Fast time integration of parabolic equations with variable coefficients

Yoonsang Lee*[1]

[1]Department of Mathematics, Dartmouth College

Last Update : February 19, 2020

Abstract

This paper proposes a fast time integration method for parabolic equations with a variable coefficient. The key idea of the proposed method is to approximate the differential operator using a constant coefficient operator, which provides an efficient mechanism to expedite the calculation of a solution of an algebraic equation in an implicit method. The method does not require to specify the constant as the constant is implicitly incorporated in the method. Without relying on an iterative solver, the computational complexity of the proposed method at each time step remains at $\mathcal{O}(N \log N)$ for an N-dimensional solution vector. We analyze the accuracy and the stability of the new method and discuss its connection with other methods, including multiscale time integrators. The efficiency and the stability of the method are validated through a suite of numerical tests in 1D and 2D with multiscale random coefficients. The method is also applied to solve an elliptic problem and a quasilinear diffusion model in the semiconductor.

1 Introduction

We propose a fast numerical time integration of parabolic equations with a variable coefficient in a domain $\Omega = (0, L)^d \subset \mathbb{R}^d$. In particular, we consider the following diffusion equation of a scalar field $u(t, x) : (0, T) \times \Omega \to \mathbb{R}$ with a variable diffusion coefficient $a(x) : \Omega \to \mathbb{R}^+$ and an external source field $f(x) : \Omega \to \mathbb{R}$,

$$\begin{aligned} u_t &= \nabla \cdot (a(x)\nabla u) + f(x), \quad (t, x) \in (0, T) \times \Omega \\ u(0, x) &= u_0(x) \end{aligned} \tag{1}$$

with an appropriate boundary condition on $\partial\Omega$ that guarantees a unique solution. We assume that the differential operator $\nabla \cdot (a(x)\nabla u)$ is uniformly elliptic with $a(x) \geq \alpha > 0$, and the external source f is bounded by 1 in the $L^2(\Omega)$ norm, that is, $\|f\|_2 \leq 1$. The proposed fast integrator addresses the issues related to the variable coefficient when the coefficient $a(x)$ contains fine scales. The method achieves a computational complexity $O\mathcal{O}(N \log N)$ at each time for an N-dimensional

*yoonsang.lee@dartmouth.edu

solution vector. The parabolic equation with a variable coefficient describes many problems in geoscience, chemistry, engineering, and manufacturing. Applications include diffusions in multiphase solids or heterogeneous media such as rocks, soil, and plants [22]. Due to the improvement in the acquisition and observation of material information in recent years [21], an accurate modeling of the diffusion coefficient contains a wide range of scales that requires a fine mesh to resolve the variation of the coefficient.

In the method of lines (MOL) approach that discretizes the PDE in space and then integrates in time for a finite-dimensional solution vector, the parabolic equation is a well-known model of stiff problems with large negative eigenvalues for which a time step of certain numerical methods is restricted by stability. Explicit integrators typically have a relatively small stability domain that imposes a constraint on the time step and the spatial discretization mesh. For the variable coefficient case, the spatial resolution is forced to be small to resolve the variation of the coefficient to achieve a certain accuracy. If $\epsilon \ll 1$ represents the finest scale of $a(x)$, the spatial resolution is required to be small enough to resolve ϵ. In this case, as the stability constraint on the time step is proportional to the power of the spatial mesh size, the time step size becomes unrealistically small to satisfy stability. Exponential integrators [8] overcome the stability issue of the explicit integration through an analytic integration of the stiff problem (see [4] for an application of exponential integrators for variable coefficient parabolic equations). The computational bottleneck of this approach is the matrix exponentiation, which costs at least $\mathcal{O}(N^2)$, where N is the size of the discretized solution vector [16]. The method can be more expensive if the coefficient changes in time or depends on the solution as the matrix exponentiation must be calculated at each time step.

Implicit methods, on the other hand, have a relatively large stability domain and thus enable to use a large time step. One issue with implicit methods is that they require a solution of an algebraic equation at each integration step. For the model problem we consider here, where the algebraic equation is related to a symmetric positive definite matrix, there are fast iterative solvers to solve the algebraic equation, including conjugate gradient methods, possibly with preconditioning [20, 18]. Our proposed method aims at a fast method that does not require an iterative solver while maintaining a large time step.

Several fast and stable integration methods are available when there is a special structure in the coefficient. For a constant coefficient $a(x) = a > 0$, a variant of the fast Poisson solver [5, 17] can solve the algebraic equation in implicit methods with $\mathcal{O}(N \log N)$ complexity. Another nontrivial example includes the homogenization theory [3]. If there is scale separation between the finest scale ϵ of $a(x)$ and the macroscopic scale of interest, the homogenization theory provides a robust coarse-scale model through a homogenized coefficient, which allows to use a coarse space mesh and a large time step accordingly without losing stability. An issue of homogenization-based methods is its dependence on the scale separation assumption and thus the difficulty in calculating the homogenized coefficient for non-separable scales.

Our proposed method aims at a large time step integration without losing stability when the spatial variation is resolved using a fine mesh. Resolving the spatial resolution is particularly important when there is no scale separation in the model, and fine resolution details are of interest. The proposed method has a stability property comparable to the implicit theta method with a low computational cost. The cost is comparable to the fast methods for a constant coefficient case that uses the known eigenvalues and eigenvectors of the operator. Particularly for the periodic boundary condition, the proposed method becomes an explicit method using a variable time step that depends on wavenumbers. The key idea of the proposed method is to mix the variable coefficient differential operator and a constant coefficient differential operator for efficiency where the con-

stant coefficient operator is used for inversion. This approach is different from the homogenized approach that requires to specify the unknown constant coefficient. The proposed method does not specify the unknown constant coefficient; the unknown coefficient is implicitly incorporated in the method.

This paper has the following structure. Section 2 reviews related methods, a multiscale time integrator for scale-separated ODE problems, and an implicit time integration, the theta method. These methods provide central ideas of the proposed method for fast integration and stability. The proposed method is explained in Section 3, and we consider specific characteristics for various boundary conditions. We also analyze the accuracy and stability of the proposed method. In Section 4, we apply the proposed method to several test problems in 1D and 2D with multiscale variable coefficients, including a quasilinear diffusion problem. We conclude in Section 5 with discussions of future extensions and limitations of the proposed method.

2 Related methods

In this section, we review two methods that provide central ideas of the proposed method, i) a multiscale time integrator using variable time steps [13] and ii) the theta method [19, 9]. The multiscale method shows that an explicit method with variable time stepping can have improved stability as some component of the system has an effect of an implicit integration. This observation suggests that a modification of an implicit method can improve the efficiency of the method.

2.1 A multiscale time integrator using variable time steps

For the past decade, multiscale time integrators ([7, 6, 2] and references therein) have been actively investigated as a method to effectively capture the solution of multiscale dynamical systems without losing stability and accuracy. One of the standard models for multiscale time integrators has the following representation

$$\begin{aligned} \frac{dX}{dt} &= f_0(X,Y), \quad X(0) = X_0 \\ \frac{dY}{dt} &= \frac{f_1(X,Y)}{\epsilon}, \quad Y(0) = Y_0, \quad 0 < \epsilon \ll 1 \end{aligned} \tag{2}$$

where f_0 and f_1 are of order $\mathcal{O}(1)$ while $\epsilon \ll 1$ represents the short time scale of the system. It is typically assumed that the fast variable Y is ergodic for a fixed X, and multiscale integrators aim for the averaged solution X without resolving all fine details of Y. This class of methods is efficient as the averaged X is resolved using a coarse time step larger than ϵ while Y is resolved using a fine time step only for a short period of time to capture the ergodic behavior of Y.

In [13], it is shown that a time integration using a variable time step can improve the standard multiscale integrators with enhanced efficiency and accuracy. The idea of the variable time step is to use two different time steps, a corse time step $\delta t < \mathcal{O}(1)$ for X and a fine time step $\delta t < \mathcal{O}(\epsilon)$ for Y instead of using the fine step δt for both X and Y. It is analyzed in [13] that the method has the same effect as solving the problem (2) with an effect similar to the theta method in the integration of X (thus, it allows a large time step for stability). As a specific example, if we use the forward Euler for the integration using δt and δt for X and Y respectively, one-step integration of (2) from

$t^n = n\Delta\tau$ to $t^{n+1} = (n+1)\Delta\tau$ has the following equivalent formula

$$\begin{aligned} X^{n+1} &= X^n + \Delta\tau\left(\theta f_0(X^n, Y^{n+1}) + (1-\theta) f_0(X^n, Y^n)\right) \\ Y^{n+1} &= Y^n + \Delta\tau \frac{1}{\epsilon'} f_1(X^n, Y^n) \end{aligned} \tag{3}$$

where $\Delta\tau = \delta t + \delta t$, $\theta = \frac{\delta t}{\delta t + \delta t}$, and $\epsilon' = \frac{\delta t + \delta t}{\delta t}\epsilon > \epsilon$. The idea of using different scale time steps has a natural extension to several different scale problems, i.e., using different scale time steps for each corresponding temporal scale dynamics, and it has been applied to sparse dynamics of Fourier spectral methods ([12] and chapter 5 of [11]).

