (1 + \lambda\Delta T/s)^s$ for s steps of forward Euler integrators. However, the expressions of r will be much more complicated when $k \geq 1$.

For the initial approximation, we have

$$\Phi_n^0 = R^n \varphi(0) = R^n, \quad n = 0, 1, \dots, N. \quad (41)$$

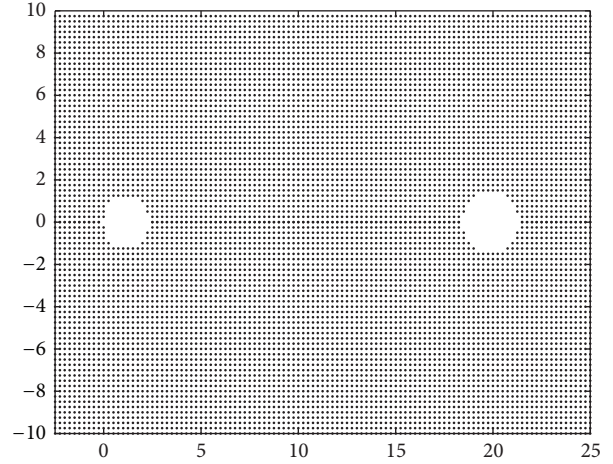


FIGURE 1: Instability region (blank area) using implicit Euler integrators for both the coarse and fine propagators. The x -axis is $\text{Re}(\lambda\Delta T)$ and the y -axis is $\text{Im}(\lambda\Delta T)$, where $N = 20$ and $s = 20$.

For the first iteration ($k = 0$), we have

$$\begin{aligned} \Phi_{n+1}^1 &= R\Phi_n^1 + r\Phi_n^0 - R\Phi_n^0 = R\Phi_n^1 + (r - R)\Phi_n^0, \\ n &= 0, 1, \dots, N - 1, \end{aligned} \quad (42)$$

and, after inductions, we get

$$\Phi_N^1 = R^{N-1} (Nr - (N-1)R), \quad (43)$$

which is also the amplification factor after one iteration of the PDC method. We will continue to analyze the stability in two different cases, respectively.

Case 1. For the s steps of implicit Euler integrators as the fine propagators

$$\begin{aligned} \text{Am}_{\text{ImIm}}(\lambda\Delta T, k = 0) &= \left| \frac{1}{(1 - \lambda\Delta T)^{N-1}} \left(\frac{N}{(1 - \lambda\Delta T/s)^s} - \frac{N-1}{1 - \lambda\Delta T} \right) \right|. \end{aligned} \quad (44)$$

We denote $z = \lambda\Delta T \in \mathbb{C}$, and the factor $\text{Am}_{\text{ImIm}}(\lambda\Delta T)$ is always smaller than 1 except two areas around its two poles $z = 1$ and $z = s$. If we choose $N = 20$ and $s = 20$, the stability regions are shown in Figure 1.

When $k \geq 1$, we can not get the explicit expressions of the amplification factors. The stability regions are estimated numerically in Figure 2. We see that the size of the instable region expands as the iteration number is increased. Nevertheless, these instable regions always lie in the right half plane, so all the PDC methods with implicit fine propagators at these tested iterations are A -stable.

Additionally, we notice that

$$\lim_{\text{Re}(z) \rightarrow \infty} \text{Am}_{\text{ImIm}}(z, 0) = 0, \quad (45)$$

and the PDC method with both implicit propagators is L -stable at the first iteration. We take several values for λ along

