Implementations of Exponential Time Differencing Methods for Allen–Cahn Equations in Programming

Pinzhong Zheng

June 4, 2024

Abstract

In this note, the utilization of the Fast Fourier Transform (FFT) and its variations to implement Exponential Time Differencing (ETD) methods for Allen–Cahn equations on a regular mesh in $\mathbb{R}^d$ for $d = 1, 2, 3$ is demonstrated. Three types of boundary conditions will be discussed: Dirichlet boundary conditions, periodic conditions, and Neumann conditions. This note builds upon previous preliminary codes for ETDRK methods for the Allen-Cahn equations and is inspired by [1].

Contents

1 Introduction

Consider an open rectangular domain $\Omega \in \mathbb{R}^d$ for $d = 1, 2, 3$ and a given time interval $T > 0$. We aim to solve numerically the semilinear parabolic equations of the following form:

$$u_t = D\Delta u + f(u), \quad \mathbf{x} \in \Omega, t \in [t_0, t_0 + T],$$

where D denotes the diffusion coefficient. Equations of this type are of broad interest as they model various physical phenomena. To illustrate, consider the widely used phase-field model, the Allen–Cahn equation:

$$\begin{cases} u_t = \varepsilon^2 \Delta u + f(u), & \mathbf{x} \in \Omega,\ t \in (0, T], \\ u(\mathbf{x}, 0) = u^0(\mathbf{x}), & \mathbf{x} \in \bar{\Omega}, \end{cases} \tag{1}$$

equipped with dirichlet, periodic or Neumann boundary conditions. Here, Δ representing the Laplacian operator over d dimensions, the unknown function u denotes the phase variable, and the parameter $\varepsilon > 0$ represents the interfacial width. The nonlinear term $f(u) = -F'(u)$, where F is a double-well potential with two wells at $\pm\beta$ for some $\beta > 0$. More precisely, $f : \mathbb{R} \to \mathbb{R}$ is a continuously differentiable function satisfying:

$$\exists \text{ a constant } \beta > 0, \quad \text{such that } f(\beta) \leq 0 \leq f(-\beta). \tag{2}$$

A notable feature of the Allen–Cahn equation is the *maximum bound principle* (MBP), i.e., if the initial values are within β in absolute value, the solution remains bounded by β at all times. Furthermore, this model satisfies the so-called *energy dissipation law*, because (1) can be viewed as an L^2 gradient flow with respect to the energy functional:

$$E(u) = \int_\Omega \left(\frac{\varepsilon^2}{2} |\nabla u|^2 + F(u) \right) \, \mathrm{d}x. \tag{3}$$

The energy dissipation law is more precisely formulated as:

$$\frac{\mathrm{d}}{\mathrm{d}t} E(u) = \left(\frac{\delta E(u)}{\delta u}, \frac{\partial u}{\partial t} \right) = - \left\| \partial_t u \right\|^2 \leq 0, \quad \forall t > 0, \tag{4}$$

where $(\cdot, \cdot)$ and $\| \cdot \|$ represent the standard L^2 inner product and norm.

Assume that f satisfies the Lipschitz condition with a Lipschitz constant C_l, i.e.,

$$|f(u) - f(v)| \leq C_l |u - v| \quad \forall u, v \in \mathbb{R}. \tag{5}$$

Following this, we introduce a stabilizing constant κ, satisfying:

$$\kappa \geq C_l. \tag{6}$$

By adding and subtracting a stabilization term κu to the Allen–Cahn equation (1), we derive an equivalent form of (1)

$$u_t = \mathcal{L}u + \mathcal{N}(u), \quad \mathbf{x} \in \Omega,\ t \in (0, T], \tag{7}$$

where the linear operator $\mathcal{L}$ and nonlinear operator $\mathcal{N}$ are defined as

$$\mathcal{L} \coloneqq \varepsilon^2 \Delta - \kappa \mathcal{I}, \quad \mathcal{N} \coloneqq f + \kappa \mathcal{I}, \tag{8}$$

and $\mathcal{I}$ denotes the identity operator.