A relation between the variable step and an implicit method can also be observed for the parabolic equation (1) when the coefficient $a(x)$ is a constant. Let us consider (1) in the 1D case with the periodic boundary condition in $(0, 2\pi)$ and $f = 0$ for simplicity. If the coefficient $a(x)$ is a constant $\alpha > 0$, the Fourier transform $\hat{(\cdot)}$ of the equation yields

$$\hat{u}_t(t,k) = -\alpha k^2 \hat{u}(t,k), \quad k = 0, \pm 1, \pm 2, ... \tag{4}$$

The Crank-Nicolson using a (fixed) time step δt for (4) has the following one-step integration

$$\hat{u}^{n+1} = \hat{u}^n - \frac{\delta t \alpha k^2}{2}\hat{u}^{n+1} - \frac{\delta t \alpha k^2}{2}\hat{u}^n. \tag{5}$$

This equation can be further simplified as

$$\begin{aligned} (1 + \frac{\delta t \alpha k^2}{2})\hat{u}^{n+1} &= (1 + \frac{\delta t \alpha k^2}{2})\hat{u}^n - \delta t \alpha k^2 \hat{u}^n \\ \Rightarrow \hat{u}^{n+1} &= \hat{u}^{n+1} - \frac{\delta t}{(1 + \frac{\delta t \alpha k^2}{2})}\alpha k^2 \hat{u}^n. \end{aligned} \tag{6}$$

A key observation of this formula is that it is actually an explicit integration of (4) using a variable time step $H(k) = \frac{\delta t}{(1 + \frac{\delta t \alpha k^2}{2})}$ that depends on the wavenumber k. As the Crank-Nicolson is A-stable, the explicit integration allows a large time step δt. When the coefficient $a(x)$ is variable, which is of our interest in this paper, the variable time integration for stability (6) is not straightforward as the coefficient and $\hat{u}$ are convoluted, and the time derivative of $\hat{u}$ has all mixed contributions from different wavenumber components. To use the nice property of the constant coefficient case, we incorporate the flexibility of the theta method that allows a range of parameter values for stability.

2.2 The theta method

The theta method [19, 9] is an one-step time integration method that enjoys A-stability for certain parameter values. As an illustration of the method, we consider the method of line approach for the parabolic equation (1) after a spatial discretization

$$\mathbf{u}_t = \mathbf{L}_a \mathbf{u} + \mathbf{f}. \tag{7}$$

Here $\mathbf{L}_a$ is a $N \times N$ matrix approximating the differential operator $\nabla \cdot (a(x)\nabla)$, which is symmetric negative definite, $\mathbf{u}$ and $\mathbf{f}$ are vectors in $\mathbb{R}^N$ approximating u and f respectively. The theta method uses a parameter θ that determines the weights of two right hand side terms, $\mathbf{L}_a\mathbf{u}^{n+1}$ and $\mathbf{L}_a\mathbf{u}^n$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \delta t \theta \mathbf{L}_a \mathbf{u}^{n+1} + \delta t (1-\theta)\mathbf{L}_a \mathbf{u}^n + \delta t \mathbf{f} \tag{8}$$

where $\mathbf{u}^n$ is the numerical solution at $t = n\delta t$ using a time step δt. The method has a local truncation error

$$\mathbf{e}^{n+1} = \mathbf{e}^n + \begin{cases} \mathcal{O}(\delta t^3) & \theta = 1/2 \\ \mathcal{O}((2\theta - 1)\delta t^2) & \theta \neq 1/2 \end{cases} \tag{9}$$

where $\mathbf{e}^{n+1} := \mathbf{u}^{n+1} - \mathbf{u}((n+1)\delta t)$, which is a second-order method for $\theta = 1/2$ and a first-order method otherwise. If $\theta \geq 1/2$, the theta method is A-stable, that is, the method is stable for any choice of δt. Although the method enjoys the strong stability for $\theta \geq 1/2$, the method is implicit and it requires to solve a solution $\mathbf{u}^{n+1}$ of an algebraic equation

$$(\mathbf{I} - \theta \mathbf{L}_a)\mathbf{u}^{n+1} = \mathbf{b} \tag{10}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix and $\mathbf{b} = \mathbf{u}^n + (1-\theta)\mathbf{L}_a\mathbf{u}^n + \mathbf{f}$. Except for the constant-coefficient case, where fast algorithms are available to invert $(\mathbf{I} - \theta\mathbf{L}_a)$ efficiently, the inversion of $(\mathbf{I} - \theta\mathbf{L}_a)$ is a computational bottle neck of the theta method and it typically relies on an optimized iterative solver, such as preconditioned conjugate gradient method. In the next section, we propose a fast and stable method to integrate (1) with a variable coefficient, which utilizes the property of the constant-coefficient case and the flexibility of θ to maintain stability.

3 Fast time integrator

We introduce a fast time integration of the parabolic equation (1) with a variable constant diffusion coefficient, which achieves a low computational complexity comparable to the constant coefficient case. If the coefficient $a(x)$ is periodic for which a homogenized constant coefficient is available (after solving a cell problem or the harmonic average for 1D), it is straightforward to achieve a low computational cost using the homogenized constant diffusion coefficient (along with a coarse spatial discretization mesh size). If the diffusion coefficient does not have a nice structure such as periodicity, it is challenging to calculate the homogenized coefficient as the explicit scale separation exists to solve a cell problem. The idea of the proposed method is to incorporate the specification of the value of the constant coefficient into the theta method without calculating the unknown effective constant coefficient.

3.1 Algorithm

The proposed method can be derived by rewriting the theta method (8).

$$\begin{aligned} (\mathbf{I} - \delta t\theta\mathbf{L}_a)\mathbf{u}^{n+1} &= (\mathbf{I} - \delta t\theta\mathbf{L}_a)\mathbf{u}^n + \delta t\mathbf{L}_a\mathbf{u}^n + \delta t\mathbf{f} \\ \Rightarrow \mathbf{u}^{n+1} &= \mathbf{u}^n + (\mathbf{I} - \delta t\theta\mathbf{L}_a)^{-1}\left(\delta t\mathbf{L}_a\mathbf{u}^n + \delta t\mathbf{f}\right) \\ &= \mathbf{u}^n + (\mathbf{I} - \delta t\theta\mathbf{L}_a)^{-1}\delta t\left(\mathbf{L}_a\mathbf{u}^n + \mathbf{f}\right). \end{aligned} \tag{11}$$

The calculation of $\mathbf{L}_a\mathbf{u}^n$ is straightforward; once a space discretization is specified, $\mathbf{L}_a\mathbf{u}^n$ is a matrix-vector product that depends on only the current step value and thus cheap compared to the inversion of $(\mathbf{I} - \delta t\theta\mathbf{L}_a)$. The idea of the proposed method is to approximate $\mathbf{L}_a$ with an operator that is easy to invert, that is, the differential operator with a constant coefficient $\mathbf{L}_{\bar{a}}$ (which is the Laplacian multiplied by a constant $\bar{a}$)

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + (\mathbf{I} - \delta t\theta\mathbf{L}_{\bar{a}})^{-1}\delta t\left(\mathbf{L}_a\mathbf{u}^n + \mathbf{f}\right) \\ &= \mathbf{u}^n + (\mathbf{I} - \delta t\theta\bar{a}\mathbf{L}_1)^{-1}\delta t\left(\mathbf{L}_a\mathbf{u}^n + \mathbf{f}\right). \end{aligned} \tag{12}$$

Here $\mathbf{L}_1$ is a $N \times N$ matrix approximating the Laplacian Δ. One characteristic of (12) is that specification of θ and the constant coefficient $\bar{a}$ is not necessary. We specify only their product $\theta\bar{a}$ as the other terms are independent of θ and $\bar{a}$. In the specification of the product $\theta\bar{a}$, we consider a value that guarantees the stability of the method for a large time step δt.