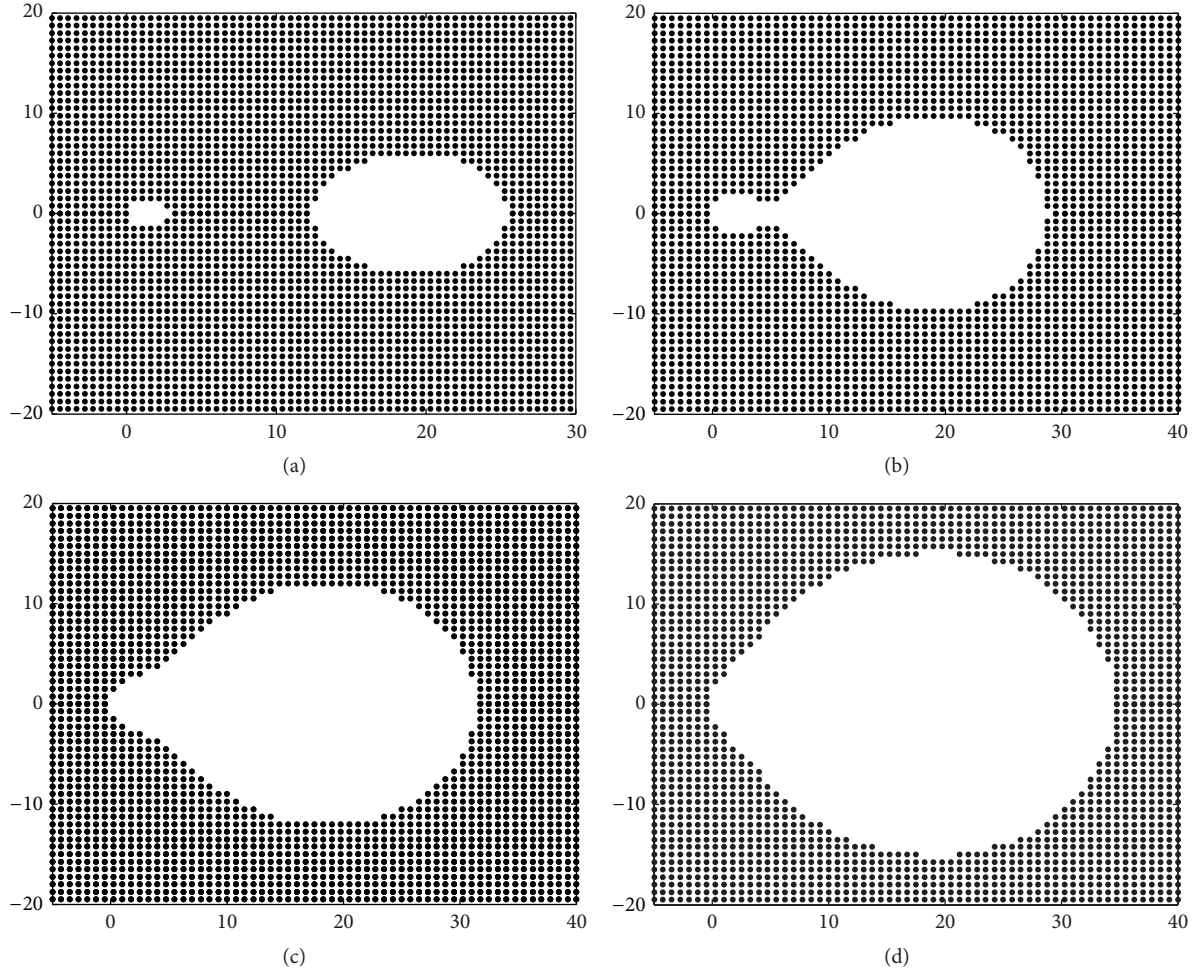


FIGURE 2: Instability regions (blank area) of the PDC method using implicit Euler integrators for both the coarse and fine propagators after 2 iterations (a), 3 iterations (b), 4 iterations (c), and 6 iterations (d). The x -axis is $\text{Re}(\lambda\Delta T)$ and the y -axis is $\text{Im}(\lambda\Delta T)$, where $N = 20$ and $s = 20$.

TABLE 1: Amplification factors at different values of λ .

λ	-10	-50	-100	-200	-400	-800	-1600
$k = 1$	$1.68e - 04$	$9.41e - 10$	$4.00e - 14$	$2.52e - 19$	$6.15e - 25$	$9.49e - 31$	$1.16e - 36$
$k = 2$	$1.26e - 05$	$3.39e - 09$	$2.09e - 13$	$1.42e - 18$	$3.48e - 24$	$5.38e - 30$	$6.56e - 36$
$k = 3$	$4.97e - 05$	$8.46e - 09$	$7.62e - 13$	$5.66e - 18$	$1.39e - 23$	$2.15e - 29$	$2.62e - 35$
$k = 4$	$4.55e - 05$	$1.55e - 08$	$2.07e - 12$	$1.69e - 17$	$4.18e - 23$	$6.45e - 29$	$7.87e - 35$

x -axis, and the corresponding amplification factors are in Table 1.

From Table 1, we can conjecture that the PDC methods at different iterations could also be L -stable.

Case 2. For the s steps of explicit Euler integrators as the fine propagators, the amplification factor is

$$\text{Am}_{\text{ImEx}}(\lambda\Delta T, k=0) = \left| \frac{1}{(1-\lambda\Delta T)^{N-1}} \left(N \left(1 + \frac{\lambda\Delta T}{s} \right)^s - \frac{N-1}{1-\lambda\Delta T} \right) \right|. \quad (46)$$

The regions of $z = \lambda\Delta T$ to make $\text{Am}_{\text{ImEx}}(\lambda\Delta T, k=0) \leq 1$ are related to the relationship between s and N .

If $s > N - 1$, the factor $\text{Am}_{\text{ImEx}}(z, 0)$ has two poles, that is, $z = 1$ and infinity. We choose $N = 20$ and $s = 20$, and the stability regions of the PDC method at different iterations are shown in Figure 3, in which the stable areas are shrunken as the iteration proceeds.

If $s < N - 1$, the factor $\text{Am}_{\text{ImEx}}(z, 0)$ has one pole, that is, $z = 1$. We choose $N = 20$ and $s = 16$, the stability regions of the PDC method at different iterations are shown in Figure 4. In fact, there are very small unstable areas located at the left half plane in Figures 4(a) and 4(b), so the PDC methods are not A -stable in such situations either.

