Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 1739–1753
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# Krylov deferred correction accelerated method of lines transpose for parabolic problems

Jun Jia [a,*,1], Jingfang Huang [b,2]

[a] Oak Ridge National Laboratory, Oak Ridge, TN 37831-6367, USA
[b] Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599-3250, USA

Received 27 March 2007; received in revised form 12 September 2007; accepted 26 September 2007
Available online 6 October 2007

**Abstract**

In this paper, a new class of numerical methods for the accurate and efficient solutions of parabolic partial differential equations is presented. Unlike traditional method of lines (MoL), the new *Krylov deferred correction (KDC) accelerated method of lines transpose* ($\text{MoL}^T$) first discretizes the temporal direction using Gaussian type nodes and spectral integration, and symbolically applies low-order time marching schemes to form a preconditioned elliptic system, which is then solved iteratively using Newton–Krylov techniques such as Newton–GMRES or Newton–BiCGStab method. Each function evaluation in the Newton–Krylov method is simply one low-order time-stepping approximation of the error by solving a decoupled system using available fast elliptic equation solvers. Preliminary numerical experiments show that the KDC accelerated $\text{MoL}^T$ technique is unconditionally stable, can be spectrally accurate in both temporal and spatial directions, and allows optimal time-stepsizes in long-time simulations.


*MSC:* 65B05; 65F10; 65M99

*Keywords:* Krylov subspace methods; Krylov deferred correction; Method of lines; Preconditioners; Parabolic partial differential equations

## 1. Introduction

In this paper, we consider the numerical solution of a general parabolic partial differential equation (PDE) of the form

* Corresponding author.
*E-mail addresses:* jiaj@ornl.gov (J. Jia), huang@amath.unc.edu (J. Huang).
[1] The work of this author was supported by the Office of Advanced Scientific Computing Research; US Department of Energy. This document describes activities performed under Contract Number DE-AC0500OR22750 between the US Department of Energy and Oak Ridge Associated Universities. All opinions expressed in this report are the authors' and do not necessarily relect policies and views of the US Department of Energy or the Oak Ridge Institute for Science and Education.
[2] The work of this author was supported in part by NSF under Grant Nos. DMS0411920 and DMS0327896.

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.09.018

$$\mathcal{L}(u_t, u, u_x, u_{xx}) = 0 \tag{1}$$

where $u = u(x,t)$, $x \in [a,b]$, $t \in [0,T]$, and proper initial and boundary conditions are given. Examples include the well-known diffusion (heat) equation

$$u_t = Du_{xx} + f(x,t)$$

where $D$ is the diffusion coefficient; the diffusion-reaction equation

$$u_t = Du_{xx} + f(u)$$

which models many biological and chemical reaction processes; the nonlinear Schrödinger equation (NLS)

$$i\epsilon u_t + \frac{\epsilon^2}{2}u_{xx} - Vu - f(\|u\|^2)u - \epsilon\tau \arg(u)u = 0$$

in solid state physics; and the Richards' equation

$$[c(u) + S_s S_a(u)]u_t = [K_x(u)(u_x + 1)]_x$$

which simulates fluid flow and species transport in subsurface systems. In Eq. (1), $u$ may be a vector of unknowns and hence $\mathcal{L} = 0$ is a system of equations. To simplify the discussions and focus on the ideas, we describe our algorithms in $1+1$ dimensions in this paper. However, the paper is organized such that generalization of the analyses and algorithms to higher dimensions is straightforward, and numerical results in $3+1$ dimensions are also presented.

Perhaps due to the popularity and success of numerical methods for solving initial value problems governed by systems of ordinary differential equations (ODEs) and differential algebraic equations (DAEs), for time dependent parabolic PDEs, a common practice is to first discretize the PDEs in the spatial direction using finite difference, finite element, or spectral methods, and apply available solvers to the resulting ODE/DAE system in which each entry represents the approximate solution at a specific location for all times. This approach is commonly referred to as the "method of lines" (MoL) and interested readers are referred to [24] for further discussions. We want to mention that although great progress has been made in the last century, the disadvantages of existing MoL type time-stepping schemes using available ODE/DAE solvers are also becoming obvious. These include (a) the step-sizes of available ODE/DAE solvers are often constrained by the stability properties of the underlying methods, which may take the form of a CFL condition for certain PDE problems and lead to unacceptably small time steps for many applications; (b) higher order versions of existing ODE/DAE solvers either lack desired stability properties or are extremely complicated to solve, in particular, as far as we know, no spectral or pseudo-spectral discretization based solvers are available for initial value problems; and (c) discretization in the spatial direction changes the structure of the PDEs, hence existing extremely efficient elliptic equation solvers (e.g. fast multipole methods (FMM) accelerated integral equation methods, and multigrid accelerated finite difference or finite element solvers) cannot be easily utilized by available ODE solvers.

In this paper, inspired by the recent success of fast elliptic equation solvers (in particular, the FMM accelerated integral equation methods [6,20]) and the unconditionally stable arbitrary order Krylov deferred correction (KDC) methods for initial value problems [14,15], we discuss a new framework for constructing efficient and accurate numerical methods for Eq. (1). In the new methods, to march the evolutionary PDEs from $t = 0$ to $\Delta t$, we first discretize the temporal direction (the transpose direction of traditional MoL) using $p$ Gaussian type nodes $\mathbf{t} = [t_1, t_2, \ldots, t_p]^{\mathrm{T}}$. To avoid the numerically unstable differentiation operator, we define at each node point $U_i(x) = U(x,t_i) = u_t(x,t_i)$ as the new unknown, and the discretized equations become

$$\mathbf{L}\left(\mathbf{U}, \mathbf{u_0} + \Delta t S \otimes \mathbf{U}, \frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{u_0} + \Delta t S \otimes \mathbf{U}), \frac{\mathrm{d}^2}{\mathrm{d}x^2}(\mathbf{u_0} + \Delta t S \otimes \mathbf{U})\right) = \mathbf{0} \tag{2}$$

where $\mathbf{U} = [U_1(x), U_2(x), \ldots, U_p(x)]^{\mathrm{T}}$ is the desired approximation of $U(x,t)$ at different node points, $\mathbf{u_0} = [u(x,0), u(x,0), \ldots, u(x,0)]^{\mathrm{T}}$ the initial condition, $S$ the spectral integration matrix as discussed in [8], and $\otimes$ the tensor product (i.e. $\Delta t S$ is applied to each component of $\mathbf{U}$). Due to the use of Gaussian type node points, this "Method of Lines Transpose" ($\mathrm{MoL}^{\mathrm{T}}$) discretization achieves optimal order accuracy in each time step for non-degenerating parabolic equations. However, the resulting elliptic equations are coupled as the

spectral integration matrix $S$ is dense, hence direct solution of Eq. (2) is in general computationally inefficient. Instead, similar to the KDC methods, we assume a provisional solution $\widetilde{\mathbf{U}} = [\widetilde{U}_1(x), \widetilde{U}_2(x), \ldots, \widetilde{U}_p(x)]^{\mathrm{T}}$ is given, and define the error as $\boldsymbol{\delta} = \mathbf{U} - \widetilde{\mathbf{U}} = [\delta_1, \delta_2, \ldots, \delta_p]^{\mathrm{T}}$. A simple substitution yields an equation for $\boldsymbol{\delta}$

$$\mathbf{L}\left(\widetilde{\mathbf{U}} + \boldsymbol{\delta}, \mathbf{u_0} + \Delta t S \otimes (\widetilde{\mathbf{U}} + \boldsymbol{\delta}), \frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{u_0} + \Delta t S \otimes (\widetilde{\mathbf{U}} + \boldsymbol{\delta})), \frac{\mathrm{d}^2}{\mathrm{d}x^2}(\mathbf{u_0} + \Delta t S \otimes (\widetilde{\mathbf{U}} + \boldsymbol{\delta}))\right) = \mathbf{0} \tag{3}$$