As it will be shown in Section 3.2, the method is stable when $\theta \geq 1/2$ (with an additional constraint on δt but the method allows a large time step; see Theorem 1). Thus, the choice of the product $\theta\bar{a}$ must guarantee that $\theta \geq 1/2$ without specifying the unknown coefficient $\bar{a}$. For example, in the homogenization of a periodic coefficient $a(x, \frac{x}{\epsilon})$ where $a(x, y)$ is periodic in y, the homogenized tensor of $a(x, \frac{x}{\epsilon})$ has an upper bound, that is, its largest eigenvalue is bounded by $\int_{(0,1)^d} a(x, y)dy$, which is the arithmetic mean of the coefficient [10]. As the local averaging domain is not obvious for a general coefficient without scale separation, the upper bound for the coefficient will be the maximum value of the coefficient.

Based on this argument, we choose $\theta\bar{a}$ to be $\frac{a_{sup}}{2}$ where $a_{sup} = \|a(x)\|_\infty$, which yields the following one-step integration of the proposed method

$$\mathbf{u}^{n+1} = \mathbf{u}^n + (\mathbf{I} - \frac{\delta t a_{sup}}{2}\mathbf{L}_1)^{-1}\delta t(\mathbf{L}_a\mathbf{u}^n + \mathbf{f}) \tag{13}$$

In the rectangular domain $\Omega = (0, L)^d$, the inversion of $(\mathbf{I} - \frac{ka_{sup}}{2}\mathbf{L}_1)$ can be done efficiently as we know the eigenvalues and the eigenfunctions of Laplacian in $\Omega = (0, L)^d$ for periodic/Dirichlet/Neumann boundary conditions. As the eigenfunctions are tensor product of trigonometric functions, the fast Fourier transform can invert the matrix by solving for $\mathbf{r}$

$$(\mathbf{I} - \frac{ka_{sup}}{2}\mathbf{L}_1)\mathbf{r} = \mathbf{L}_a\mathbf{u}^n + \mathbf{f}. \tag{14}$$

We summarize the one-step integration method of the proposed algorithm

Algorithm: one time step integration of the proposed method using a time step δt.

---

1. Calculate $\mathbf{b} = \mathbf{L}_a\mathbf{u}^n + \mathbf{f}$
2. Use the fast Fourier transform to solve for $\mathbf{r}$
   $(\mathbf{I} - \frac{ka_{sup}}{2}\mathbf{L}_1)\mathbf{r} = \mathbf{b}$
3. Add the correction $\delta t\mathbf{r}$ to $\mathbf{u}^n$, which yields $\mathbf{u}^{n+1}$
   $\mathbf{u}^{n+1} = \mathbf{u}^n + \delta t\mathbf{r}$

---

* One-step integration complexity is $\mathcal{O}(N \log N)$ where N is the number of grid points.

3.2 Stability

The method introduces an approximation in an operator (a constant coefficient differential operator approximating the variable coefficient differential operator), which enables to use the fast matrix inversion using the known eigenvalues and eigenfunctions of Laplacian. Due to this approximation, the proposed method loses A-stability of the theta method (for $\theta \geq 1/2$). In this section, we show that the proposed method is stable for a sufficiently large time step under certain conditions. In this section, $\|\cdot\|$ represents the l_2 norm in $\mathbb{R}^N$.

Theorem 1. *If there exists symmetric negative definite* $\mathbf{L}_{\bar{a}}$ *such that* $\|\mathbf{L}_{\bar{a}} - \mathbf{L}_a\| \leq \epsilon \ll 1$ *and* $|\epsilon| < |\lambda|$ *for each eigenvalue* λ *of* $\mathbf{L}_{\bar{a}}$*, the proposed time integrator* (13) *is stable for* $\delta t \leq \mathcal{O}(\frac{1}{\epsilon})$ *and* $\theta \geq \frac{1}{2}$.

Proof. We rewrite the proposed method (12) without forcing f

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t \mathbf{L}_a \mathbf{u}^n \\ &= (\mathbf{I} + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t(\mathbf{L}_{\bar{a}} + \tilde{\mathbf{L}}))\mathbf{u}^n \end{aligned} \tag{15}$$

where $\tilde{\mathbf{L}} = \mathbf{L}_a - \mathbf{L}_{\bar{a}}$, which satisfies $\|\tilde{\mathbf{L}}\| \leq \epsilon$ from the assumption. For stability, we need to show the norm of the operator $(1 + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t(\mathbf{L}_{\bar{a}} + \tilde{\mathbf{L}}))$ is less than 1. The norm of this matrix is bounded by

$$\begin{aligned} \|(1 + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t(\mathbf{L}_{\bar{a}} + \tilde{\mathbf{L}}))\| &\leq \|(\mathbf{I} + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t \mathbf{L}_{\bar{a}})\| + \|(\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t\tilde{\mathbf{L}}\| \\ &\leq \|(\mathbf{I} + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t \mathbf{L}_{\bar{a}})\| + \|(\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\|\|\delta t\tilde{\mathbf{L}}\| \\ &= \left|1 + \frac{\delta t\lambda}{1 - \delta t\theta\lambda}\right| + \left|\frac{1}{1 - \delta t\theta\lambda}\right| \delta t\|\tilde{\mathbf{L}}\|. \end{aligned} \tag{16}$$

for an eigenvalue of $\mathbf{L}_{\bar{a}}$. We will show that this is bounded by 1 by showing

$$\left|1 + \frac{\delta t(\lambda + e)}{1 - \delta t\theta\lambda}\right| < 1 \tag{17}$$

for $e = \pm\|\tilde{\mathbf{L}}\|$, which comes from $|a| + |b| = \max(|a + b|, |a - b|)$. This inequality is equivalent to show the following inequalities

$$-2 < \frac{\delta t(\lambda + e)}{1 - \delta t\theta\lambda} < 0. \tag{18}$$

For $\delta t < \frac{2}{|e|} = \mathcal{O}(1/\epsilon)$, $2 + \delta te > 0$. For $\theta \geq 1/2$, $(2\theta - 1)\delta t\lambda$ is non-positive as λ is negative. Therefore,

$$\begin{aligned} (2\theta - 1)\delta t\lambda &< 2 + \delta te \\ -2 + 2\theta\delta t\lambda &< \delta t\lambda + \delta te \end{aligned} \tag{19}$$

By dividing by $1 - \delta t\theta\lambda$ (which is positive) on both sides, we have the first inequality of (18). For the second inequality of (18), the magnitude of e is smaller than $|\lambda|$ for each eigenvalue of $\mathbf{L}_{\bar{a}}$. Thus $\lambda + e$ is always negative for all λs, which proves the inequality. □

The proposed method chooses $\theta\bar{a}$ to be $\frac{a_{sup}}{2} = \frac{\|a(x)\|_\infty}{2}$, which guarantees that $\theta = \frac{a_{sup}}{2\bar{a}} \geq 1/2$. Note that the upper bound of δt, $\mathcal{O}(\frac{1}{\epsilon})$, is larger than $\mathcal{O}(1)$ when $\epsilon \ll 1$. Thus, the proposed method allows a large time step for $\theta \geq 1/2$. As it is often required to have accuracy less than $\mathcal{O}(1)$, the time step of the proposed method is restricted by accuracy rather than by stability. We now analyze the proposed method for accuracy.

3.3 Error analysis

For the error analysis, we compare the error between two numerical solutions, the solution of the proposed method (12) denoted as $\mathbf{u}^n$, and the solution of the theta method without the operator approximation (8) denoted as $\mathbf{v}^n$. Before we show the error analysis, we need the following lemma.

Lemma 1. *The solution $\mathbf{u}^n$ of the proposed method is bounded by $kn\|f\|$ when δt is chosen for stability. Therefore, in the integration of the model equation up to a time T of order 1, the solution $\mathbf{u}^n$ remains bounded by $\|f\|$ (which we assume to be $\mathcal{O}(1)$).*

Proof. From (12),

$$\mathbf{u}^{n+1} = (\mathbf{I} + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t(\mathbf{L}_{\bar{a}} + \tilde{\mathbf{L}})\mathbf{u}^n + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}k\mathbf{f} \tag{20}$$

As $\mathbf{L}_{\bar{a}}$ is negative definite, $\|(\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\| < 1$ for all $k > 0$ and $\theta > 0$. From the proof of the stability, $\|(\mathbf{I} + (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}\delta t(\mathbf{L}_{\bar{a}} + \tilde{\mathbf{L}})\| < 1$ when the method is stable. Using these facts, an iterative application of the triangle inequality proves the lemma. □

Theorem 2. *The difference $\mathbf{e}^n = \mathbf{u}^n - \mathbf{v}^n$ between the solutions of the proposed method* (12) *and the theta method* (8) *satisfies the following relation*

$$\mathbf{e}^{n+1} = A\mathbf{e}^n + \mathcal{O}(\delta t\theta\epsilon) \tag{21}$$

where $\|A\| < 1$ for $\theta \geq 1/2$.