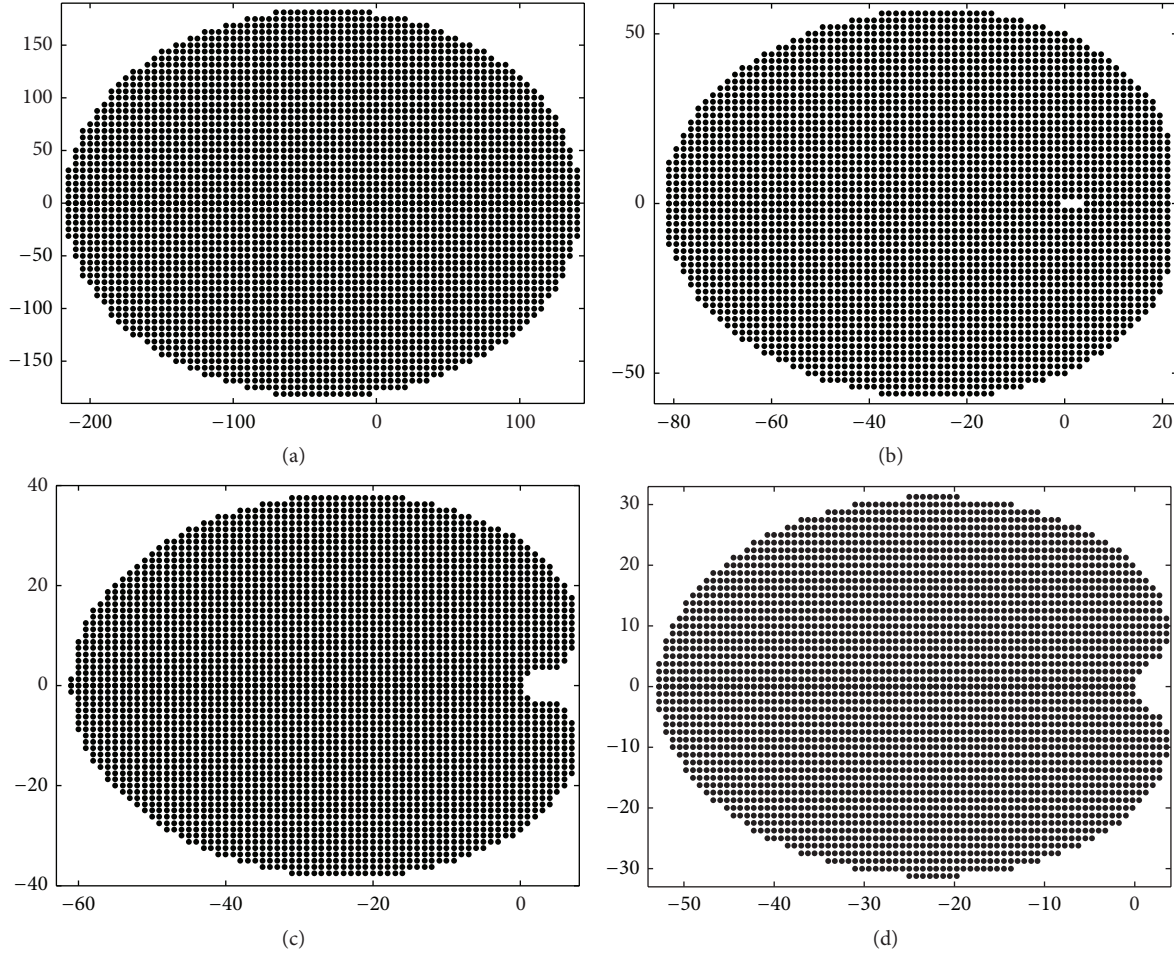


FIGURE 3: Stability region (dotted area) using implicit Euler integrators for the coarse propagators and explicit Euler integrators for the fine propagators after 2 iterations (a), 3 iterations (b), 4 iterations (c), and 6 iterations (d). In fact, there exists a relatively very small unstable area around the point $z = 1$ in (a). The x -axis is $\text{Re}(\lambda\Delta T)$ and the y -axis is $\text{Im}(\lambda\Delta T)$, where $N = 20$ and $s = 20$.

If $s = N - 1$, the factor $\text{Am}_{\text{ImEx}}(z|_{|z|=\infty}, 0)$ is always a positive constant much smaller than 1 because the number of time steps s is usually a moderate integer. Therefore, we are reasonable in merging this case into the previous one.

In Figure 4, we observe that there is a significant change about the pattern of the size of the stable region at the third iteration. The stability region before the third iteration is quite similar to the stability region of the backward Euler method, while the stability region after the third iteration has a similar shape of the stability region of the forward Euler method. The change means that the dominant control is transferred between the components of the PDC method. Figure 5 illustrates the relationship between this critical iteration and the number of interpolation points s .

We can see a common trend that the stability regions are decreased with more iterations according to Figures 2–4 in this section. In fact, there are several factors having impact on the size of the stability region, including the coarse propagators, the fine propagators, the way of interpolation at those equispaced nodes, and the numerical differentiation formula of the original ODEs [32, 33]. We believe that the change of the stable regions is the result of all these factors working together accumulatively.

6. The Choice of the Fine Time Step

The PDC method includes two iterative processes, that is, parareal and deferred correction. Their individually different iteration speeds will alternatively influence the performance of the proposed hybrid algorithm. To this end, we study the balance strategy to avoid the waste of resources as well as a slowed-down speed. It is intuitively to tune the parameters of deferred correction since they are embedded in the framework parareal for our proposed PDC method. In fact, there are several ways to improve the accuracy of the final approximations for pure deferred corrections, such as smaller time step, higher order numerical integrators, or more iterations. In this study, we choose to tune the time step of the fine propagators of the PDC method.

Here we take the following system as an example to show how to find an optimized time step for the PDC method:

$$\begin{aligned} \frac{du(t)}{dt} &= Au(t), \quad t \in [0, T], \\ u(0) &= U_0. \end{aligned} \quad (47)$$