A low-order approximation $\bar{\boldsymbol{\delta}}$ of $\boldsymbol{\delta}$ can then be obtained by solving the equation

$$\mathbf{L}\left(\widetilde{\mathbf{U}} + \bar{\boldsymbol{\delta}}, \mathbf{u_0} + \Delta t S \otimes \widetilde{\mathbf{U}} + \Delta t \widetilde{S} \otimes \bar{\boldsymbol{\delta}}, \frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{u_0} + \Delta t S \otimes \widetilde{\mathbf{U}} + \Delta t \widetilde{S} \otimes \bar{\boldsymbol{\delta}}), \frac{\mathrm{d}^2}{\mathrm{d}x^2}(\mathbf{u_0} + \Delta t S \otimes \widetilde{\mathbf{U}} + \Delta t \widetilde{S} \otimes \bar{\boldsymbol{\delta}})\right) = \mathbf{0} \tag{4}$$

using a low-order time marching scheme, where $\widetilde{S}$ is the resulting lower-triangular approximation of the spectral integration matrix $S$. Notice that the equations for $\bar{\boldsymbol{\delta}}$ are "decoupled" at different times (the equation at node $t_i$ only involves $U_i(x)$ and previously computed $U_j(x)$ for $j < i$), hence the elliptic equation at each node can be solved efficiently using available fast adaptive elliptic equation solvers. As the Jacobian matrix of the "implicit" function $\bar{\boldsymbol{\delta}} = \widetilde{\mathbf{H}}(\widetilde{\mathbf{U}})$ (we consider $\widetilde{\mathbf{U}}$ as the input variable in Eq. (4), and $\bar{\boldsymbol{\delta}}$ the output) is close to $-I$, the Newton–Krylov methods can be applied to find the zero of $\widetilde{\mathbf{H}}(\widetilde{\mathbf{U}}) = \mathbf{0}$, an equivalent form of the collocation formulation in Eq. (2). We refer to this new time-stepping procedure as the KDC accelerated MoL$^{\mathrm{T}}$.

We want to mention that although less commonly used when compared with traditional MoL, the MoL$^T$ is not new and has been previously studied by different authors. In particular, the transversal line method (Rothe's method) has been applied to PDEs of both parabolic and hyperbolic types, with the immediate advantage that existing finite difference or finite element elliptic equation solvers can be easily adapted for better efficiency and accuracy. However, in these attempts, very low-order time discretization schemes (e.g. backward Euler's method) were used, to avoid the coupled elliptic type equations in space resulting from higher order discretizations. In this paper, by coupling the KDC methods and the fast elliptic equation solvers, we show how this hurdle can be overcome for optimal accuracy and efficiency in both time and space.

This paper is organized as follows. In Section 2, we briefly discuss several essential preliminaries, including the Krylov deferred correction methods for ODE and DAE problems and fast elliptic equation solvers using integral equation formulations. The KDC accelerated MoL$^T$ is then introduced in Section 3, and properties of the algorithm are briefly discussed in Section 3.4. In Section 4, preliminary numerical results are presented.

## 2. Preliminaries

In this section, we discuss two recently developed numerical techniques which serve as fundamental building blocks for the new class of time-stepping methods for parabolic PDEs. Specifically, Krylov deferred correction methods for ODE/DAE initial value problems and fast elliptic solvers are discussed in order.

### 2.1. Krylov deferred correction methods for initial value problems

The Krylov deferred correction methods [14] are based on the Newton–Krylov methods and spectral deferred correction techniques briefly presented next.

#### 2.1.1. Newton–Krylov methods

Consider a general algebraic system $M(x) = 0$ with $N$ equations and unknowns, and suppose an approximate solution $x_0$ is known. Newton's method can be used to iteratively compute a sequence of quadratically convergent approximations (assuming the Jacobian matrix $J_M$ is non-singular at the solution) by updating

$$x_{n+1} = x_n - \delta x,$$

where $\delta x$ is the solution of the linear equation

$$J_M(x_n)\delta x = b \tag{5}$$

with $b = M(x_n)$ and $J_M(x_n)$ the Jacobian matrix of $M(x)$ at $x_n$. When the matrix $J_M$ is dense, computing the solution of this linear equation with Gaussian elimination requires $O(N^3)$ operations. However, for many special matrices, the amount of work required to find the solution can be greatly reduced. Consider the case

$$J_M(x_n) = \pm I - C$$

where most of the eigenvalues of $C$ are clustered close to 0. Because of the rapid decay of most eigenmodes in $C^q b$, a more efficient approach than Gaussian elimination is to iteratively search for the optimal solution in the Krylov subspace defined by

$$K_q(J_M, b) = \{b, Cb, C^2 b, \ldots, C^q b\}$$

The iterations in Newton's method and Krylov subspace methods can then be intertwined by reducing the residual of the linearized system (5) by a prescribed factor using the Krylov subspace methods (instead of solving Eq. (5) exactly), and then start a new Newton's iteration. Interested readers are referred to [16,17,23] for detailed discussions of the resulting Newton–Krylov methods. In general, an efficient numerical implementation of a Newton–Krylov method depends on: **(a)** a formulation of the problem $M(x) = 0$ such that $J_M$ is close to the identity matrix $\pm I$; and **(b)** an efficient procedure for computing the matrix vector product $Cb$ (or equivalently $J_M b$). For (a), one common technique is to apply a "preconditioner" to the original system; and for (b), when the analytical Jacobian is not readily available or inefficient to evaluate, a general forward difference approximation can be adapted where for any vector $v$, $J_M(x)v$ is approximated by

$$D_h M(x : v) = (M(x + hv) - M(x))/h \tag{6}$$

for some properly chosen parameter $h$ ($h$ may be complex). This difference approximation technique as well as the choice of $h$ have been carefully studied previously [16] and interested readers are referred to [18] for a survey of existing Jacobian-free Newton–Krylov methods and applications.

### 2.1.2. Spectral deferred correction (SDC) methods

Instead of a discussion of classical deferred and defect correction methods first introduced by Pereyra and Zadunaisky [22,27,28], in the following, we focus on the SDC methods introduced in 2000 [5] for ODE initial value problems

$$\varphi'(t) = f(t, \varphi(t)), \quad t \in [a, b], \quad \varphi(a) = \varphi_0.$$

As with classical deferred and defect correction methods, a single time step of an SDC method begins by first dividing the time step $[0, \Delta t]$ into a set of intermediate sub-steps defined by the points $\mathbf{t} = [t_1, t_2, \ldots, t_p]^T$ with $0 \leqslant t_1 < \cdots < t_p \leqslant \Delta t$ (in SDC methods, $\mathbf{t}$ corresponds to the quadrature nodes of Gaussian type to avoid the unstable interpolation using uniform grids). Second, a provisional approximation $\tilde{\boldsymbol{\varphi}} = [\varphi_1^0, \varphi_2^0, \ldots, \varphi_p^0]^T$ is computed at the intermediate points using a low-order time-stepping scheme. Applying standard interpolation theory, the continuous counterpart of $\tilde{\boldsymbol{\varphi}}$ (for example, the Legendre polynomial expansion) is then constructed and is denoted by $\varphi^0(t)$. Third, utilizing the Picard integral equation, a corresponding integral equation for the error $\delta(t) = \varphi(t) - \varphi^0(t)$ is constructed. Specifically

$$\delta(t) = \int_0^t [f(\tau, \varphi^0(\tau) + \delta(\tau)) - f(\tau, \varphi^0(\tau))]\mathrm{d}\tau + \epsilon(t) \tag{7}$$

where

$$\epsilon(t) = \varphi_0 + \int_0^t f(\tau, \varphi^0(\tau))\mathrm{d}\tau - \varphi^0(t) \tag{8}$$

and the approximations $[\epsilon_1, \epsilon_2, \ldots, \epsilon_p]$ can be computed using spectral integration [8]. Note that the Picard integral equation is numerically more stable than the original ODE which is used in the original deferred and defect correction methods. Fourth, a lower order method is applied to approximate $\delta(t)$ in Eq. (7). For example, the backward Euler type method

$$\tilde{\delta}_{m+1} = \tilde{\delta}_m + \Delta t_m \left[ f\left(t_{m+1}, \varphi_{m+1}^0 + \tilde{\delta}_{m+1}\right) - f\left(t_{m+1}, \varphi_{m+1}^0\right) \right] + \epsilon_{m+1} - \epsilon_m. \tag{9}$$

The solution $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}_1, \tilde{\delta}_2, \ldots, \tilde{\delta}_p]^T$ which approximates $\delta(t)$ is then added to the provisional solution $\tilde{\boldsymbol{\varphi}}$ to form a more accurate provisional solution. Finally, spectral integration is then again applied using Eq. (8) to accurately compute the new $\epsilon(t)$, and the iteration procedure continues until the residual is smaller than a prescribed tolerance or a maximum prescribed number of iterations is reached. Note that the computational complexity of solving the implicit Eq. (9) is the same as that of the low-order time-stepping method.