Proof. Multiply (11) and (12) by $(\mathbf{I} - \delta t\theta \mathbf{L}_a)^{-1}$ and $(\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})^{-1}$ respectively, which yields

$$(\mathbf{I} - \delta t\theta \mathbf{L}_a)\mathbf{v}^{n+1} = (\mathbf{I} - \delta t\theta \mathbf{L}_a)\mathbf{v}^n + \delta t(\mathbf{L}_a\mathbf{v}^n + \mathbf{f}) \tag{22}$$

and

$$(\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})\mathbf{u}^{n+1} = (\mathbf{I} - \delta t\theta \mathbf{L}_{\bar{a}})\mathbf{u}^n + \delta t(\mathbf{L}_a\mathbf{u}^n + \mathbf{f}). \tag{23}$$

Using the same decomposition of $\mathbf{L}_{\bar{a}}$ in the previous theorem

$$\mathbf{L}_{\bar{a}} = \mathbf{L}_a - \tilde{\mathbf{L}}, \qquad \|\tilde{\mathbf{L}}\| \leq \epsilon, \tag{24}$$

after subtracting the first equation from the second one, the equation for $\mathbf{e}^{n+1}$ is

$$\begin{aligned}(\mathbf{I} - \delta t\theta \mathbf{L}_a)\mathbf{e}^{n+1} + \delta t\theta\tilde{\mathbf{L}}\mathbf{u}^{n+1} &= (\mathbf{I} - \delta t\theta \mathbf{L}_a)\mathbf{e}^n + \delta t\mathbf{L}_a\mathbf{e}^n + \delta t\theta\tilde{\mathbf{L}}\mathbf{u}^n \\ &= (\mathbf{I} + \delta t(1-\theta)\mathbf{L}_a)\mathbf{e}^n + \delta t\theta\tilde{\mathbf{L}}\mathbf{u}^n.\end{aligned} \tag{25}$$

Move the term containing $\mathbf{u}^{n+1}$ to the right hand side

$$(\mathbf{I} - \delta t\theta \mathbf{L}_a)\mathbf{e}^{n+1} = (\mathbf{I} + \delta t(1-\theta)\mathbf{L}_a)\mathbf{e}^n - \delta t\theta\tilde{L}(\mathbf{u}^{n+1} - \mathbf{u}^n), \tag{26}$$

and invert $(\mathbf{I} - \delta t\theta L)$ to have

$$\begin{aligned}\mathbf{e}^{n+1} &= (\mathbf{I} - \delta t\theta \mathbf{L}_a)^{-1}(\mathbf{I} + \delta t(1-\theta)\mathbf{L}_a)\mathbf{e}^n - (\mathbf{I} - \delta t\theta \mathbf{L}_a)^{-1}\delta t\theta\tilde{\mathbf{L}}(\mathbf{u}^{n+1} - \mathbf{u}^n) \\ &= A\mathbf{e}^n - \delta t\theta B(\mathbf{u}^{n+1} - \mathbf{u}^n)\end{aligned} \tag{27}$$

where $A = (\mathbf{I} - \delta t\theta \mathbf{L}_a)^{-1}(\mathbf{I} + \delta t(1-\theta)\mathbf{L}_a)$ and $B = (\mathbf{I} - \delta t\theta \mathbf{L}_a)^{-1}\tilde{\mathbf{L}}$. A is the matrix related to the theta method that is bounded by 1 for $\theta \geq 1/2$ (this fact can also be derived from the proof of the stability theorem by taking $\epsilon \to 0$. Regarding the matrix B, $\|B\| \leq \epsilon$ as $\|(\mathbf{I} - \delta t\theta \mathbf{L}_a)^{-1}\| < 1$ for all positive δt and θ. For the difference $\mathbf{u}^{n+1} - \mathbf{u}^n$, it is also of order $\mathcal{O}(1)$ from Lemma 1 as $\mathbf{u}^{n+1}$ and $\mathbf{u}^n$ are bounded. Thus we have the following local error

$$\mathbf{e}^{n+1} = A\mathbf{e}^n + \mathcal{O}(\delta t\theta\epsilon) \tag{28}$$

□

We want to note that the difference $\mathbf{u}^{n+1} - \mathbf{u}^n = (\mathbf{I} - \delta t\theta\bar{a}\mathbf{L}_1)^{-1}\delta t(\mathbf{L}_a\mathbf{u}^n + \mathbf{f})$ can be larger than $\mathcal{O}(k)$ as $\mathbf{L}_a$ is a stiff operator. In the proposed method, it is not possible to specifically choose $\theta = 1/2$ as we do not specify the unknown coefficient. Thus the best accuracy we can expect is the first-order accuracy. By combining the above analysis with the local error analysis of the theta method (9), we can obtain the following global error analysis of the proposed method.

Corollary 1. *In the integration of* (7) *using the proposed method* (12) *up to a time of order 1, the error of the proposed method is bounded by* $\mathcal{O}((2\theta - 1)\delta t) + \mathcal{O}(\theta\epsilon)$.

4 Numerical Tests

In this section, we test the proposed method for the model problem (1) in 1D and 2D with $\Omega = (0, 2\pi)^d, d = 1, 2$, and variable coefficients that include random variations. Two boundary conditions are considered to complete the model equation (1); periodic and Dirichlet boundary conditions. In the periodic boundary case, we use the Fourier spectral method using the fast Fourier transform to discretize in the space. In the Dirichlet boundary case, we use the standard second-order centered difference that uses two adjacent grid points in 1D (or four points in 2D) for spatial discretization. The boundary value is homogeneous with 0 everywhere on the boundary. Therefore, we use the odd extension and use the fast Fourier transform in the extended domain to invert $(\mathbf{I} - \frac{a_{sup}}{2}\mathbf{L}_1)$.

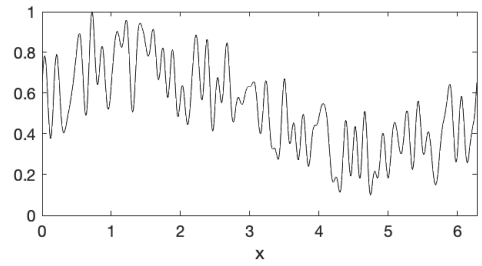
To measure the performance of the proposed method, we compare with a fourth-order explicit Runge-Kutta (RK4) method. Additionally, we also compare the proposed method with the Crank-Nicolson method with a preconditioned conjugate gradient method in some of the tests. In each test, we specify a marginally stable time step (i.e., the largest time step that prevents the method blows up) to check the computational saving of the proposed method. For a reference simulation to compare with the proposed method, we use a fine step that guarantees convergence (with an error less than 10^{-12}). We measure the qualitative and quantitative performance of the proposed method through 1) solution plots and 2) errors in l_2 and l_∞ norms in comparison with the reference result.

4.1 1D random coefficient with a Dirichlet boundary condition

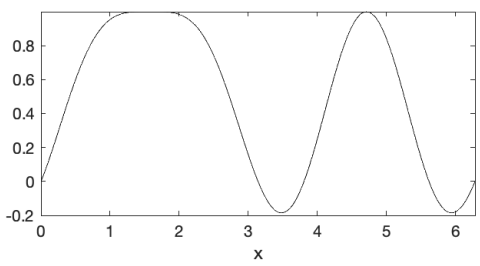
Our first test is the model problem (1) in 1D with $f(x) = 0$ and a Dirichlet boundary condition $u(t, x) = 0$ at $x = 0, 2\pi$. The coefficient contains both random and deterministic variations, which is shown in Figure 1 (a). To generate the coefficient, we draw 100 random numbers from the uniform distribution in $[0, 1]$. These numbers are assigned to 100 uniformly spaced grid points and then are interpolated on the 1000 uniform grid points in $(0, 2\pi)$. In addition to this random component, we add a deterministic component $\sin(x)$ and rescale the coefficient so that the maximum and the minimum of the coefficient are 1 and 0.1 respectively. The spatial discretization uses $N = 1000$ grid points so that there are 10 grid points to resolve the finest variation of the coefficient. The initial value $u_0(x)$ is a smooth periodic function having zero values on the boundary (see Figure 1 (b) for a plot of the initial value)

$$u_0(x) = -\cos^2(x)(1 - \sin(x)) + 1. \tag{29}$$

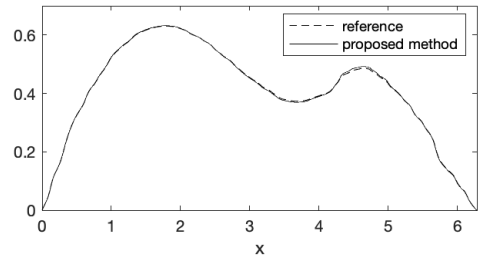
We solve the model equation up to $t = 1$ and compare with a reference solution described below.