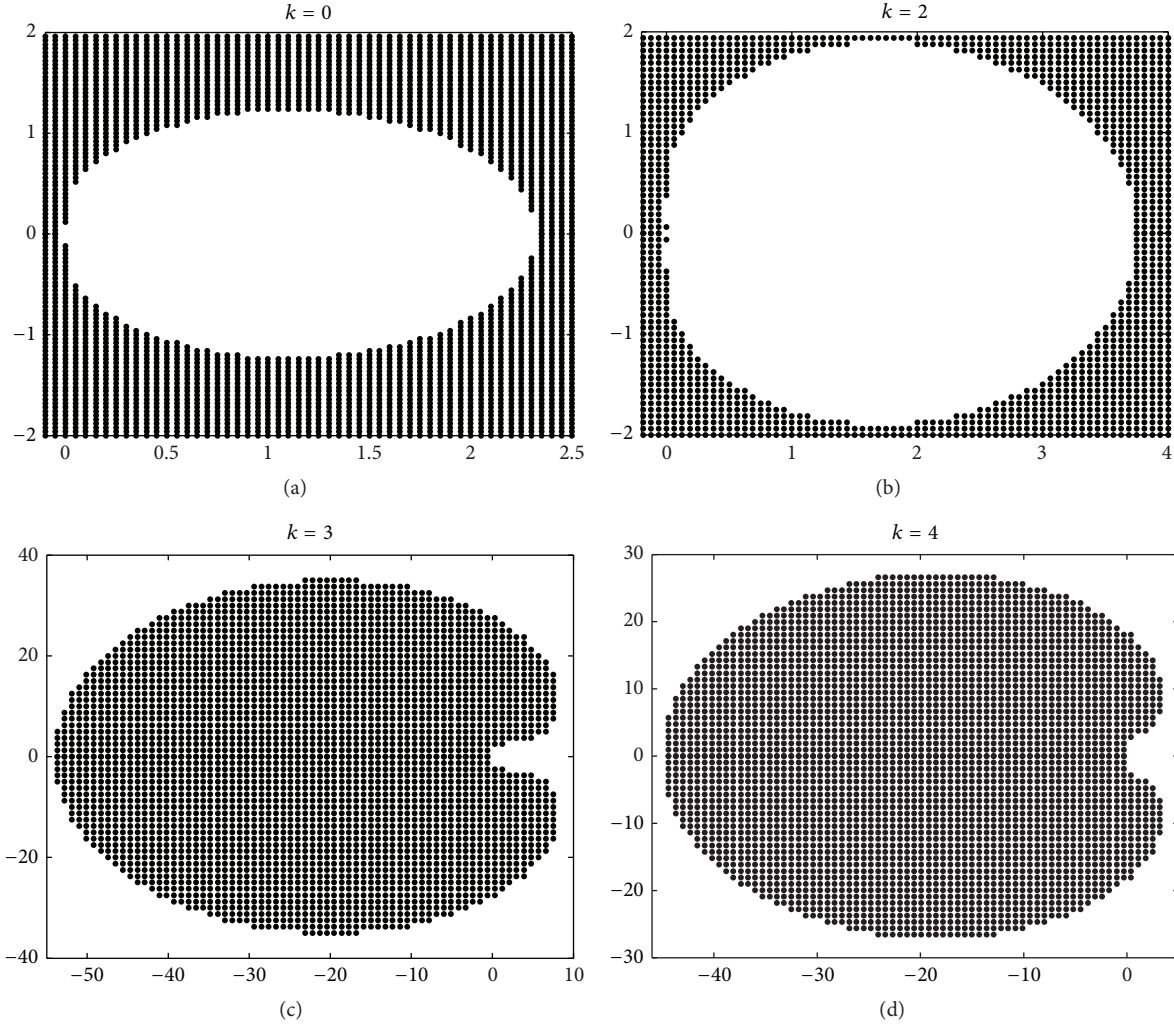


FIGURE 4: Stability region (dotted area) using implicit Euler integrators for the coarse propagators and explicit Euler integrators for the fine propagators after 1 iteration (a), 3 iterations (b), 4 iterations (c), and 5 iterations (d). The x -axis is $\text{Re}(\lambda\Delta T)$ and the y -axis is $\text{Im}(\lambda\Delta T)$, where $N = 20$, and $s = 16$.

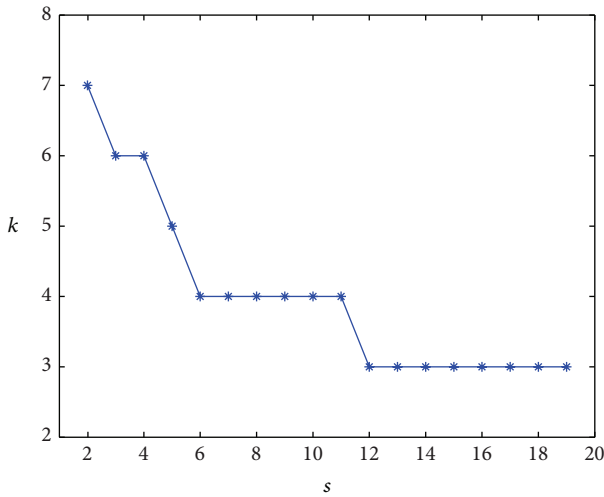


FIGURE 5: The relationship between the critical iteration and $s (< N)$, where $N = 20$.

We assume the G propagator is one-step backward Euler integrator on I_n with $G(T_{n+1}, T_n, U_n^k) = (I - A\Delta T)^{-1}U_n^k$, and the fine propagator is the M -step backward Euler integrator for prediction and each correction loop. Therefore, we have

$$\begin{aligned}
 & u(T_{n+1}) - U_{n+1}^{k+1} \\
 &= e^{A\Delta T} u(T_n) - G(T_{n+1}, T_n, U_n^{k+1}) \\
 &\quad - F_{\text{DC}}(T_{n+1}, T_n, U_n^k) + G(T_{n+1}, T_n, U_n^k) \\
 &= [e^{A\Delta T} u(T_n) - e^{A\Delta T} U_n^k] \\
 &\quad + [e^{A\Delta T} U_n^k - F_{\text{DC}}(T_{n+1}, T_n, U_n^k)] \\
 &\quad + [G(T_{n+1}, T_n, u(T_n)) - G(T_{n+1}, T_n, U_n^{k+1})] \\
 &\quad - [G(T_{n+1}, T_n, u(T_n)) - G(T_{n+1}, T_n, U_n^k)]
 \end{aligned}$$

$$\begin{aligned}
&= (e^{A\Delta T} - (I - A\Delta T)^{-1})(u(T_n) - U_n^k) \\
&\quad + (I - A\Delta T)^{-1}(u(T_n) - U_n^{k+1}) + \mathcal{O}(h^{k+1}),
\end{aligned} \tag{48}$$

where h is the fine time step. Let $\bar{\alpha} = \|e^{A\Delta T} - (I - A\Delta T)^{-1}\|$ and $\bar{\beta} = \|(I - A\Delta T)^{-1}\|$; then an upper bound of the error can be obtained from the inductive equation

$$\bar{e}_{n+1}^{k+1} = \bar{\alpha}\bar{e}_n^k + \bar{\beta}\bar{e}_n^{k+1} + \mathcal{O}(h^{k+1}). \tag{49}$$

We rewrite the recurrence relation in form of vectors, which is

$$\begin{aligned}
\begin{pmatrix} \bar{e}_1^{k+1} \\ \bar{e}_2^{k+1} \\ \vdots \\ \bar{e}_N^{k+1} \end{pmatrix} &= \begin{pmatrix} 1 & & & \\ -\bar{\beta} & 1 & & \\ \vdots & \ddots & \ddots & \\ 0 & \cdots & -\bar{\beta} & 1 \end{pmatrix}^{-1} \\
&\cdot \left[\begin{pmatrix} 0 & & & \\ \bar{\alpha} & 0 & & \\ \vdots & \ddots & \ddots & \\ 0 & \cdots & \bar{\alpha} & 0 \end{pmatrix} \begin{pmatrix} \bar{e}_1^k \\ \bar{e}_2^k \\ \vdots \\ \bar{e}_N^k \end{pmatrix} \right. \\
&\quad \left. + h \begin{pmatrix} \mathcal{O}(h^k) \\ \mathcal{O}(h^k) \\ \vdots \\ \mathcal{O}(h^k) \end{pmatrix} \right].
\end{aligned} \tag{50}$$

The fine time step h can be regarded as the convergence factor of deferred correction iterations. If r th order numerical integrator is employed in deferred correction, this factor will be obviously h^r . The constant $\bar{\alpha}$, which describes the difference between the true-solution propagator and the coarse propagator, can be regarded as a part of the convergence factor of parareal in contrast to deferred correction. We can choose h close enough to $\bar{\alpha}$ to balance the two iterations in the PDC method.