*2.1.3. Krylov deferred correction (KDC) methods*

In [19], it was noticed that when the original SDC methods are applied to stiff ODE problems, similar to traditional initial value problem solvers, order reduction can be observed. This phenomenon was explained in [14] by observing that the original SDC method for a linear problem is equivalent to a Neumann series expansion

$$x = b + Cb + C^2 b + \cdots + C^k b \cdots$$

for solving a preconditioned system $(I - C)x = b$, where the preconditioner is one SDC iteration. Clearly, when some of the eigenvalues of $C$ are close to (but less than) 1, the Neumann series converges slowly, resulting in the "order reduction". Even worse, for DAE systems of the form

$$F(y(t), y'(t), t) = 0, \tag{10}$$

it is numerically shown in [15] that straightforward extension of the SDC method is divergent for many systems as there exist eigenvalues with magnitude greater than one independent of the time step-size.

To accelerate the convergence (and avoid the divergence) of the original SDC methods, in [15], the KDC methods were invented for general DAE systems (ODEs are index 0 DAEs). In these methods, a Picard integral type formulation is first employed

$$F\left(y_0 + \int_0^t Y(\tau)\mathrm{d}\tau, Y(t), t\right) = 0$$

where $Y(t) = y\prime(t)$ is introduced as the new unknown function and $y(t)$ is recovered using quadratures. Introducing the spectral integration matrix $S$ as in [14], the discretized collocation formulation is given by

$$\vec{F}(\vec{y_0} + \Delta t S \otimes \mathbf{Y}, \mathbf{Y}, \mathbf{t}) = \mathbf{0} \tag{11}$$

which is symbolically denoted as $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$. The direct solution of $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$ when $p$ is large is in general computationally inefficient as the matrix $S$ is dense. Instead, in the KDC methods, a provisional solution $\widetilde{\mathbf{Y}} = [\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_p]^T$ is assumed, and the discretized error vector is defined as $\boldsymbol{\delta} = \mathbf{Y} - \widetilde{\mathbf{Y}} = [\delta_1, \delta_2, \ldots, \delta_p]^{\mathbf{T}}$. A simple substitution yields an error equation for $\boldsymbol{\delta}$

$$\vec{F}(\vec{y_0} + \Delta t S \otimes \widetilde{\mathbf{Y}} + \Delta t S \otimes \boldsymbol{\delta}, \widetilde{\mathbf{Y}} + \boldsymbol{\delta}, \mathbf{t}) = \mathbf{0}. \tag{12}$$

Following the strategy for SDC methods for ODEs, a low-order time-stepping procedure can be applied to Eq. (12) to yield an approximation $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}_1, \tilde{\delta}_2, \ldots, \tilde{\delta}_p]^T$ to the error. This is equivalent to solving

$$\vec{F}(\vec{y_0} + \Delta t S \otimes \widetilde{\mathbf{Y}} + \Delta t \widetilde{S} \otimes \tilde{\boldsymbol{\delta}}, \widetilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}, \mathbf{t}) = \mathbf{0}, \tag{13}$$

where $\widetilde{S}$ is a low-order, lower-triangular approximation of the spectral integration matrix $S$. One such SDC correction procedure can be considered as an "implicit" function

$$\tilde{\boldsymbol{\delta}} = \widetilde{\mathbf{H}}(\widetilde{\mathbf{Y}}) \tag{14}$$

where the provisional solution $\widetilde{\mathbf{Y}}$ is the input variable and the output is $\tilde{\boldsymbol{\delta}}$. As a reminder, the evaluation of $\widetilde{\mathbf{H}}$ is nothing more than one iteration of the SDC procedure. Notice that when $\tilde{\boldsymbol{\delta}} = \mathbf{0}$, the solution to Eq. (13) is identical to that of Eq. (11), i.e. solving the collocation formulation $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$ is equivalent to finding the zero of the implicit equation $\widetilde{\mathbf{H}}(\mathbf{Y}) = \mathbf{0}$. Therefore, instead of simply accepting the Neumann series solution using the original SDC methods, the Newton–Krylov subspace methods can be applied to solve $\widetilde{\mathbf{H}}(\mathbf{Y}) = \mathbf{0}$, and the resulting algorithm is referred to as the Krylov deferred correction (KDC) method.

Because the low-order method solves an approximation of the collocation formulation, it is not surprising that the explicit function $\widetilde{\mathbf{H}}(\mathbf{Y}) = \mathbf{0}$ is better conditioned compared with the original collocation formulation in (11) as shown by the Jacobian matrix $J_{\widetilde{\mathbf{H}}}$ of $\widetilde{\mathbf{H}}$

$$J_{\widetilde{\mathbf{H}}} = -\left(\frac{\partial \vec{F}}{\partial \mathbf{Y}} + \frac{\partial \vec{F}}{\partial \mathbf{y}} \Delta t \tilde{S}\right)^{-1} \left(\frac{\partial \vec{F}}{\partial \mathbf{Y}} + \frac{\partial \vec{F}}{\partial \mathbf{y}} \Delta t S\right) = -I + C \tag{15}$$

When $\frac{\partial \vec{F}}{\partial \mathbf{Y}}$ is non-singular $\left(\text{e.g. } \frac{\partial \vec{F}}{\partial \mathbf{Y}} = I \text{ for ODE systems}\right)$, since $\tilde{S}$ is an approximation of $S$ and $\Delta t$ is small, $J_{\widetilde{\mathbf{H}}}$ is close to $-I$. This is the first requirement for the efficient application of Newton–Krylov methods. For comparison, the Jacobian matrix of $\mathbf{H} = \mathbf{0}$ is given by $J_{\mathbf{H}} = \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \left(\frac{\partial \vec{F}}{\partial \mathbf{Y}} + \frac{\partial \vec{F}}{\partial \mathbf{y}} \Delta t S\right)$. Note that when any eigenvalue $\lambda$ of the matrix $C$ satisfies $\|\lambda\| \geqslant 1$, the original SDC methods (consider the Neumann series for linear problems) become divergent, on the other hand, the Newton–Krylov methods converge efficiently as long as the number of such eigenvalues is small. We recall that the second requirement for the efficient application of Newton–Krylov methods is an efficient procedure for computing the function $\widetilde{\mathbf{H}}$. As noted earlier, this is simply one sweep of SDC correction described succinctly in Eq. (13).