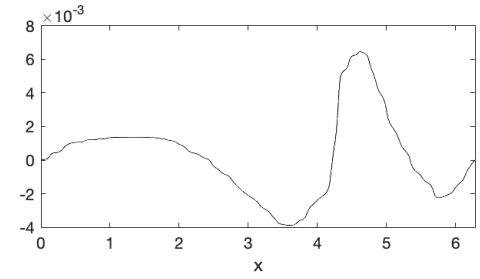
(a) Diffusion coefficient $a(x)$



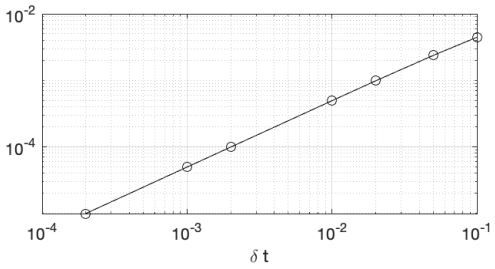
(b) Initial value



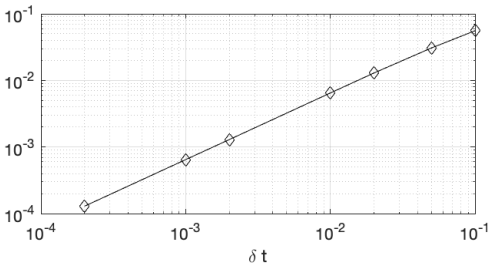
(c) Solutions at $t = 1$



(d) Pointwise error



(e) l_2 error convergence rate



(f) l_∞ error convergence rate

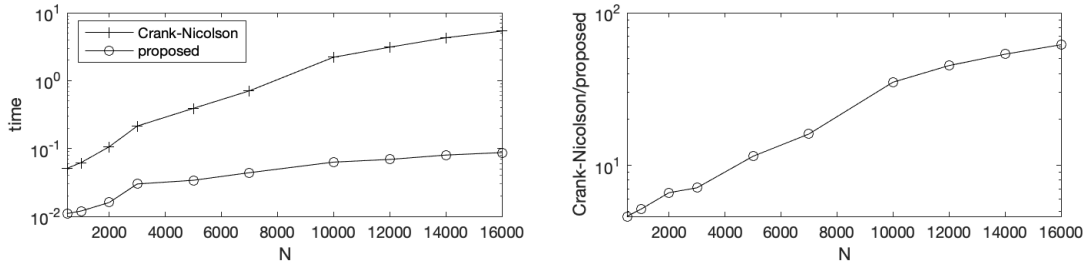
Figure 1: Diffusion with a 1D random coefficient with a Dirichlet boundary condition in $(0, 2\pi)$. Spatial discretization uses the the second-order centered finite difference. The proposed method in (c) and (d) uses a time step $\delta t = 10^{-2}$.

As the maximum of the coefficient is 1, the proposed method needs to invert $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_1)$ where δt is the time step. To utilize the fast Fourier transform to invert this operator, vectors are extended to $(-2\pi, 2\pi)$ using the odd extension and uses the fast Fourier transform with the periodic boundary condition in the extended domain. Note that it is not necessary to store the matrix $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_1)$ on a computer memory as we use the known eigenvalues and eigenvectors of the constant coefficient Laplacian $\mathbf{L}_1$.

By varying the time step δt, we have checked that the proposed method does not diverge for time steps even larger than 10 although it loses accuracy. From the consideration of accuracy, the solution of the proposed method using a time step $k = 10^{-1}$ is shown (solid line) in Figure 1 (c),

which is on top of the the RK4 reference solution (dash line; this reference result uses a time step 10^{-6} for convergence). The marginally stable time step of the RK4 method is 2.58×10^{-5}, that is, the proposed method can use a time step at least 380 times longer than the RK4 method.

Figure 1 (d) shows the pointwise error of the proposed method (that is, the difference between the proposed method solution and the reference solution) along with l_2 and l_∞ errors for varying time steps in (e) and (f). The l_2 and l_∞ errors shown in the log-scale follows the first-order accuracy of the global error estimate, which implies that the magnitude of the operator error, ϵ, is comparable to or less than the time step (or the spatial resolution). If the effective coefficient $\bar{a}$ varies over locations, we expect that the operator error will be comparable to the spatial resolution. As an evidence supporting this claim, we check in Figure 1 (d) that the max error is at around $x = 4.75$ where the coefficient obtains the minimum value 0.1. From the error analysis (Theorem 2), the error will be large when θ is large (assuming that other factors do not change). If the local effective coefficient $\bar{a}$ is small, then its corresponding θ will be large as we use a global constant for the product $\theta\bar{a}$ without specifying them individually (as we cannot specify $\bar{a}$). It is natural to speculate that the efficient coefficient around $x = 4.75$ is the smallest among other locations as the coefficient has the minimum. Thus, the corresponding effective θ value at this location will be the largest, which explains the largest error.



(a) (b)

Figure 2: Savings of the proposed method against the Crank-Nicolson method using an iterative solver.

In addition to the comparison with the explicit RK4, we compare the proposed method with the Crank-Nicolson method that correspond to $\theta = 1/2$ of the theta method (8). To invert $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_a)$, we use the preconditioned conjugate gradient method using the incomplete LU factorization with a relative error tolerance 10^{-2}. Figure 2 (a) shows the computational times to solve the model equation (1) for up to $t = 1$ using the same time step $\delta t = 10^{-2}$ and various spatial resolutions ranging from 100 to 16000 grid points along with their ratio ($\frac{\text{time of Crank-Nicolson}}{\text{time of the proposed method}}$) in (b). As the total number of grid points N increases, the Crank-Nicolson with an iterative solver shows a larger increase in the computation time than the $\mathcal{O}(N \log N)$ complexity of the proposed method. When $N = 16,000$, the proposed method is about 60 times faster than the Crank-Nicolson method. We want to note that in this special 1D case with the centered finite differencing, the tridiagonal solver with a $\mathcal{O}(N)$ complexity can be used to invert the matrix $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_a)$. In Section 4.4, we will compare the proposed method with an iterative solver in a 2D problem where no linear complexity solver is available to invert the matrix $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_a)$.

4.2 1D random coefficient with the periodic boundary condition

The second test solves the model problem in 1D with the periodic boundary condition with a periodicity 2π. Instead of the second-order centered finite difference method in the previous test, we use the Fourier spectral discretization and does time integration in the Fourier space. The initial value and the coefficient are the same as in the previous test. The Fourier spectral method allows to integrate the model equation in the Fourier domain

$$\hat{u}^{n+1} = \hat{u}^n + \frac{\delta t}{1 + \frac{\delta t}{2} k^2} \widehat{\nabla \cdot (a(x) \nabla u^n)} \tag{30}$$

where k is the wavenumber. This method is an explicit integration using a variable time step $\frac{\delta t}{1+\frac{\delta t}{2}k^2}$ where the variable time step corresponds to $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_1)^{-1}$ in the Fourier space. Although we do not use the fast Fourier transform to invert a matrix, we use the fast Fourier transform for the differential operator $\nabla \cdot (a(x)\nabla u^n)$ and thus the complexity remains at $\mathcal{O}(N \log N)$.