Note that the fine time step h is crucial for the final approximation performance of the PDC method. We denote the number of the fine time steps on each time subdomain I_n by s , where $s = \Delta T/h$. Polynomials with degree of s are employed in the process of deferred correction, and s is the upper bound of the final approximate order of the PDC method. So there should be a moderate integer r_1 such that $s \geq r_1$, which means the final result could reach at least the accuracy $\mathcal{O}(h^{r_1})$. However, the s can not be very large either because the polynomials of low degree are preferred for interpolations. So there should be a little bigger integer r_2 such that $s \leq r_2$ to avoid Runge's phenomenon.

Now, we get a value interval for the fine time step, with $\Delta T/r_2 \leq h \leq \Delta T/r_1$. The choice of the optimal time step is

$$h = \begin{cases} \frac{\Delta T}{r_2}, & \bar{\alpha} \leq \frac{\Delta T}{r_2}, \\ \bar{\alpha}, & \frac{\Delta T}{r_1} \leq \bar{\alpha} \leq \frac{\Delta T}{r_2}, \\ \frac{\Delta T}{r_1}, & \bar{\alpha} \geq \frac{\Delta T}{r_1}, \end{cases} \tag{51}$$

which means that h should be chosen as the nearest point to $\bar{\alpha}$ from the value interval $[\Delta T/r_2, \Delta T/r_1]$.

7. System Efficiency Analysis

Parareal can be implemented in two parallel mechanisms which are called serial-parallel parareal and pipelined parareal in [34]. For the pipelined parareal, each processor alternatively compute the fine and coarse propagators without waiting except the initial wait. It had been shown that the pipelined fashion could be better in terms of the computation speed. In this section, we study the parallel speedup and efficiency of the PDC method based on pipelined parareal method.

We first investigate the PDC method for ODEs system (12) from optimal one-step QSC method for PDE. Here we suppose backward Euler methods are used in both the coarse and fine propagators. We also assume that the computing costs of the algebraic equations involved in G and F propagators are the same if they have the same scale and sparse structure. When analyzing the speedup of the classical parareal method, we equalize the computing costs of one time step at each iteration. In this paper we do not give such assumption any more, since interpolations are necessary in correction steps of the PDC method, which is not included in the prediction step. The costs of information transmission and data storage are ignored in this paper.

Similar to the speedup and system efficiency analysis shown in [34], we adopt the following notation:

- (i) T is the length of time domain.
- (ii) ΔT is the length of each time subdomain.
- (iii) Δt is the time increment for the coarse propagator.
- (iv) δt is the time increment for the fine propagator.
- (v) τ_G is the cost of the numerical method at each time step in G .
- (vi) N_G is the number of time steps of the coarse propagator on each time subdomain, and we often choose $N_G = 1$.
- (vii) τ_F is the cost of the numerical method at each time step in F .
- (viii) N_F is the number of time steps of the fine propagator on each time subdomain.
- (ix) τ_I is the cost of interpolation averaged at each time step in correction steps.
- (x) N is the number of time subdomains on $[0, T]$; that is, $T = N\Delta T$.

- (xi) K is the number of PDC iterations.
- (xii) S is the speedup.
- (xiii) E is the system efficiency and equals the ratio $E = S/N$.

According to the pipelined parallel mechanism in [34], the initial step of the PDC method is done with the G propagator and the cost for N steps is $NN_G\tau_G$. The cost of the first iteration of the PDC method is $N_G\tau_G + N_F\tau_F$, while the cost of any other iteration of the PDC method is $N_G\tau_G + N_F(\tau_F + \tau_I)$.

The total cost for the initial and K iterations of the PDC method is

$$(K + N)N_G\tau_G + KN_F\tau_F + (K - 1)N_F\tau_I. \quad (52)$$

Let M be the number of deferred correction iterations to compute the same problem sequentially on the fine DC nodes with one processor to achieve the same grade accuracy as K PDC iterations. Then the cost of the serial DC method is $N(MN_F\tau_F + (M - 1)N_F\tau_I)$. The parallel speedup S of the PDC method is

$$S = \frac{N(MN_F\tau_F + (M - 1)N_F\tau_I)}{(K + N)N_G\tau_G + KN_F\tau_F + (K - 1)N_F\tau_I}, \quad (53)$$

and the system efficiency

$$E = \frac{S}{N} = \frac{MN_F\tau_F + (M - 1)N_F\tau_I}{(K + N)N_G\tau_G + KN_F\tau_F + (K - 1)N_F\tau_I}. \quad (54)$$

If we further assume $\tau_G = \tau_F$ and $\tau_I = \beta\tau_F$ for simplicity and let $\alpha = N_G/N_F$, the efficiency will be

$$\begin{aligned} E &= \frac{M + (M - 1)\beta}{(K + N)\alpha + K + (K - 1)\beta} \\ &= \frac{M - \beta/(1 + \beta)}{(K + N)(\alpha/(1 + \beta)) + K - \beta/(1 + \beta)}, \end{aligned} \quad (55)$$

where β denotes the ratio between the cost for solving algebraic equation and the cost for interpolations; it is often a small constant according to the numerical experiment later.

It is reported in [34] that the parallel efficiency of the classical pipelined parareal under our simplifying assumptions is

$$\frac{1}{(K + N)\alpha + K}, \quad (56)$$

which is automatically bounded by $1/K$. In contrast, the efficiency of PDC method is bounded by M/K , which is apparently better than that of the classical parareal method. However, we have to pay additional attention to the constant α . The constant α denotes the ratio between the time steps of the coarse and the fine propagators, since we assume $\tau_G = \tau_F$. For the classical parareal method, it is required that N_F is much greater than N_G , α for classical parareal method is much less than 1, and the resulting efficiency is very close to its upper bound. For the PDC method, the accuracy is improved by iterative corrections, and N_F is not needed to

be much greater than N_G ; then α for PDC method is not so small. Therefore, the resulting efficiency of PDC method is not so close to its upper bound as the efficiency of the classical parareal method.