### 2.2. Fast elliptic solvers

Accurate and efficient numerical schemes for boundary value elliptic equations are an active area of research. In the last twenty years, great progress has been made, examples include the fast algorithms accelerated integral equation methods (IEMs) for ODE boundary value problems of the form

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x)$$

for which a robust, adaptive, and extremely efficient algorithm was presented in 1997 by Greengard and Lee [20]. In their algorithm, the solution is first represented as $u(x) = u_h(x) + u_i(x)$ where $u_i$ is a simple linear function and $u_h$ satisfies the homogeneous boundary conditions. Next, $u_h$ is represented as the convolution of a given Green's function $G(x,t)$ (e.g. the Green's function for $\phi''(x) + q_0(x)\phi(x) = 0$ with homogeneous boundary conditions with a given $q_0$) with an unknown density function $\sigma(x)$ as $u(x) = \int_a^c G(x,t) \cdot \sigma(t)\mathrm{d}t$. A well-conditioned second kind integral equation is then derived for $\sigma(x)$

$$\sigma(x) + \tilde{p}(x) \int_a^c \frac{\mathrm{d}}{\mathrm{d}x} G(x,t)\sigma(t)\,\mathrm{d}t + \tilde{q}(x) \int_a^c G(x,t) \cdot \sigma(t)\,\mathrm{d}t = \tilde{f}(x)$$

where $\tilde{p}$, $\tilde{q}$, and $\tilde{f}$ are explicitly determined by $p(x)$, $q(x)$, $q_0(x)$ and $u_i(x)$. This integral equation is then "restricted" to each subinterval in an adaptive tree structure to resolve the solution, and the restricted local problem is solved efficiently using spectral methods. The local solutions are finally patched together by utilizing a remarkable relation between different intervals, which can be solved directly and efficiently. Instead of the implementation details of this method, which can be found in [20], in the following we list several remarkable features of the solver: (a) the method is direct and very robust. The adaptive strategy requires no *a priori* information of the solution, and the solution is resolved to a specified accuracy; (b) the method is extremely efficient. With $N$ grid points in a given mesh structure, the number of operations is asymptotically optimal $O(N)$ with a small prefactor; and more remarkably, (c) the adaptive code requires about twice as much as work as a nonadaptive code that is simply given the final resolved mesh structure as input. In higher dimensions, fast algorithm accelerated IEMs have also been successfully developed especially for constant coefficient elliptic equations, including the Laplace, Poisson, Yukawa, Helmholtz, Stokes, and biharmonic equations [3,4,6,9,10,26,21].

In addition to IEM-based schemes, spectral, multi-wavelets, higher order finite element and finite difference methods can be applied to solve the elliptic type equations efficiently. In our numerical experiments, the algorithm in [20] and fast spectral methods are used to solve the ODE two-point boundary value problems, and multi-wavelets [7], FMM based and spectral methods have been applied to higher dimensional elliptic equations.

## 3. KDC Accelerated MoL$^T$

In this section, we discuss the implementation details of the KDC accelerated MoL$^T$.

### 3.1. Spectral integration and collocation formulation

To march the evolutionary PDE from $t = 0$ to $\Delta t$, similar to the KDC methods, the MoL$^T$ first discretize the PDE in the temporal direction (the transpose direction of traditional MoL) using $p$ nodes $\mathbf{t} = [t_1, t_2, \ldots, t_p]^T$. Instead of uniform node points on which higher order interpolation is unstable, the MoL$^T$ uses Gaussian type nodes (including Gaussian, Radau, and Lobatto quadrature nodes) and the corresponding Legendre polynomial interpolation. More specifically, the Radau Ia quadrature nodes use the left end point (i.e. $t_1 = 0$), the Radau IIa nodes use the right end point (i.e. $t_p = \Delta t$), and the Lobatto quadrature nodes include both end points. Also, Chebyshev polynomials and the corresponding quadrature nodes may be used instead of Legendre polynomial based nodes, which allows the fast Fourier transform (FFT) to be used for acceleration. A comparison of the Chebyshev and Legendre polynomials based quadratures can be found in a recent paper by Trefethen [25].

On each node point, to avoid the numerically unstable differentiation operator, we define $U_i(x) = U(x, t_i) = u_t(x, t_i)$ as the new unknowns, and recover $u_i(x) = u(x, t_i)$ by integrating the interpolating Legendre polynomial (the coefficients are computed using the corresponding quadrature rules). The linear mapping from $\{U_i(x), i = 1, \ldots, p\}$ to $\{u_i(x), i = 1, \ldots, p\}$ is referred to as "spectral integration", and the corresponding matrix denoted by $S$ is the spectral integration matrix (see [8]). The discretized equation becomes

$$\mathbf{L}\left(\mathbf{U}, \mathbf{u_0} + \Delta t S \otimes \mathbf{U}, \frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{u_0} + \Delta t S \otimes \mathbf{U}), \frac{\mathrm{d}^2}{\mathrm{d}x^2}(\mathbf{u_0} + \Delta t S \otimes \mathbf{U})\right) = \mathbf{0} \tag{16}$$

where $\mathbf{U} = [U_1(x), U_2(x), \ldots, U_p(x)]^T$ is the desired approximation of $U(x, t)$ at different node points and $\mathbf{u_0} = [u(x, 0), u(x, 0), \ldots, u(x, 0)]^T$ is the initial condition. As Gaussian type nodes are used in the MoL$^T$, this collocation formulation can be spectrally accurate for non-degenerating parabolic equations.

### 3.2. Error equation and spectral deferred correction

Notice that the discretized collocation formulation in Eq. (16) consists of a group of elliptic equations coupled by the dense spectral integration matrix $S$. Its direct solution is in general computationally inefficient. Instead, similar to the KDC methods, we assume a provisional solution $\widetilde{\mathbf{U}} = [\widetilde{U}_1(x), \widetilde{U}_2(x), \ldots, \widetilde{U}_p(x)]^T$ is given, and define the error as $\boldsymbol{\delta} = \mathbf{U} - \widetilde{\mathbf{U}} = [\delta_1, \delta_2, \ldots, \delta_p]^T$. The continuous counterparts of $\boldsymbol{\delta}$ and $\widetilde{\mathbf{U}}$ derived by polynomial interpolation are denoted as $\delta(x, t)$ and $\widetilde{U}(x, t)$, respectively. A simple substitution yields an equation for $\delta(x, t)$ given by

$$\mathcal{L}\left(\widetilde{U} + \delta, u_0 + \int_0^t (\widetilde{U} + \delta)\mathrm{d}\tau, \frac{\mathrm{d}\left(u_0 + \int_0^t (\widetilde{U} + \delta)\mathrm{d}\tau\right)}{\mathrm{d}x}, \frac{\mathrm{d}^2(u_0 + \int_0^t (\widetilde{U} + \delta)\mathrm{d}\tau)}{\mathrm{d}x^2}\right) = 0 \tag{17}$$

where $u_0 = u(x, 0)$, and the variables in $\widetilde{U}(x, t)$ and $\delta(x, t)$ are omitted in the notation. Applying spectral integration to the integrals in this equation, the discretized "error equation" is given by

$$\mathbf{L}\left(\widetilde{\mathbf{U}} + \boldsymbol{\delta}, \mathbf{u_0} + \Delta t S \otimes (\widetilde{\mathbf{U}} + \boldsymbol{\delta}), \frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{u_0} + \Delta t S \otimes (\widetilde{\mathbf{U}} + \boldsymbol{\delta})), \frac{\mathrm{d}^2}{\mathrm{d}x^2}(\mathbf{u_0} + \Delta t S \otimes (\widetilde{\mathbf{U}} + \boldsymbol{\delta}))\right) = \mathbf{0} \tag{18}$$

where $\Delta t S$ is the spectral integration approximation of the operator $\int_0^t$.