Given a positive integer N_t, let the time interval $[0, T]$ be divided into N_t subintervals with a uniform time step $\tau = T/N_t$, and define $t_n = n\tau, n = 0, 1, \cdots, N$. To solve the Allen–Cahn equation (1), we focus on the equivalent equation (7) over the interval $[t_n, t_{n+1}]$, or equivalently $w^n(x, s) = u(x, t_n + s)$ satisfying the system

$$\begin{cases} \partial_s w^n = \mathcal{L}w^n + \mathcal{N}(w^n), & \mathbf{x} \in \Omega,\ s \in (0, \tau], \\ w^n(\mathbf{x}, 0) = u(\mathbf{x}, t_n), & \mathbf{x} \in \bar{\Omega}. \end{cases} \tag{9}$$

The central concept of ETD involves applying Duhamel's principle to the system, which leads to the following formulation:

$$w^n(\mathbf{x}, \tau) = \mathrm{e}^{\tau \mathcal{L}} w^n(\mathbf{x}, 0) + \int_0^\tau \mathrm{e}^{(\tau - s)\mathcal{L}} \mathcal{N}[w^n(\mathbf{x}, s)]\, \mathrm{d}s \tag{10}$$

and then approximating the nonlinear function $\mathcal{N}[u(t_n + s)]$ in the integral. Employing interpolation to approximate $\mathcal{N}[u(t_n + s)]$ leads to the development of ETD Runge–Kutta (ETDRK) methods.

2 FFT-based Numerical Methods in One Dimensions

Fast explicit numerical methods are derived for the model in one dimension. Suppose $\Omega = \{x_b < x < x_e\}$. The spatial domain is discretized using a uniform rectangular mesh as follows: $x_i = x_b + ih_x$ for $0 \leq i \leq N_x$, where $h_x = (x_e - x_b)/N_x$. The numerical solution at each mesh point is denoted by $u_i = u_i(t) \approx u(x_i, t)$ for $0 \leq i \leq N_x$. A second-order accurate central difference discretization scheme is employed for approximating the Laplace operator.

2.1 The Problem with Periodic Boundary Conditions

Suppose that the model (1) is equipped with a periodic boundary condition as

$$u(x_b,t)=u(x_e,t),\ \frac{\partial u}{\partial x}(x_b,t)=\frac{\partial u}{\partial x}(x_e,t),\quad t\in[0,T],$$

is imposed, then the set of unknowns is given as

$$\mathbf{u}=\{u_i\}_{(N_x-1)\times 1}=\begin{pmatrix} u_0\\ u_1\\ \vdots\\ u_{N_x-1}\end{pmatrix}.$$

Let

$$\left(\Delta_h^P\right)_{n\times n}\coloneqq\begin{pmatrix} -2 & 1 & 0 & 0 & \cdots & 1\\ 1 & -2 & 1 & 0 & \cdots & 0\\ & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -2 & 1\\ 1 & \cdots & 0 & 0 & 1 & -2\end{pmatrix}_{n\times n}$$

for $n\in\mathbb{N}^+$.

Then we can write the semi-discretization of (9) in space in the following compact representation:

$$\frac{d\mathbf{u}}{dt}=\mathbf{L}\mathbf{u}-\kappa\mathbf{u}+\mathcal{N}(\mathbf{u}),$$

where $\mathbf{L}^P\coloneqq\dfrac{\varepsilon^2}{h_x^2}\left(\Delta_h^P\right)_{N_x\times N_x}$. The corresponding Duhamel's principle is

$$\mathbf{u}^{n+1}=\mathrm{e}^{\tau\mathbf{L}_\kappa}\mathbf{u}^n+\int_0^\tau \mathrm{e}^{(\tau-s)\mathbf{L}_\kappa}\mathcal{N}[\mathbf{u}\left(t_n+s\right)]\,\mathrm{d}s, \tag{11}$$

where $\mathbf{L}_\kappa\coloneqq\mathbf{L}^P-\kappa\mathbf{I}$. Applying the theory of circular matrices, the following lemma presents the eigen-pairs of the circular matrix Δ_h^D of dimensions $n\times n$.

Lemma 2.1. *The eigen-pairs of the circular matrix* $\left(\Delta_h^P\right)_{n\times n}$ *is*

$$\begin{aligned}\lambda_k&=-2+w^k+w^{k(n-1)}=2(\cos\frac{2k\pi}{n}-1),\\ \vec{\phi}_k&=(1,w^k,w^{2k},...,w^{(n-1)k})^T,\end{aligned} \tag{12}$$

where $w\coloneqq e^{-i\frac{2\pi}{n}}$ *is the principal n-th root of unity, and* $k=0,1,\ldots,n-1$.