In order to show the contribution of QSC method to the whole algorithm, we investigate the PDC method for the following ODEs system generated after the classical finite difference for PDE

$$\begin{aligned} \frac{du(t)}{dt} &= \frac{1}{\Delta x^2} Au(t) + \vec{f}(t), \quad t \in [0, T] \\ u(0) &= u_0, \end{aligned} \quad (57)$$

where Δx is the size of spatial grid, the coefficient matrix

$$A = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ & & \ddots & \ddots & \ddots \\ 0 & \cdots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix}, \quad (58)$$

and $\vec{f}(t)$ is the interpolation vector of $f(x, t)$ at equispaced grid points.

Let $\bar{\tau}_G$ and $\bar{\tau}_F$ be the cost of the numerical method at each time step in G and F , respectively. Let $\bar{\tau}_I$ be the cost of the interpolation averaged at each time step and define $\bar{\beta} = \bar{\tau}_I/\bar{\tau}_F$. Obviously, $\bar{\tau}_G$ and $\bar{\tau}_F$ are directly related to the scale and structure of the algebraic equations involved. As we know, the finite difference method has the approximate order $\mathcal{O}(\Delta x^2)$, while the optimal QSC method has the order $\mathcal{O}(\Delta x^4)$. To achieve the same level of error, the dimension of system (57) should be much greater than the dimension of system (12). Let n_1 and n_2 be the dimensions of system (12) and system (57), respectively. From $((b - a)/n_1)^4 \sim ((b - a)/n_2)^2$, we obtain $n_2 \sim n_1^2$.

For the PDC method for system (12), the coefficient matrices of the linear equations have sparse structure with bandwidth 4, and the corresponding time complexity is $\mathcal{O}(4n_1)$. For the PDC method for system (57), the coefficient matrices of the linear equations have sparse structure with bandwidth 1, and the corresponding time complexity is $\mathcal{O}(n_2)$. Then we have approximate relationships $\bar{\tau}_G \approx (n_1/4)\tau_G$ and $\bar{\tau}_F \approx (n_1/4)\tau_F$, and the approximate system efficiency is

$$\bar{E} = \frac{M - \bar{\beta}/(1 + \bar{\beta})}{(K + N)(\alpha/(1 + \bar{\beta})) + K - \bar{\beta}/(1 + \bar{\beta})}, \quad (59)$$

which is almost the same as E , except for a different constant $\bar{\beta}$. Therefore, the application of the QSC method nearly has no effect to increase the system efficiency.

The total cost after K iterations of the PDC method for system (57) is

$$\begin{aligned} &(K + N)N_G\bar{\tau}_G + KN_F\bar{\tau}_F + (K - 1)N_F\bar{\tau}_I \\ &\approx \frac{n_1}{4} \left[(K + N)N_G\tau_G + KN_F\tau_F + (K - 1)N_F\frac{\bar{\beta}}{\beta}\tau_I \right], \end{aligned} \quad (60)$$

TABLE 2: The iteration error and running time of QSC-PDC method, with $N = 40, s = 5$.

Iterations	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$
Error	$2.05e - 6$	$1.65e - 5$	$1.57e - 6$	$3.73e - 7$	$1.38e - 7$	$3.25e - 9$	$4.89e - 9$	$7.71e - 10$	$1.50e - 10$	$5.71e - 11$
Time (s)	0.47	0.68	0.87	1.10	1.30	1.50	1.73	1.92	2.11	2.32

TABLE 3: The iteration error and running time of classical method for the QSC equations, with Gauss methods of $N = 40$ and 200 steps on each subdomain.

Iterations	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$
Error	$8.91e - 5$	$2.83e - 5$	$9.04e - 6$	$2.86e - 6$	$9.29e - 7$	$2.75e - 7$	$1.09e - 7$	$1.24e - 8$	$2.67e - 8$	$1.45e - 8$
Time (s)	1.24	2.13	3.11	4.10	5.09	6.02	6.97	7.91	8.95	9.83

where the ratio $\bar{\beta}/\beta$ is a small constant. The cost of PDC method for system (57) is about $n_1/4$ times of the cost of the PDC method for system (12). Since n_1 is a very big integer, the QSC method reduces the total cost greatly. Furthermore, if the optimal two-step QSC method is employed, the reduction factor will be $n_1/2$ rather than $n_1/4$.

8. Numerical Experiments

In this section, we take the following parabolic PDE as an example to present the advantages of the QSC-PDC algorithm:

$$\begin{aligned} \frac{\partial}{\partial t} u(x, t) &= \frac{\partial^2}{\partial x^2} u(x, t) + (2\pi^2 - 1) e^{-t/2} \sin(\pi x), \\ u(0, t) &= u(1, t) = 0, \quad 0 \leq t \leq 5, \\ u(x, 0) &= 2 \sin(\pi x), \quad 0 \leq x \leq 1. \end{aligned} \tag{61}$$

The true solution of system (61) is

$$u(x, t) = 2e^{-t/2} \sin(\pi x), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 5. \tag{62}$$

To perform the QSC-PDC method, we first employ a uniform partition of the spatial domain $[0, 1]$ with 160 equispaced grid points, and the step size is $\Delta x = 1/160$. Then we take the midpoint of each segment as the collocation points; that is,

$$\left\{ \tau_i = \left(i - \frac{1}{2} \right) \Delta x, \quad i = 1, 2, \dots, 160 \right\}. \tag{63}$$

Define the vector $\vec{g} = (\sin(\pi\tau_1), \sin(\pi\tau_2), \dots, \sin(\pi\tau_{160}))^T$. The resulting collocation equation is

$$\begin{aligned} Q_0 \frac{dc(t)}{dt} &= \frac{1}{\Delta x^2} \left(Q_2 + \frac{1}{24} Q_{xx} \right) c(t) \\ &+ (2\pi^2 - 1) e^{-t/2} \vec{g}, \quad t \in [0, 5], \\ c(0) &= 2Q_0 \vec{g}, \end{aligned} \tag{64}$$

where the matrices $Q_0, Q_2,$ and $Q_{xx} \in R^{160 \times 160}$ are the same as the matrices in system (12). Here, we only consider the optimal one-step QSC method.