Notice that $\int_0^t \delta(\tau)\mathrm{d}\tau (\delta(0) = 0)$ can also be approximated by low-order integration rules. For the rectangular rule using left end point (explicit Euler method), the corresponding integration matrix is given by

$$\Delta t\widetilde{S} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \Delta t_2 & 0 & \cdots & 0 & 0 \\ \Delta t_2 & \Delta t_3 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ \Delta t_2 & \Delta t_3 & \cdots & \Delta t_p & 0 \end{bmatrix} \tag{19}$$

and for the rectangular rule using right end point (implicit Euler method),

$$\Delta t\widetilde{S} = \begin{bmatrix} \Delta t_1 & 0 & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & \Delta t_{p-1} & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & \Delta t_{p-1} & \Delta t_p \end{bmatrix}. \tag{20}$$

A low-order approximation $\bar{\boldsymbol{\delta}}$ of $\boldsymbol{\delta}$ can then be obtained by solving the system

$$\mathbf{L}\left(\widetilde{\mathbf{U}} + \bar{\boldsymbol{\delta}}, \mathbf{u_0} + \Delta tS \otimes \widetilde{\mathbf{U}} + \Delta t\widetilde{S} \otimes \bar{\boldsymbol{\delta}}, \frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{u_0} + \Delta tS \otimes \widetilde{\mathbf{U}} + \Delta t\widetilde{S} \otimes \bar{\boldsymbol{\delta}}), \frac{\mathrm{d}^2}{\mathrm{d}x^2}(\mathbf{u_0} + \Delta tS \otimes \widetilde{\mathbf{U}} + \Delta t\widetilde{S} \otimes \bar{\boldsymbol{\delta}})\right) = \mathbf{0} \tag{21}$$

which is equivalent to a low-order time marching scheme where the equations for $\bar{\boldsymbol{\delta}}$ are decoupled at different times. The elliptic equation at each node $t_i$ can then be solved efficiently using available elliptic equation solvers as discussed in Section 2.2.

### 3.3. Newton–Krylov method acceleration

Notice that Eq. (21) can be considered as an "implicit function" where $\widetilde{\mathbf{U}}$ is the input variable and $\bar{\boldsymbol{\delta}}$ the output function value. We symbolically denote the explicit form of this function as

$$\bar{\boldsymbol{\delta}} = \widetilde{\mathbf{H}}(\widetilde{\mathbf{U}}). \tag{22}$$

When $\widetilde{\mathbf{U}}$ solves the collocation formulation in Eq. (16), it can be seen that $\bar{\boldsymbol{\delta}} = \mathbf{0}$. Therefore, solving the collocation formulation is equivalent to solving

$$\widetilde{\mathbf{H}}(\widetilde{\mathbf{U}}) = \mathbf{0}. \tag{23}$$

Compared with the original collocation formulation, the "preconditioned" formulation in Eq. (23) is better conditioned. Similar to Eq. (15), it is straight forward to show that the Jacobian matrix of the function $\widetilde{\mathbf{H}}(\widetilde{\mathbf{U}})$ is closer to $\pm I$ and we neglect the details. Hence the Newton–Krylov methods can be applied to the preconditioned system, in which each function evaluation of $\widetilde{\mathbf{H}}(\widetilde{\mathbf{U}})$ is simply the low-order time-stepping results in Eq. (21), and the elliptic equation at each time step can be solved efficiently using available fast elliptic equation solvers. We refer to this approach as the KDC accelerated MoL$^T$.

### 3.4. Algorithm properties

In this section, we discuss several analytical results concerning the accuracy, stability, efficiency, and orders of the new approach.

The accuracy of the KDC accelerated MoL$^T$ for each time step is determined by (a) the accuracy of the collocation formulation in Eq. (16); (b) the accuracy of the Newton–Krylov methods, and (c) the accuracy of the fast elliptic equation solvers. For (a), as the Gaussian type nodes are used, it is not hard to see that the resulting formulation is spectrally accurate, e.g. for fixed step-size, the numerical error decays exponentially when the number of nodes increases, and for prescribed accuracy requirement, when more node points are used, extremely large time step-sizes can be used. These are very important features for *long-time* simulations. The convergence and accuracy of the Newton–Krylov methods in (b) have been widely studied previ-

ously. It can be shown that super-linear local convergence can be obtained for specially chosen parameters in the Newton–Krylov schemes (see Theorem 6.1.2 in [17]). For arbitrary initial approximations, continuation/homotopy methods may become necessary to accomplish global convergence. Interested readers are referred to [16,17] for further discussions. For (c), instead of the "order of convergence" concepts commonly used in the finite difference and finite element methods, most recently developed fast IEM solvers generate numerical results with prescribed accurate digits, therefore the accuracy in (c) is guaranteed.

In [12], it was shown that for ODE problems, the collocation formulation is equivalent to certain Runge–Kutta methods (see p. 27 in [12]). When Gaussian or Radau IIa node points are used in the temporal discretization, the method is *A*-stable and *B*-stable [2,12]. Based on these existing results and assume the spatial elliptic equations are solved exactly, we conclude that the KDC accelerated MoL$^T$ is unconditionally stable. In our numerical experiments, no CFL constraints have been observed, and *optimal* step-size can be used. The efficiency of the KDC accelerated MoL$^T$ is determined by the number of iterations in the Newton–Krylov procedure which depends on the structure of the problem, low-order preconditioners (Euler, trapezoidal rule, splitting methods, semi-implicit approaches, etc.), and Krylov subspace methods (GMRES, BiCGStab, TFQMR, etc.). Currently, detailed comparisons of different strategies and acceleration techniques for optimal efficiency are being performed.

Finally, it is well-known that for ODE problems, when $p$ Gaussian nodes are used and the collocation formulation is solved exactly, the resulting scheme has order $2p$. When Radau nodes are used, the scheme has order $2p-1$ [11,13]. For stiff ODE systems and DAEs, however, order reductions of collocation methods (these reductions are different from the ones introduced by Neumann series expansion) have been observed as described in [11,13]: for stiff problems, the Gaussian and Radau IIa based methods are *B-convergent* with order $p$ (see p. 247 [13]); for index 2 DAEs, the order of the algebraic components is only $p$ or less (see p. 18, Table 2.3 [11]). For parabolic PDEs, our preliminary analyses show that the order of the method depends on the boundary conditions (periodic, Dirichlet, Neumann conditions for $u$ or $u_t$), step-size (as our schemes allow very large time steps, the *B*-convergence concept has to be introduced), and existence of any additional algebraic constraints. These will be briefly illustrated by several numerical experiments in the following section. Detailed order analysis for PDE systems is currently being performed and will be reported later.

## 4. Numerical experiments

In this section, we show several preliminary numerical results. The new methods are implemented in *Matlab* and *Fortran*. Both Radau IIa and Gaussian nodes have been tested for temporal discretizations.

### 4.1. The diffusion equation in 1-D

To illustrate the basic properties of the KDC accelerated MoL$^T$ methods, we first consider the homogeneous heat equation

$$\frac{\partial \psi}{\partial t} = \frac{\partial^2 \psi}{\partial x^2}$$

with exact solution $\psi = \cos(x)\mathrm{e}^{-t}$. We assume periodic boundary conditions on $[0, 2\pi]$ and compute the initial condition at $t = 0$ accordingly. For this equation, as the decoupled elliptic equations can be solved exactly, we focus on the errors introduced by the temporal discretization, and study the order of convergence in our new approach.

In Fig. 1, the max error at time $t = 10$ is plotted as a function of step-size. In the left plot, we use six Radau IIa points with full GMRES iteration. Numerical results show that the error quickly converges to machine precision with approximate order 11.1. Similarly in the right plot, results using six Gaussian nodes are presented showing approximate order 12.3. These agree with the theoretical results (under certain assumptions) that the collocation formulation is order $2p-1$ for $p$ Radau IIa points, and order $2p$ for $p$ Gaussian nodes.