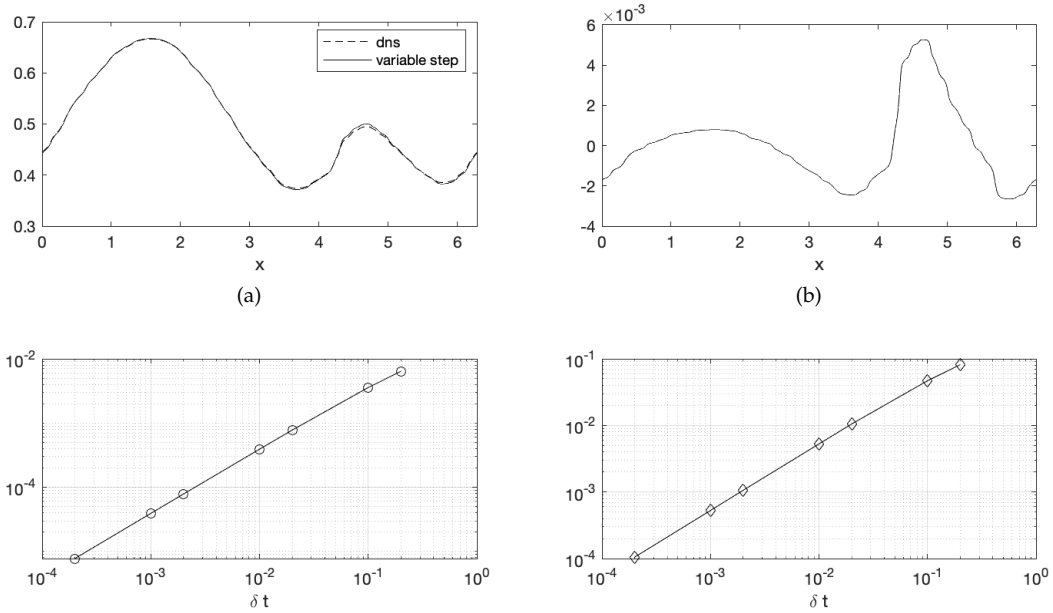


Figure 3: Diffusion with a 1D random coefficient with the periodic boundary condition in $(0, 2\pi)$. Spatial discretization uses the Fourier spectral method. The proposed method in (a) and (b) uses a time step $\delta t = 10^{-2}$.

The proposed method uses $\delta t = 10^{-2}$ that is about 800 times larger than the marginally stable time 1.15×10^{-5} of the fourth-order Runge-Kutta method. The solutions of both methods are shown in Figure 3 along with the convergence tests in l_2 and l_∞ norms. As in the Dirichlet boundary case, the proposed method shows the first-order accuracy and matches with the reference result with a maximum error less than 6×10^{-3}. In the error plot (Figure 3 (b)), we observe that

the maximum error coincides with the location where the coefficient is the minimum, $x = 4.75$. This result supports that the unspecified effective coefficient $\bar{a}$ is close to the local values of the coefficient even with the global differentiation using the Fourier spectral method.

As another experiment, we use a variable time step $\frac{\delta t}{1+\frac{\alpha \delta t}{2}k^2}$ for several α values less than the maximum of $a(x)$. This corresponds to inverting $(\mathbf{I} - \frac{\delta t \alpha}{2}\mathbf{L})$ where the product $\theta\bar{\alpha}$ is set to $\frac{\alpha}{2}$. The proposed method runs up to $t = 100$ using $\delta t = 10^{-2}$ and we check whether the solution diverges or not. If we interpret the effective constant in the context of homogenization with a periodicty 2π, the homogenized coefficient (that is, the harmonic mean) is much less than the maximum of $a(x)$ (the homogenized coefficient using the large periodicity is 0.4259). The proposed method shows no divergence when $\alpha \geq 0.996$ within 10,000 iterations but any α value less than 0.996 becomes unstable and diverges. Figure 4 shows the solution using $\alpha = 0.995$ before it diverges to the machine infinity. The solution shows a particularly large value located around $x = 0.73$ that coincides with the location where the coefficient has the maximum value 1. The value 0.996 is close to the homogenized coefficient of a at three grid points centered at $x = 0.73$; the homogenized coefficient is 0.9959. We want to note that this behavior cannot be observed clearly in the previous case. The centered finite difference is more diffusive than the Fourier spectral method and thus the proposed method is stable even for a α less than 0.996.

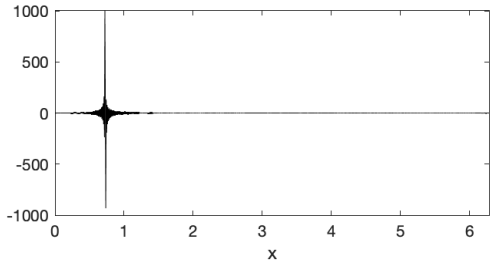


Figure 4: An unstable solution of the proposed method by inverting $(\mathbf{I} - \frac{0.995\delta t}{2}\mathbf{L}_1)$ instead of $(\mathbf{I} - \frac{\delta t}{2}\mathbf{L}_1)$ where the latter is stable. Spatial discretization uses the Fourier spectral method.

4.3 2D random coefficient with the periodic boundary condition

The next test is a 2D problem in $\Omega = (0, 2\pi)^2$ with the periodic boundary condition. A 2D diffusion coefficient is generated similarly as in the 1D case; we draw 100×100 random numbers from the uniform distribution in $[0, 1]$ and interpolate these values on 1000×1000 uniform grid points in the domain. As a deterministic component, we add $\sin(x_1)\sin(x_2)$ to the random component and rescale the coefficient so that its maximum and minimum values are 1 and 0.1 respectively (see Figure 3 (a) for the coefficient). As there is no external source ($f = 0$), the solution decays from an initial value $u_0(x_1, x_2)$ that is given as

$$u_0(x_1, x_2) = (\cos^2(x_1)(1 + \sin(x_1)) - 1) \times (\cos^2(2x_2)(1 + \sin(x_2)) - 1) \tag{31}$$

that is a tensor product of the modified 1D initial value (29) (the 2D initial value is shown in Figure 3 (b)). Using the Fourier spectral method for spatial discretization with 1000 grid points in each

direction, we integrate the model equation in the Fourier domain using a variable time step

$$\frac{\delta t}{1 + \frac{\delta t}{2}(k_{x_1}^2 + k_{x_2}^2)} \tag{32}$$

where k_{x_1} and k_{x_2} are the wavenumbers in the x_1 and x_2 directions respectively, and δt is the base time step corresponding to the constant term of the solution (i.e., $k_x = k_y = 0$). Figure 5 (c) shows the solution of the proposed method at $t = 0.5$ using $\delta t = 10^{-2}$ along with the reference solution in (d) that uses a time step 10^{-6} for the fourth-order Runge-Kutta explicit integration method (the marginally stable time step of the Runge-Kutta method is 6.6×10^{-6}, which is 1500 times shorter than the time step of the proposed method). The solution using the proposed method stays on top of the reference solution with a maximum error less than 8.2×10^{-3}. Also the l_2 and l_∞ norms follow the first-order convergence rate predicted by the analysis (see Figure 5 (g) and (h)). This implies that the operator error is less than or comparable to δt.

In this 2D test, we can check the similar relationship between the location of the max error and the location of the minimum coefficient value as in the 1D tests. From the 2D plots of the error magnitude and the coefficient (Figure 5 (e) and (f) respectively), we observe that the local maxima of the error coincides with the location in which the coefficient is the minimum. This result again supports that the effective constant $\bar{a}$ is close to the local value of the coefficient and thus yields an effect of a large θ in the time integration. To check local stability when the coefficient obtains the maximum value, we use the following variable time step

$$\frac{\delta t}{1 + \frac{\alpha \delta t}{2}(k_{x_1}^2 + k_{x_2}^2)} \tag{33}$$

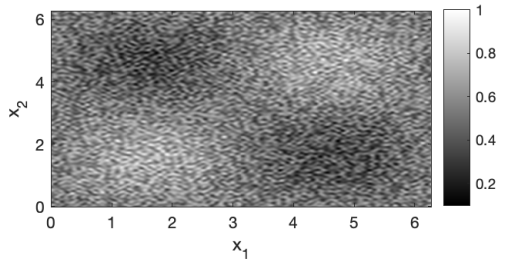
for $\alpha = 0.995$ which is slightly smaller than the local homogenization coefficient 0.9955 (the local homogenization coefficient is calculated using the four adjacent grid points in addition to the maximum point). Using this α value and the same base time step $\delta t = 10^{-2}$, the variable time step integration diverges after 3500 iterations. Figure 5 (f) shows the unstable solution of the proposed method before divergence; the solution has a significantly large value at the location $(x_1, x_2) = (4.36, 4.31)$ that corresponds to the location where $a(x)$ has the maximum value. We want to note that the location of the maximum of the coefficient is different from the peaks of the deterministic component $\sin(x_1)\sin(x_2)$. Due to the random component, the coefficient has the maximum value 1 only at $(4.36, 4.31)$; the the maximum of the other peak is 0.98. If $\alpha = 0.996$, the proposed method is stable (tested for time steps up to 10).