We divide the time domain $[0, 5]$ into 40 subdomains. On each subdomain, we use 1-step backward Euler method for

the G propagators and use 5-step backward Euler method for the F_{DC} propagators. Polynomials of degree 5 are employed for interpolation in each correction loop. The PDC method is programmed in the pipelined manner, and information is communicated between processors through MPI, and the code is written using C language with GNU Scientific Library. The program is carried out on the Shared Hierarchical Academic Research Computing Network of Canada (<https://www.sharcnet.ca/>). 41 CPUs are employed in the experiments, including one root CPU and the other 40 CPUs. The root CPU is in charge of sending and receiving data and displaying outputs. The iterative computing of the propagators on the 40 time subdomains are distributed onto the other 40 CPUs.

For simplicity, we only examine the error at a special point $(x, t) = ((80 - 1/2)\Delta x, 5)$. The resulting PDC error and corresponding running time are shown in Table 2. In order to exhibit the advantages of the QSC-PDC method, we perform the classical parareal method for system (64), and the spatial variable is also discretized by optimal QSC. Here, we call the reference method as QSC-parareal method. The time domain $[0, 5]$ is also divided into 40 subdomains, and the coarse propagator is also chosen as 1-step backward Euler method on each subdomain. The fine propagator is chosen as 100-step 2-stage Gauss method on each subdomain, which is of high order and easy to carry out.

The classical parareal method for (64) is also programmed in pipelined manner, and the corresponding error at the same point $(x, t) = ((80 - 1/2)\Delta x, 5)$ and running time are shown in Table 3.

To show the advantages of the deferred correction clearly, we plot the iteration error and running time of the two methods together in Figure 6. We can see that the deferred correction technique makes the parareal method achieve better approximations in fixed iterations and costs much less running time.

9. Conclusions

In this paper, we firstly applied the optimal one-step QSC method to solve linear parabolic PDEs, which lead to linear ODEs with respect to the degrees of freedom. We then combined the parareal algorithm with deferred correction method along the temporal direction to deal with the collocation equations. With the fourth order error on the spatial

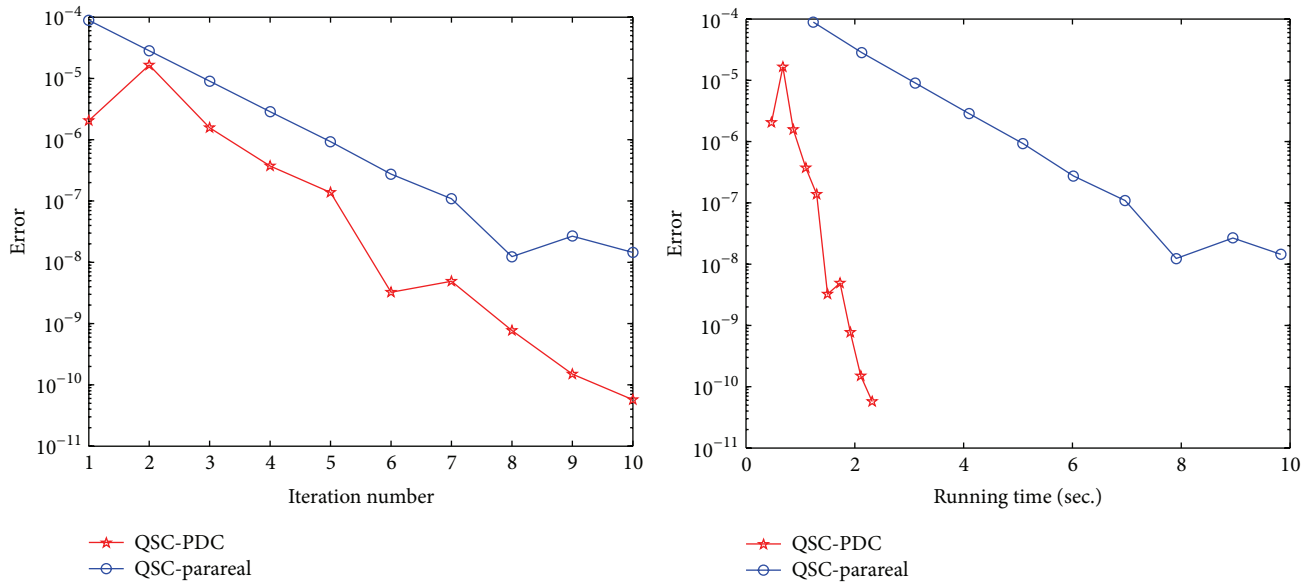


FIGURE 6: The comparison of error and cost between QSC-PDC method and QSC-parareal method for system (61).

step size, high accuracy could be obtained without too many collocation points. We also supplied an error estimation of the PDC methods, which illustrates good convergence with small time subdomains. We also analyzed the stabilities of the PDC method with different cases of coarse and fine propagators. We presented the tuning strategy for the time step size of the deferred correction method to balance the parareal and the deferred correction iterations of distinguish individual computational speeds.

Compared with the classical parareal method, PDC method has much higher parallel efficiency, and the optimal QSC can reduce the computational cost greatly when achieving a desired accuracy. By the numerical experiments on multicore computers, the advantages of the QSC-PDC algorithm are obvious.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank Professor Kenneth R. Jackson of University of Toronto for his valuable suggestions. They also thank Shared Hierarchical Academic Research Computing Network of Canada and its staff for supplying multicore computers and for guiding them to use it. This work was supported by the National Natural Science Foundation of China (no. 11401589, no. 11301543, no. 11326246, and no. 11571367), Shandong Provincial Natural Science Foundation (no. ZR2013AL018), and the Fundamental Research Funds for the Central Universities (no. 14CX02148A).