Since $\mathbf{\Phi}\mathbf{\Phi}^H = \mathbf{\Phi}^H\mathbf{\Phi} = nI$, the eigen-decomposition of Δ_h^P is

$$\left(\Delta_h^P\right)_{n\times n} = \frac{1}{\sqrt{n}}\mathbf{\Phi}\left(\Lambda^P\right)_{n\times n}\frac{1}{\sqrt{n}}\mathbf{\Phi}^H,$$

where $\mathbf{\Phi} = \left[\vec{\phi}_0, \vec{\phi}_1, \ldots, \vec{\phi}_{n-1}\right]$. Please note that $\mathbf{\Phi}$ is the matrix of the discrete Fourier transform (DFT) and $\frac{1}{n}\mathbf{\Phi}^H$ is the matrix of the discrete Fourier transform (iDFT).

Lemma 2.2. *For two matrice A, P, if P is invertible, then $e^{PAP^{-1}} = Pe^AP^{-1}$.*

This lemma can be proved directly by using the definition of matrix exponential, i.e. the series form. To compute (11), the most challenging aspect involves computing the products of $\exp(\mathbf{L})$ and $\mathbf{L}^k$ with vectors. When the spatial domain of the problem is regular and the matrix $\mathbf{L}$ possesses specific structural properties, such as being a circular matrix, algorithms based on the Fast Fourier Transform (FFT) are particularly effective for calculating these products.

Define $\gamma(z) = ce^z + \sum c_k z^k$ for $c, c_k \in \mathbb{R}$. For $\mathbf{L}^P \in \mathbb{R}^{n\times n}$ and any vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$, the FFT can be used to compute $\gamma(a\mathbf{L}^P + b\mathbf{I})\mathbf{v}$. More precisely,

$$\begin{aligned}
\phi(a\mathbf{L}^P + b\mathbf{I})\mathbf{v} &= \gamma(\frac{1}{\sqrt{n}}\mathbf{\Phi}\Lambda_1\frac{1}{\sqrt{n}}\mathbf{\Phi}^H)\mathbf{v} \\
&= \frac{1}{\sqrt{n}}\mathbf{\Phi}\gamma\left(\Lambda_1\right)\frac{1}{\sqrt{n}}\mathbf{\Phi}^H\mathbf{v} \\
&= \mathbf{\Phi}\gamma\left(\Lambda_1\right)\left(\frac{1}{n}\mathbf{\Phi}^H\right)\mathbf{v} \\
&= \mathrm{DFT}\left[\gamma\left(\mathrm{diag}\left(\Lambda_1\right)\right)\odot \mathrm{iDFT}\left(\mathbf{v}\right)\right],
\end{aligned} \tag{13}$$

where $\mathrm{diag}(A)$ denotes the diagonal vector of A, and $\odot$ denotes element by element multiplication between two arrays of same sizes. The computational complexity of (13) is $2O(n\log_2 n)+O(n) = O(n\log_2 n)$, while the complexity of direct computation of $\mathbf{L}^P\mathbf{v}$ is $O(n^2)$.

2.2 The Problem with Dirichlet Boundary Conditions

Suppose that the model (1) is equipped with a Dirichlet boundary condition

$$u(\mathbf{x}, t) = \alpha(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega,\ t \in [0, T].$$

In this case, the set of unknowns is given as

$$\mathbf{u} = \{u_i\}_{(N_x-1)\times 1} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_x-1} \end{pmatrix}.$$

Let

$$\mathbf{b}^D \coloneqq \frac{\varepsilon^2}{h_x^2} \begin{pmatrix} \alpha(x_0,t) \\ 0 \\ \vdots \\ 0 \\ \alpha(x_{N_x},t) \end{pmatrix}_{(N_x-1)\times 1},$$

and

$$\left(\Delta_h^D\right)_{n\times n} \coloneqq \begin{pmatrix} -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 1 & -2 \end{pmatrix}_{n\times n}$$

for $n \in \mathbb{N}^+$.

Then we can write the semi-discretization of (9) in space in the following compact representation:

$$\frac{d\mathbf{u}}{dt} = \mathbf{L}\mathbf{u} - \kappa\mathbf{u} + \mathbf{b}^D + \mathcal{N}(\mathbf{u}),$$

where $\mathbf{L}^D \coloneqq \dfrac{\varepsilon^2}{h_x^2}\left(\Delta_h^D\right)_{(N_x-1)\times(N_x-1)}$. The corresponding Duhamel's principle is

$$\mathbf{u}^{n+1} = \mathrm{e}^{\tau\mathbf{L}_\kappa}\mathbf{u}^n + \int_0^\tau \mathrm{e}^{(\tau-s)\mathbf{L}_\kappa}\left\{\mathbf{b}^D(t_n+s) + \mathcal{N}[\mathbf{u}(t_n+s)]\right\}\,\mathrm{d}s, \tag{14}$$

where $\mathbf{L}_\kappa \coloneqq \mathbf{L}^D - \kappa\mathbf{I}$.