4.4 An elliptic problem with a random coefficient and a Dirichlet boundary condition

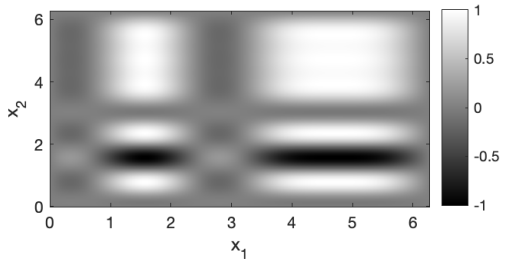
The proposed method is stable for a large step and we use this property to solve an elliptic problem, that is, the steady state of the diffusion equation. The long time limit of the model equation with the homogeneous Dirichlet boundary condition is the following elliptic problem

$$\begin{aligned} -\nabla \cdot (a(x)\nabla u) &= f \quad \text{in} \quad \Omega = (0, 2\pi)^2, \\ u(x) &= 0 \quad \text{on} \quad \partial\Omega. \end{aligned} \tag{34}$$

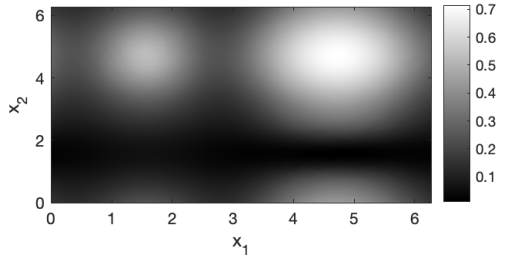
To have a non-trivial solution, we set $f(x) = f(x_1, x_2) = 5\sin(3x_1)\sin(x_2)$. The spatial discretization is the second-order centered finite difference scheme with 1000 grid points in each direction



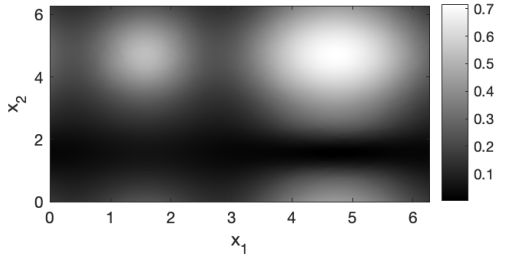
(a) Coefficient



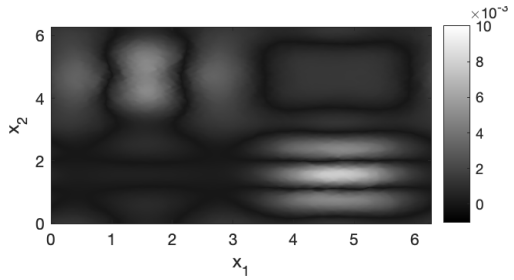
(b) Initial value



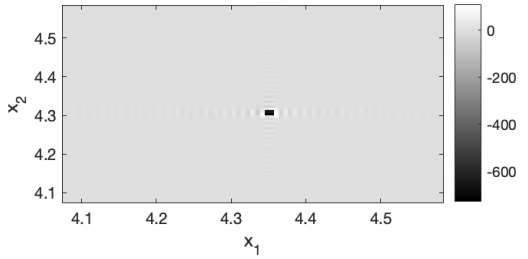
(c) Solution of the reference method at $t = 1$



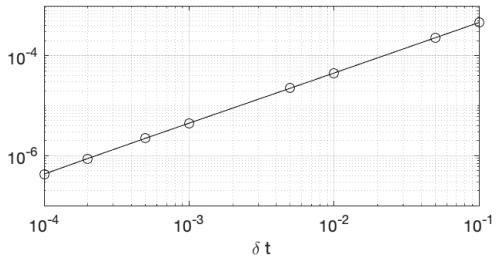
(d) Solution of the proposed method at $t = 1$



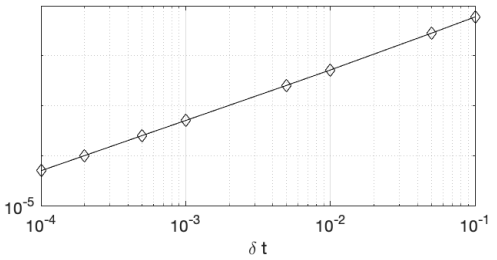
(e) Magnitude of error



(f) Unstable solution with $(\mathbf{I} - \frac{0.995\delta t}{2}\mathbf{L}_1)^{-1}$



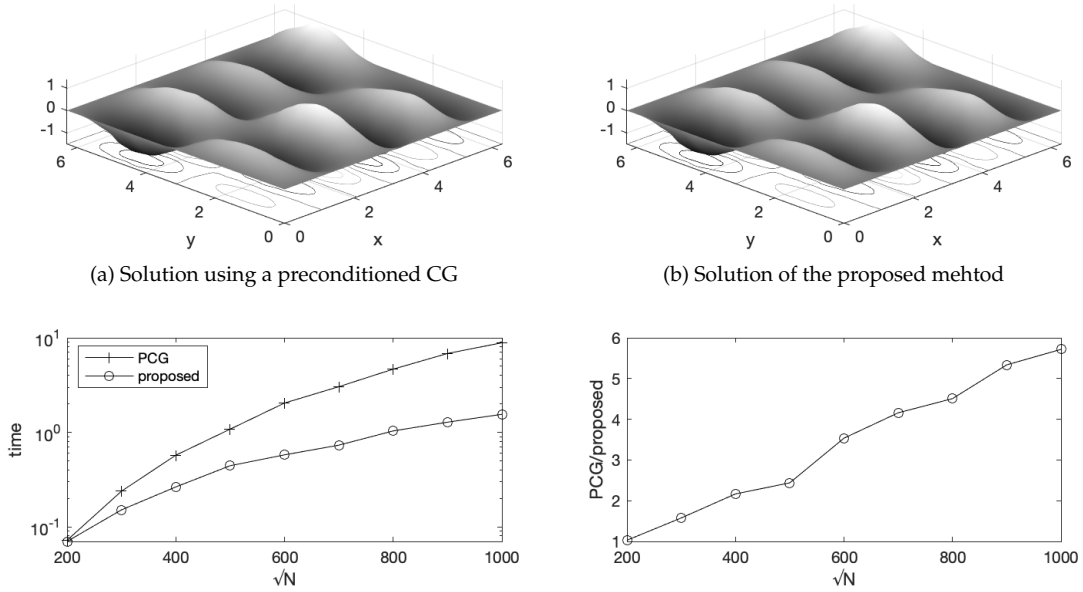
(g) l_2 errors for various δt



(h) l_∞ errors for various δt

Figure 5: 2D random coefficient with the periodic boundary condition. Spatial discretization uses the Fourier spectral method.

and the diffusion coefficient is the same coefficient used in the previous test. The proposed method uses a large time step $\delta t = 1$ that is stable for the method. To compare the result, we use the preconditioned conjugate gradient (PCG) method where the preconditioner is the incomplete LU factorization. Both the reference method and the proposed method start from the zero initial guess and stop iterations when the relative residual is less than 10^{-2}. The solutions of the two methods are shown in Figure 6 (a) and (b) that show a good match to each other (the maximum error between these two solutions is less than 2×10^{-2}).



(a) Solution using a preconditioned CG

(b) Solution of the proposed mehtod

(c) Computation times

(d) Ratio of computation times

Figure 6: An elliptic problem with the homogeneous Dirichlet boundary condition. The PCG method has a steeper increase in the computation time as the total number of grid points increases. The proposed method is 6 times faster than PCG with $N = 10^6$.

Figure 6 (c) show the computation times of the proposed method and the PCG method for various $\sqrt{N}$ values and their ratios in (d) (note the $\log$ scale in the vertical axis). For each $\sqrt{N}$, the coefficient is generated by using $(\frac{\sqrt{N}}{10})^2$ random numbers and then are interpolated on N grid points. The PCG method has a steeper increase rate than the proposed method, which implies a low complexity of the proposed method in comparison with the PCG method. With $N = 10^6$ grid points, the proposed method is 6 times faster than the PCG method. This shows a potential use of the method as a fast solver for an elliptic problem. One issue of the proposed method as an elliptic problem solver is its accuracy. By decreasing the error tolerance for the iteration, the PCG method can achieve a high accuracy but the error of the proposed method remains bounded from below (the lower bound is about 10^{-2}). If a high accuracy is a primary interest of solving an elliptic problem, the proposed method can be used as a method to generate a good initial guess for an iterative solver.