References

- [1] J. L. Lions, Y. Maday, and G. Turinici, "A 'parareal' in time discretization of PDE's," *Comptes Rendus del Academie des Sciences Serie I-Mathematique*, vol. 332, no. 7, pp. 661–668, 2001.
- [2] G. Bal, "On the convergence and the stability of the parareal algorithm to solve partial differential equations," in *Proceedings of the 15th International Conference on Domain Decomposition Methods on Science and Engineering*, Berlin, Germany, July 2003.
- [3] G. Bal and Y. Maday, "A 'parareal' time discretization for non-linear PDE's with application to the pricing of an American put," in *Proceedings of the Workshop on Domain Decomposition Methods*, Zurich, Switzerland, June 2001.
- [4] S. Vandewalle and M. J. Gander, "The parareal algorithm in a historical perspective," in *Proceedings of the 16th International Conference on Domain Decomposition Methods on Science and Engineering*, New York, NY, USA, January 2005.
- [5] M. J. Gander, "Analysis of the parareal algorithm applied to hyperbolic problems using characteristics," *Boletín de la Sociedad Española de Matemática Aplicada*, vol. 42, pp. 21–35, 2008.
- [6] P. F. Fischer, F. Hecht, and Y. Maday, "A parareal in time semi-implicit approximation of the Navier-Stokes equations," in *Proceedings of the 15th International Conference on Domain Decomposition Methods in Science and Engineering*, Berlin, Germany, July 2003.
- [7] 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.
- [8] Y. Maday and G. Turinici, "Parallel in time algorithms for quantum control: parareal time discretization scheme," *International Journal of Quantum Chemistry*, vol. 93, no. 3, pp. 223–228, 2003.
- [9] Y. Maday and G. Turinici, "A parareal in time procedure for the control of partial differential equations," *Comptes Rendus Mathematique*, vol. 335, no. 4, pp. 387–392, 2002.
- [10] T. P. Mathew, M. Sarkis, and C. Schaerer, "Analysis of block parareal preconditioners for parabolic optimal control problems," *SIAM Journal on Scientific Computing*, vol. 32, no. 3, pp. 1180–1200, 2010.
- [11] M. Minion and S. Williams, "Parareal and spectral deferred corrections," *AIP Conference Proceedings*, vol. 1048, pp. 388–391, 2008.

- [12] Y. Maday and G. Turinici, "The parareal in time iterative solver: a further direction to parallel implementation," in *Proceedings of the 15th International Conference on Domain Decomposition Methods in Science and Engineering*, Berlin, Germany, July 2003.
- [13] M. J. Gander and M. Petcu, "Analysis of a Krylov subspace enhanced parareal algorithm for linear problems," *ESAIM. Proceedings*, vol. 25, pp. 114–129, 2008.
- [14] D. Guibert and D. Tromeur-Dervout, "Adaptive parareal for systems of ODEs," in *Proceedings of the 16th International Conference on Domain Decomposition Methods on Science and Engineering*, New York, NY, USA, January 2005.
- [15] J. Liu and Y.-L. Jiang, "A parareal algorithm based on waveform relaxation," *Mathematics and Computers in Simulation*, vol. 82, no. 11, pp. 2167–2181, 2012.
- [16] J. Liu and Y.-L. Jiang, "A parareal waveform relaxation algorithm for semi-linear parabolic partial differential equations," *Journal of Computational and Applied Mathematics*, vol. 236, no. 17, pp. 4245–4263, 2012.
- [17] J. Liu, Y. Wang, and R. J. Li, "Quadratic spline collocation and parareal deferred correction method for parabolic PDEs," *AIP Conference Proceedings*, In press.
- [18] 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.
- [19] S. Vandewalle, *Multilevel Time-Integration: Analysis of the Parareal Algorithm*, Numerical Analysis Seminar Series, Oxford Computing Laboratory, University of Oxford, New York, NY, USA, 2007.
- [20] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations," *BIT Numerical Mathematics*, vol. 40, no. 2, pp. 241–266, 2000.
- [21] A. C. Hansen and J. Strain, "On the order of deferred correction," *Applied Numerical Mathematics*, vol. 61, no. 8, pp. 961–973, 2011.
- [22] K. Böhmer and H. J. Stetter, *Defect Correction Methods: Theory and Applications*, Springer, New York, NY, USA, 1984.
- [23] R. Frank and C. W. Ueberhuber, "Iterated deferred correction for the efficient solution of stiff systems of ordinary differential equations," *BIT*, vol. 17, pp. 146–159, 1977.
- [24] E. Hairer, "On the order of iterated defect correction," *Numerical Mathematics*, vol. 29, no. 4, pp. 409–424, 1978.
- [25] E. N. Houstis, C. C. Christara, and J. R. Rice, "Quadratic-spline collocation methods for two-point boundary value problems," *International Journal for Numerical Methods in Engineering*, vol. 26, no. 4, pp. 935–952, 1988.
- [26] C. C. Christara, "Quadratic spline collocation methods for elliptic partial differential equations," *BIT Numerical Mathematics*, vol. 34, no. 1, pp. 33–61, 1994.
- [27] T. Chen, *An efficient algorithm based on quadratic spline collocation and finite difference methods for parabolic partial differential equations [M.S. thesis]*, University of Toronto, Toronto, Canada, 2005.
- [28] C. C. Christara, T. Chen, and D. M. Dang, "Quadratic spline collocation for one-dimensional linear parabolic partial differential equations," *Numerical Algorithms*, vol. 53, no. 4, pp. 511–553, 2010.
- [29] M. J. Gander and E. Hairer, "Nonlinear convergence analysis for the parareal algorithm," in *Domain Decomposition Methods in Science and Engineering XVII*, vol. 60 of *Lecture Notes in Computational Science and Engineering*, pp. 45–56, Springer, Berlin, Germany, 2008.
- [30] A. Christlieb, B. Ong, and J.-M. Qiu, "Integral deferred correction methods constructed with high order Runge-Kutta integrators," *Mathematics of Computation*, vol. 79, no. 270, pp. 761–783, 2010.
- [31] 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.
- [32] J. Huang, J. Jia, and M. Minion, "Accelerating the convergence of spectral deferred correction methods," *Journal of Computational Physics*, vol. 214, no. 2, pp. 633–656, 2006.
- [33] L. N. Trefethen and M. R. Trummer, "An instability phenomenon in spectral methods," *SIAM Journal on Numerical Analysis*, vol. 24, no. 5, pp. 1008–1023, 1987.
- [34] 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.



Hindawi

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