In this case, $\mathbf{L}^D$ is not a circular matrix, but a topeliz matirx. The DFT will become discrete sine transform (DST). We have similar lemma:

Lemma 2.3. *The eigen-pairs of the circular matrix* $\left(\Delta_h^D\right)_{n\times n}$ *is*

$$\begin{aligned} \lambda_k &= 2(\cos\frac{k\pi}{n+1} - 1), \\ \vec{\phi}_k &= \left(\sin\frac{k\pi}{n+1}, \sin\frac{2k\pi}{n+1}, ..., \sin\frac{nk\pi}{n+1}\right)^T, \end{aligned} \tag{15}$$

where $k = 1, 2, \ldots, n$.

Since $\mathbf{\Phi}\mathbf{\Phi}^H = \mathbf{\Phi}^H\mathbf{\Phi} = \frac{n+1}{2}I$, the eigen-decomposition of Δ_h^D is

$$\left(\Delta_h^D\right)_{n\times n} = \sqrt{\frac{2}{n+1}}\mathbf{\Phi}\left(\Lambda^D\right)_{n\times n}\sqrt{\frac{2}{n+1}}\mathbf{\Phi}^H,$$

where $\mathbf{\Phi} = \left[\vec{\phi}_1, \vec{\phi}_2, \ldots, \vec{\phi}_n\right]$. Please note that $\mathbf{\Phi}$ is the matrix of the discrete Sine transform (DST) and $\frac{2}{n+1}\mathbf{\Phi}^H$ is the matrix of the discrete Sine transform (iDST).

Define $\gamma(z) = ce^z + \sum c_k z^k$ for $c, c_k \in \mathbb{R}$. For $\mathbf{L}^D \in \mathbb{R}^{n\times n}$ and any vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$, the fast DST can be used to compute $\gamma(a\mathbf{L}^D + b\mathbf{I})\mathbf{v}$. More precisely,

$$\begin{aligned}
\phi(a\mathbf{L}^D + b\mathbf{I})\mathbf{v} &= \gamma(\sqrt{\frac{2}{n+1}}\mathbf{\Phi}\Lambda_1\sqrt{\frac{2}{n+1}}\mathbf{\Phi}^H)\mathbf{v} \\
&= \sqrt{\frac{2}{n+1}}\mathbf{\Phi}\gamma\left(\Lambda_1\right)\sqrt{\frac{2}{n+1}}\mathbf{\Phi}^H\mathbf{v} \\
&= \mathbf{\Phi}\gamma\left(\Lambda_1\right)\left(\frac{2}{n+1}\mathbf{\Phi}^H\right)\mathbf{v} \\
&= \mathrm{DST}\left(\gamma\left(\mathrm{diag}\left(\Lambda_1\right)\right)\odot \mathrm{iDST}\left(\mathbf{v}\right)\right),
\end{aligned} \tag{16}$$

where $\mathrm{diag}(A)$ denotes the diagonal vector of A, and $\odot$ denotes element by element multiplication between two arrays of same sizes. The computational complexity of (13) is $2O(n\log_2 n) + O(n) = O(n\log_2 n)$, whereas the complexity of directly computing $\mathbf{L}^D\mathbf{v}$ is $O(n^2)$. There are multiple methods to achieve fast DST via FFT. Please read [2] and the Signal Processing Stack Exchange for more information. One convenient approach involves using $2n$ FFT padding as described in [2]:

$$\left[\left(\begin{array}{cccc|cccc}
-2 & 1 & & & & & & 1 \\
1 & -2 & 1 & & & & & \\
 & & \ddots & & & & & \\
 & & 1 & -2 & 1 & & & \\
\hline
 & & & 1 & -2 & 1 & & \\
 & & & & & \ddots & & \\
 & & & & & 1 & -2 & 1 \\
1 & & & & & & 1 & -2
\end{array}\right)\left(\begin{array}{c}
u_1 \\ u_2 \\ \vdots \\ u_{N_x-1} \\ \hline 0 \\ \vdots \\ 0 \\ 0
\end{array}\right)\right](1 : N_x - 1).$$

2.3 The Problem with Neumann Boundary Conditions

Suppose that the model (1) is equipped with a Neumann boundary condition

$$\frac{\partial u}{\partial \mathbf{x}} = \beta(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega,\ t \in [0, T].$$

In this case, the set of unknowns is given as

$$\mathbf{u} = \{u_i\}_{(N_x-1)\times 1} = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N_x} \end{pmatrix}.$$

Let

$$\mathbf{b}^N \coloneqq \frac{2\varepsilon}{h_x} \begin{pmatrix} \beta(x_0, t) \\ 0 \\ \vdots \\ 0 \\ \beta(x_{N_x}, t) \end{pmatrix}_{(N_x+1)\times 1},$$

and

$$\left(\Delta_h^N\right)_{n\times n} \coloneqq \begin{pmatrix} -2 & 2 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 2 & -2 \end{pmatrix}_{n\times n}$$

for $n \in \mathbb{N}^+$.