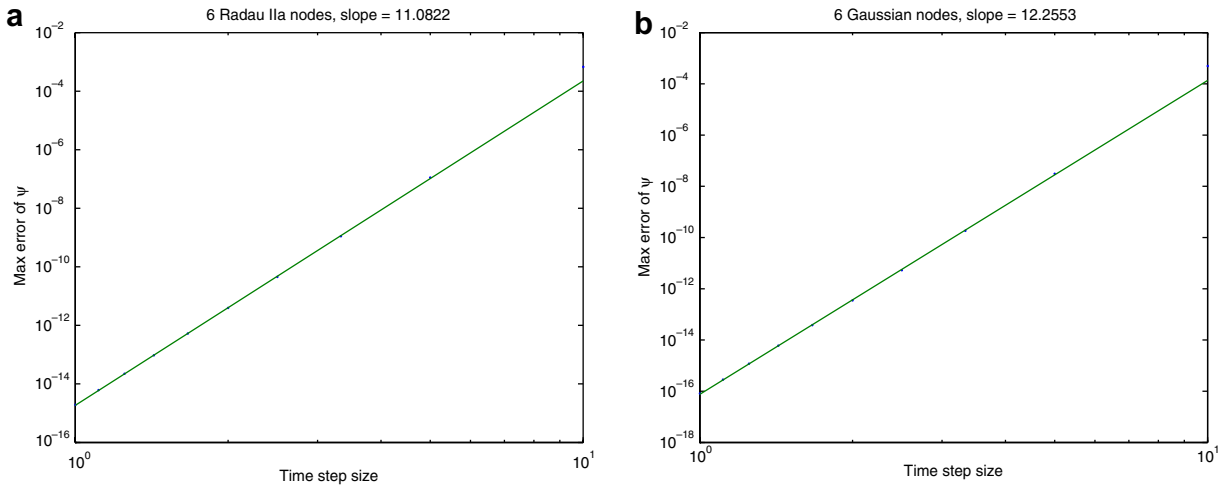


Fig. 1. Convergence of the KDC Accelerated MoL$^T$ with full GMRES using Radau IIa (left) and Gaussian (right) nodes.

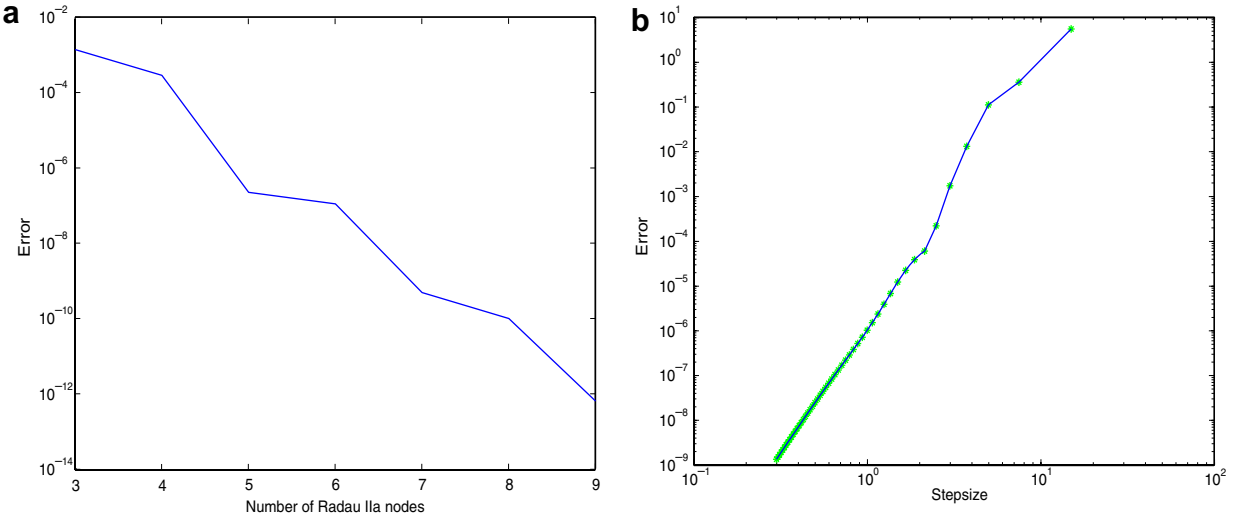


Fig. 2. Numerical error as a function of (a) number of Radau IIa nodes and (b) step-size.

We also consider the inhomogeneous diffusion equation

$$\frac{\partial \psi}{\partial t} = \frac{\partial^2 \psi}{\partial x^2} + f(x,t)$$

with exact solution $\psi = \cos(x)\sin(t)$ on $[-1,1]$, and Dirichlet boundary conditions and $f(x,t)$ are specified accordingly. The elliptic equation solver used in this example is based on the spectral integration method introduced by Greengard in [8]. In (a) of Fig. 2, we march from $t = 0$ to $t = 10$ using six uniform steps, and plot the max error as a function of the number of Radau IIa collocation points. It can be seen that the error quickly decreases to machines precision when the number of nodes increases. In (b), we fix the number of Radau IIa nodes to $p = 4$, and plot how the error decays as a function of step-size. In this experiment, order reduction has been observed, we believe this is due to (a) the boundary conditions which serve as algebraic constraints in the formulation; and (b) large step-sizes so $B$-convergence has to be discussed. This order reduction is being studied and detailed analysis will be reported later.

### 4.2. Variable coefficient parabolic problem

For our next example, we consider a parabolic system on $x \in [0,1]$ and $t \in [0,T]$ given by

$$M(x)\frac{\partial \psi}{\partial t} = N(x)\frac{\partial^2 \psi}{\partial x^2} + f(x,t)$$

This equation represents a class of parabolic systems including the linearized Richards' equation in porous medium and the variable coefficient diffusion system

$$M(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial x}\left(K(\psi)\frac{\partial \psi}{\partial x}\right)$$

We use the exact solution

$$\psi(x,t) = \mathrm{e}^{\cos(2\pi(x+t^2))-kt}$$

and define

$$M(x) = 2+\cos(2\pi x), \quad N(x) = 2+\cos(4\pi x).$$

Periodic boundary conditions are imposed and we set $\psi_0(x) = \mathrm{e}^{\cos(2\pi x)}$ at $t = 0$ as the initial condition. The source term $f(x,t)$ is determined accordingly. In the simulation, as the solution is periodic, the spatial domain equations can be solved using Fourier series based spectral methods. Similar to the homogeneous heat equation case, our numerical experiments show that the KDC accelerated MoL$^T$ approach converges with desired order, and we neglect the details.

Previous research shows that when GMRES is applied in the Newton–Krylov method, the memory required increases linearly with the iteration number $k$, and the number of multiplications scales like $\frac{1}{2}k^2N$. Hence, for large $k$, the GMRES procedure becomes very expensive and requires excessive memory storage. For these reasons, instead of a full orthogonalization procedure, GMRES can be restarted every $k_0$ steps where $k_0 < N$ is some fixed integer parameter. The restarted version is often denoted as GMRES($k_0$). Alternative options include the biconjugate gradients stabilized (BiCGStab) method and transpose-free quasi-minimal residual (TFQMR) algorithm (See [1] for a summary of existing Newton–Krylov methods). The storage required in these methods is independent of the iteration number $k$, and the number of multiplications grows only linearly as a function of $k$. We next study the effect of using different Krylov subspace methods on the efficiency of our approach. In Fig. 3, we show how the errors decay with increased number of iterations for different Krylov subspace methods. In the calculation, we use 32 grid points in space and 26 Radau IIa nodes in time to fully resolve the solution to machine precision within one time step from $t = 0$ to $t = 0.5$.

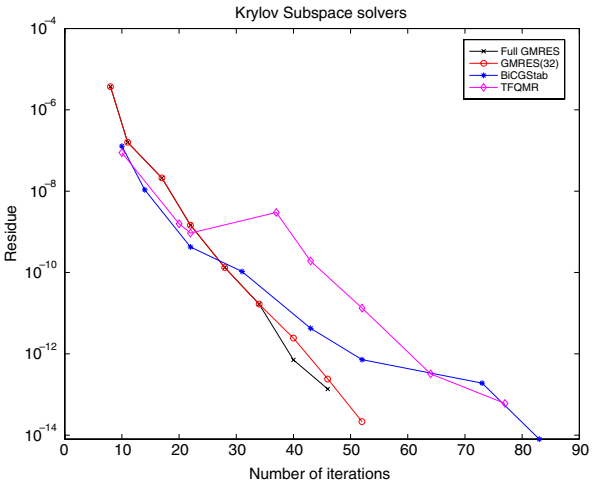


Fig. 3. Comparison of different Krylov subspace solvers.

It can be seen that alternative approaches perform similarly to the full GMRES scheme in efficiency for this specific example, while requiring much less memory storage.