4.5 1D quasilinear diffusion with a Dirichlet boundary condition

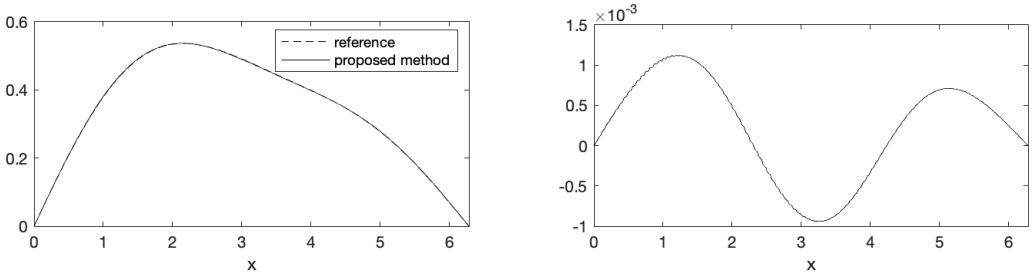
Our last test is an 1D quasilinear diffusion problem in which the diffusion coefficient depends on the solution. Particularly, as a heat transfer model in the semiconductor [14], we solve the following model using the proposed method

$$\begin{aligned} u_t &= \nabla \cdot (e^{\alpha(x)u} \nabla u), \quad (t,x) \in (0,1) \times (0,2\pi) \\ u(0,x) &= u_0(x) \end{aligned} \tag{35}$$

with the homogeneous Dirichlet boundary condition $u(t,0) = u(t,2\pi) = 0$. The initial value $u_0(x)$ is the same as (29) used in test 1 and 2. The diffusion coefficient $e^{\alpha(x)u}$ is always positive with an imposed finest scale in $\alpha(x)$

$$\alpha(x) = \phi(100x) \tag{36}$$

where $\phi(x) = \begin{cases} 1 & \pi < x \leq 2\pi \\ -1 & 0 \leq x < \pi \end{cases}$.



(a) Solutions at $t = 1$ (b) Pointwise error

Figure 7: Solutions of the quasilinear diffusion problem. The proposed method freezes the coefficient using u^n and applies to the linear part after freezing. The reference method does not use freezing.

We use the second-order centered finite difference scheme using $N = 1000$ grid points. In the context of the method of lines, the equation is integrated in time using the proposed method and the fourth-order explicit Runge-Kutta method. In the proposed method, to handle the dependence of the diffusion coefficient on the solution, we freeze the solution to u^n for the coefficient and apply the proposed method for the linear equation after freezing the coefficient

$$u^{n+1} - u^n = \delta t \theta B(u^n, u^{n+1}) + \delta t (1-\theta) B(u^n, u^n) \tag{37}$$

where $B(v,w) = \nabla \cdot (e^{\alpha v} \nabla w)$. The explicit Runge-Kutta method, on the other hand, does not freeze the coefficient. The marginally stable time step of the fourth-order Runge-Kutta is 3.2×10^{-5} while the proposed method uses $\delta t = 10^{-2}$ that is 300 times larger than the marginally stable time step. The solutions at $t = 1$ of the proposed method and the reference simulation using the Runge-Kutta are shown in Figure 7 (a); the proposed method shows a good match with the reference solution with a maximum error around 10^{-3} (Figure 7 (b)). The proposed method is stable even for a large time step $\delta t = 1$ although it loses accuracy.

5 Discussions

This paper has introduced a fast time integration method for diffusion problems with variable coefficients that requires a high spatial resolution to resolve the variation of the coefficients. The proposed method modifies the theta method, an implicit time integration method, to allow a large time step without losing stability and also to speed up calculations in an inversion of an operator. The key idea of the proposed method is to approximate the variable coefficient differential operator using a constant coefficient differential operator so that we can use the known eigenvalues and the eigenfunctions of Laplacian to expedite the matrix inversion in the theta method. The method has been applied to a suite of numerical tests, including a quasilinear problem and an elliptic problem with significantly faster computation than other standard methods including the preconditioned conjugate gradient method. We have provided stability and error analysis of the proposed method under the assumption of the existence of a global effective coefficient to approximate the variable coefficient differential operator.

Several numerical tests in this paper showed that the unspecified effective coefficient is close to the local coefficient value rather than to be modeled as a global constant in the domain. We plan to extend the analysis of this paper to a less restrictive setup that does not require the existence of a global effective constant. In this paper, we considered diffusion problems as a stiff problem with only negative eigenvalues. It is natural to investigate an application of the proposed method to a general class of stiff problems where eigenvalues are complex with negative real parts. The advection-diffusion problem in the turbulent system is a model to test in this perspective as the multiscale characteristic in the velocity field requires a high spatial resolution while the diffusion part imposes large negative real parts in eigenvalues.

We believe that the proposed method can expedite the homogenization of a time-dependent problem, particularly with many different scales. In the numerical homogenization of a time dependent problem [1], local cell problems must quickly reach a quasi-stationary state to estimate the effective behavior of the system. For an iterative homogenization where there are many different scale components, the coarsest level cell problem needs a high resolution to resolve smaller scale variations. Thus it takes a long time to reach the quasi-stationary state using standard time integration methods. Another potential application of the proposed method is a coarse integrator in parallel time integration, Parareal [15]. Instead of using a coarse resolution solver as an initial guess in Parareal that requires high-order interpolations, the proposed method can serve as a fast method to provide an initial guess with the same spatial resolution of the full resolution solver. The proposed method will be particularly useful in Parareal when the interpolation affects the convergence of Parareal. We plan to investigate the proposed method in the above potential directions, which will be reported in another paper.

Acknowledgement

The author is supported by NSF DMS-1912999 and the Burke award at Dartmouth College.

References

[1] A. ABDULLE AND W. E, *Finite difference heterogeneous multi-scale method for homogenization problems*, Journal of Computational Physics, 191 (2003), pp. 18–39.

[2] A. ABDULLE, W. E, B. ENGQUIST, AND E. VANDEN-EIJNDEN, *The heterogeneous multiscale method*, Acta Numerica, 21 (2012), p. 1–87.

[3] A. BENSOUSSAN, J.-L. LIONS, AND G. PAPANICOLAOU, *Asymptotic analysis for periodic structures*, vol. 374, American Mathematical Soc., 2011.

[4] M. M. BUTT AND M. S. TAJ, *Numerical methods for heat equation with variable coefficients*, International Journal of Computer Mathematics, 86 (2009), pp. 1612–1623.

[5] B. L. BUZBEE, G. H. GOLUB, AND C. W. NIELSON, *On direct methods for solving poisson's equations*, SIAM Journal on Numerical Analysis, 7 (1970), pp. 627–656.

[6] W. E, W. REN, AND E. VANDEN-EIJNDEN, *A general strategy for designing seamless multiscale methods*, Journal of Computational Physics, 228 (2009), pp. 5437 – 5453.

[7] B. ENGQUIST AND Y.-H. TSAI, *Heterogeneous multiscale methods for stiff ordinary differential equations*, Mathematics of Computation, 74 (2005), pp. 1707–1742.

[8] M. HOCHBRUCK AND A. OSTERMANN, *Exponential integrators*, Acta Numerica, 19 (2010), p. 209–286.

[9] A. ISERLES, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2 ed., 2008.

[10] V. JIKOV, G. YOSIFIAN, S. KOZLOV, AND O. OLEINIK, *Homogenization of Differential Operators and Integral Functionals*, SpringerLink : Bücher, Springer Berlin Heidelberg, 2012.

[11] Y. LEE, *Towards seamless multiscale computations*, PhD thesis, The University of Texas at Austin, 2013.

[12] Y. LEE AND B. ENGQUIST, *Fast integrators for dynamical systems with several temporal scales*, arXiv:1510.05728.

[13] ——, *Variable step size multiscale methods for stiff and highly oscillatory dynamical systems*, Discrete and Continuous Dynamical Systems - Series A, 34 (2014), pp. 1079–1097.

[14] J. LIENEMANN, A. YOUSEFI, AND J. G. KORVINK, *Nonlinear heat transfer modeling*, in Dimension Reduction of Large-Scale Systems, P. Benner, D. C. Sorensen, and V. Mehrmann, eds., Berlin, Heidelberg, 2005, Springer Berlin Heidelberg, pp. 327–331.

[15] J.-L. LIONS, Y. MADAY, AND G. TURINICI, *A pararеal in time discretization of pde's*, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics, 332 (2001), pp. 661–668.

[16] C. MOLER AND C. VAN LOAN, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45 (2003), pp. 3–49.

[17] W. PICKERING, *On the solution of poisson's equation on a regular hexagonal grid using fft methods*, Journal of Computational Physics, 64 (1986), pp. 320 – 333.

[18] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, USA, 2nd ed., 2003.

[19] A. STUART AND A. HUMPHRIES, *Dynamical Systems and Numerical Analysis*, no. v. 8 in Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1998.

[20] L. N. TREFETHEN AND D. BAU, *Numerical Linear Algebra*, SIAM, 1997.

[21] E. B. WATSON AND E. F. BAXTER, *Diffusion in solid-Earth systems*, Earth and Planetary Science Letters, 253 (2007), pp. 307–327.

[22] Y. ZHANG AND L. LIU, *On Diffusion in Heterogeneous Media*, American Journal of Science, 312 (2013), pp. 1028–1047.