### 4.3. Schrödinger equation

In this example, we consider the linear time dependent Schrödinger equation in one-dimensional space

$$\psi_t(x,t) = -\mathrm{i}H(x,t)\psi(x,t) \tag{24}$$

where $H(x,t) = -\frac{1}{2}\nabla^2 + V(x,t)$ is the Hamiltonian and the potential term is given by $V(x,t) = -\mathrm{e}^{-0.1(x-0.1t)^2}$. We set the initial condition to $\psi(x,0) = \mathrm{e}^{\mathrm{i}xv}\psi_0(x)$, where $\psi_0(x)$ is an eigen-function of $H$. The exact solution at time $t$ can be analytically computed as $\mathrm{e}^{\mathrm{i}xv - i(E+v^2/2)t}\psi_0(x-vt)$, with $E$ the eigenvalue corresponding to $\psi_0$.

In our simulation, as the solution decays rapidly for large $x$, periodic boundary conditions on $[-L,L]$ with $L = 30$ can be imposed. We use the Fourier series expansion with 128 sampling points in space, and 3 Gaussian type nodes in time to march the solution from time $t = 0$ to $t = 2.5$. The error tolerance is set to $1e-10$ when solving the collocation formulations. The orders of convergence are presented for Gaussian nodes in (a) (approximate order 6) and Radau IIa nodes in (b) (approximate order 5) of Fig. 4, respectively.

In [12], it was shown that when Gaussian nodes are used, the resulting collocation formulation better preserves the Hamiltonian structure of the original PDEs, and hence the method is called "symplectic." In (a) of Fig. 5, we can see that even though the solution error is large, the energy is conserved when symplectic schemes are used, while both solution and energy errors increase rapidly in (b) when a non-symplectic scheme using Radau IIa nodes is applied. Therefore, when the energy conservation is critical in a simulation (e.g. for stability considerations), Gaussian nodes are recommended. However for general partial differential algebraic systems, severe order reduction may be observed for Gaussian nodes, and Radau IIa nodes are recommended instead, as discussed in [11].

### 4.4. A nonlinear parabolic equation

In this example, we consider a nonlinear diffusion equation of the form

$$\psi_t = (\psi^{6/5} + 1)\psi_{xx} + 2\psi\psi_x(\psi_x + 1) + f$$

with Dirichlet boundary conditions, whose analytical solution is given by

$$\psi(x,t) = \mathrm{e}^{x^2 + \sin t}$$

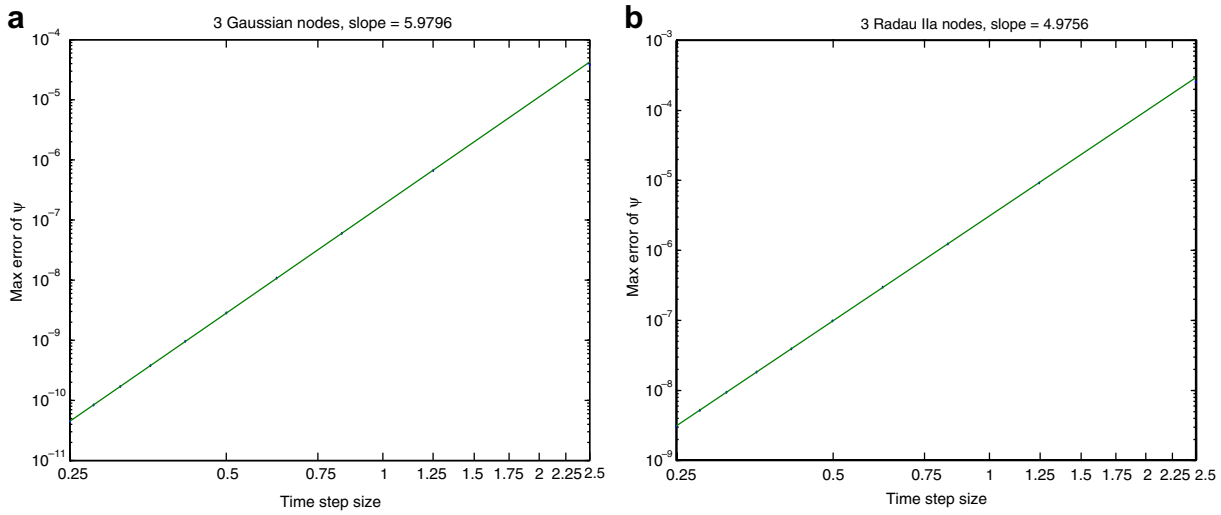


Fig. 4. Convergence Order using Gaussian (left) and Radau IIa (right) nodes.

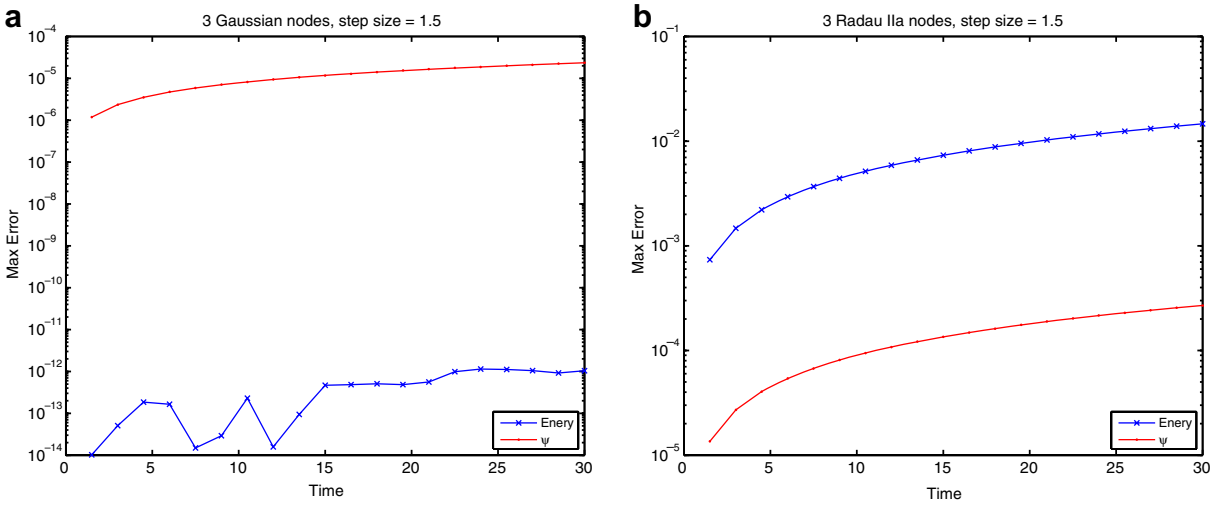


Fig. 5. Energy conservation for Gaussian (left) and Radau IIa (right) nodes.

In the numerical simulation, we apply the Jacobian-free Newton–Krylov technique in which the matrix vector product is approximated by the forward difference approximation (see Eq. (6)). The decoupled nonlinear elliptic equation in every step of the low-order time-stepping method required by each function evaluation is solved by a nonlinear version, two-point boundary value problem solver developed by Greengard and Lee [20] briefly introduced in Section 2.2.

The simulations run from $t = 0$ to $t = 5$ using Radau IIa nodes for each time step and the error tolerance is set to $1e - 11$. The spatial domain $[-1, 1]$ is divided into 20 equi-spaced subintervals and a 10th order polynomial is used inside each subinterval to resolve the solution. In the Newton–Krylov methods, we terminate the Krylov subspace iterations as long as the residual is reduced by a certain factor (determined dynamically, see [16,17]) or a prescribed number of iterations is reached. As shown in Fig. 6, using time step-size $\Delta t = 1$, the global error at $t = 5$ decays very rapidly as the number of Radau IIa nodes increases.

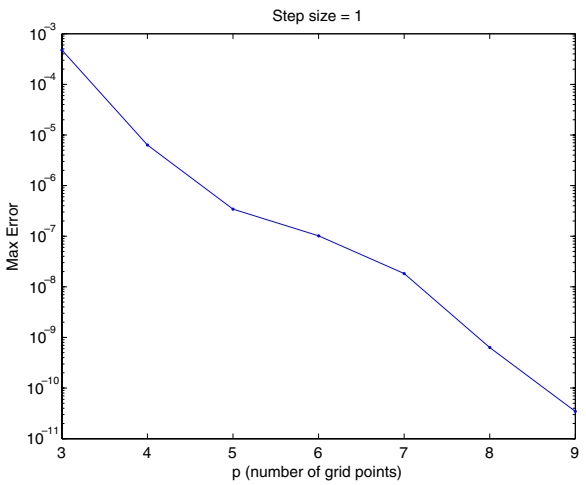


Fig. 6. Numerical error as a function of number of grid points.

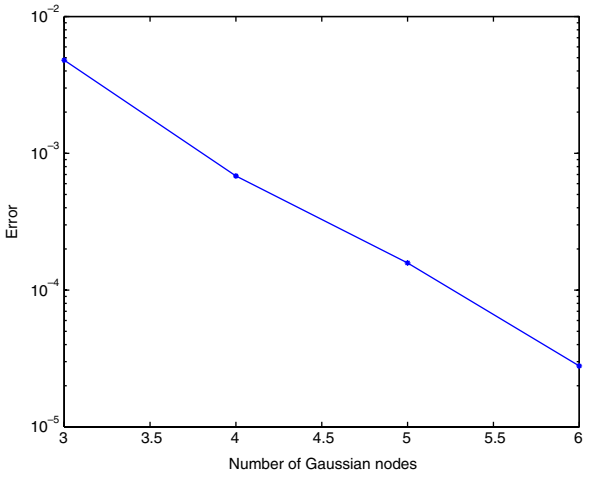


Fig. 7. Numerical error as a function of number of nodes.

### 4.5. Diffusion equation in 3 + 1 dimensions

Finally in this section, we consider the three-dimensional diffusion equation

$$\frac{\partial \psi(\vec{x},t)}{\partial t} = \nabla^2 \psi(\vec{x},t) + f(\vec{x},t)$$

where $\vec{x} \in \mathbb{R}^3$, $t \in [0,T]$, and asymptotic boundary conditions are imposed at far field. We define the analytical solution $\psi$ as

$$\psi(\vec{x},t) = \mathrm{e}^{-|\vec{x}|^2} \sin(t)$$

and compute the source term $f$ accordingly.

Our simulations run from time $t = 0$ to $T = 4$ for one time step using 3 – 6 Gaussian nodes. The computational domain is set to $[-6,6]^3$, so that the real solution and its derivatives (up to second order) vanish to about $1e-11$ on the boundaries. The multi-wavelets method-based package MADNESS [7] is applied for the efficient solution of the three-dimensional elliptic equations. We use nine multi-wavelets of degrees 0–8 in each spatial dimension and set the threshold to $1e-9$. In Fig. 7, we plot the numerical errors as a function of the number of Gaussian nodes. Similar to previous results, the error rapidly decays to prescribed accuracy as the number of Gaussian nodes increases.

## 5. Concluding remarks

In this paper, a new framework for the efficient and accurate solutions of time dependent parabolic equations is presented. Our preliminary analysis and numerical experiments show that the KDC accelerated MoL$^T$ methods are very promising for accurate and efficient long-time large-scale simulations. However, further analyses and optimized implementation are required in order to fully explore the efficiency and accuracy of the new methods. These include the detailed convergence order analyses of the methods (in particular, the order reductions related with boundary conditions), optimized strategies for adaptive mesh refinements, order selections in both time and space, proper Newton–Krylov methods, simplified Newton's method, proper treatment of the differential and algebraic components in the equations, as well as the optimal choice of many different parameters. Also, this new technique can be generalized to problems with multiple temporal and spatial scales, and to other type PDEs when the solution is smooth so higher order methods are advantageous. Results along these lines will be presented in the future.

## Acknowledgement

Finally, we would like to express our gratitude to Prof. Carl T. Kelly of the North Carolina State University for discussions of the inexact Newton methods, to Professor June-Yub Lee from Ewha Women's University of South Korea for providing the state-of-the-art fast adaptive ODE boundary value problem solver, and to Professors Michael Minion and Casey Miller of the University of North Carolina, Drs. George Fann and Robert Harrison of Oak Ridge National Laboratory for many helpful discussions.

## References

[1] R. Barrett et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd ed., SIAM, Philadelphia, 1994.
[2] R. Barrio, On the A-stability of Runge–Kutta collocation methods based on orthogonal polynomials, SIAM J. Numer. Anal. 36 (4) (1999) 1291–1303.
[3] H.W. Cheng et al., A wideband fast multipole method for the Helmholtz equation in three dimensions, J. Comput. Phys. 216 (1) (2006) 300–325.
[4] H.W. Cheng, J.F. Huang, T.J. Leiterman, An adaptive fast solver for the modified Helmholtz equation in two dimensions, J. Comput. Phys. 211 (2) (2006) 616–637.
[5] S. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2) (2000) 241–266.
[6] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM J. Sci. Comput. 23 (3) (2001) 741–760.
[7] G.I. Fann, R.J. Harrison, G. Beylkin, J. Jia, R. Hartman-Baker, W.A. Shelton, S. Sugiki, MADNESS applied to density functional theory in chemistry and nuclear physics, J. Phys.: Conf. Ser. 78 (2007), 2018.
[8] L. Greengard, Spectral integration and two-point boundary value problems, SIAM J. Numer. Anal. 28 (1991) 1071–1080.
[9] L. Greengard, J.Y. Lee, A direct adaptive Poisson solver of arbitrary order accuracy, J. Comput. Phys. 125 (2) (1996) 415–424.
[10] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the laplace equation in three dimensions, Acta Numer 6 (1997) 229–269.
[11] E. Hairer, C. Lubich, M. Roche, The numerical solution of differential-algebraic systems by Runge–Kutta methods, Springer-Verlag, 1989.
[12] E. Hairer, C. Lubich, G. Wanner, Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations, Springer-Verlag, 2002.
[13] E. Hairer, G. Wanner, Solving ordinary differential equations II, Springer, 1996.
[14] J. Huang, J. Jia, M. Minion, Accelerating the convergence of spectral deferred correction methods, J. Comp. Phys. 214 (2) (2006) 633–656.
[15] J. Huang, J. Jia, M. Minion, Arbitrary order Krylov deferred correction methods for differential algebraic equations, J. Comp. Phys. 221 (2) (2007) 739–760.
[16] C.T. Kelly, Solving Nonlinear Equations with Newton's Method, SIAM, 2003.
[17] C.T. Kelly, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[18] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: A survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[19] A.T. Layton, M.L. Minion, Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations, BIT 45 (2) (2005) 341–373.
[20] J. Lee, L. Greengard, A fast adaptive numerical method for stiff two-point boundary value problems, SIAM J. Sci. Comput. 18 (2) (1997) 403–429.
[21] P.G. Martinsson, V. Rokhlin, A fast direct solver for boundary integral equations in two dimensions, J. Comput. Phys. 205 (2005) 1–23.
[22] V. Pereyra, Iterated deferred correction for nonlinear boundary value problems, Numer. Math. 11 (1968) 111–125.
[23] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[24] W.E. Schiesser, The Numerical Method of Lines: Integration of Partial Differential Equations, Academic Press, San Diego, 1991.
[25] L.N. Trefethen, Is Gauss Quadrature Better than Clenshaw–Curtis? SIAM Review, to appear.
[26] L. Ying, G. Biros, D. Zorin, H. Langston, A new parallel kernel-independent fast multiple algorithm, in: Proceedings of SC03, 2003.
[27] P.E. Zadunaisky, A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations, The Theory of Orbits in the Solar System and in Stellar Systems, in: Proceedings of International Astronomical Union, Symposium 25, 1964.
[28] P.E. Zadunaisky, On the estimation of errors propagated in the numerical integration of ordinary differential equations, Numer. Math. 27 (1976) 21–40.