Then we can write the semi-discretization of (9) in space in the following compact representation:

$$\frac{d\mathbf{u}}{dt} = \mathbf{L}^N\mathbf{u} - \kappa\mathbf{u} + \mathbf{b}^N + \mathcal{N}(\mathbf{u}),$$

where $\mathbf{L}^N \coloneqq \dfrac{\varepsilon^2}{h_x^2}\left(\Delta_h^N\right)_{(N_x+1)\times(N_x+1)}$. The corresponding Duhamel's principle is

$$\mathbf{u}^{n+1} = \mathrm{e}^{\tau\mathbf{L}_\kappa}\mathbf{u}^n + \int_0^\tau \mathrm{e}^{(\tau-s)\mathbf{L}_\kappa}\left\{\mathbf{b}^N(t_n+s) + \mathcal{N}[\mathbf{u}(t_n+s)]\right\}\,\mathrm{d}s, \tag{17}$$

where $\mathbf{L}_\kappa \coloneqq \mathbf{L}^N - \kappa\mathbf{I}$. In this case, $\mathbf{L}^N$ is neither a circular matrix nor a Toeplitz matrix. To employ a FFT-based fast algorithm, we must make

specific modifications. An effective and equivalent approach is to apply symmetric extension to the unknowns U in each direction, achieving a periodic scenario [3]. This allows us to still solve the problem within the original domain. More precisely,

$$\left[\left(\begin{array}{cccc|ccccc} -2 & 1 & & & & & & & 1 \\ 1 & -2 & 1 & & & & & & \\ & & \ddots & & & & & & \\ & 1 & -2 & 1 & & & & & \\ & & 1 & -2 & 1 & & & & \\ \hline & & & 1 & -2 & 1 & & & \\ & & & & 1 & -2 & 1 & & \\ & & & & & \ddots & & & \\ & & & & & 1 & -2 & 1 & \\ 1 & & & & & & 1 & -2 & \end{array}\right)\left(\begin{array}{c} u_0 \\ u_1 \\ \vdots \\ u_{N_x} \\ u_{N_x+1} \\ \hline u_{N_x} \\ u_{N_x-1} \\ \vdots \\ u_1 \end{array}\right)\right](1:N_x+1).$$

3 FFT-based Numerical Methods in Two Dimensions

In Section 2, we know that three types boundary conditions can be treated by FFT. In fact, the 2D and 3D cases are similar.

Suppose $\Omega = \{x_b < x < x_e, y_b < y < y_e\}$. Let us discretize the spatial domain by a rectangular mesh which is uniform in each direction as follows: $(x_i, y_j) = (x_b + ih_x, y_b + jh_y)$ for $0 \le i \le N_x$ and $0 \le j \le N_y$ with $h_x = (x_e - x_b)/N_x$ and $h_y = (y_e - y_b)/N_y$. Let $u_i, j = u_{i,j}(t) \approx u(t, x_i, y_j)$ for $0 \le i \le N_x$ and $0 \le j \le N_y$ denote the numerical solution. We will use the second-order accurate central difference discretization scheme for the spatial derivative terms. Let us further discretize the time period as $t_n = t_0 + n\Delta t, n = 1, \ldots, N_t$ with $\Delta t = T/N_t$.

The set of unknown is given as

$$\mathbf{U} = \{u_{i,j}\}_{m\times n} = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m,1} & u_{m,2} & \cdots & u_{m,n} \end{pmatrix}. \tag{18}$$

It is easy to check that

$$(I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x)\,\mathrm{vec}\,(\mathbf{U}) \approx \Delta(\mathrm{vec}\mathbf{U}). \tag{19}$$

In fact, $I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x$ represents the 2D DFT. Using $(AB) \otimes (CD) = (A \otimes C)(B \otimes D)$, we have

$$\begin{aligned} I \otimes A &= \left(QIQ^H\right) \otimes (Q\Lambda Q^H) \\ &= (Q \otimes Q)(I \otimes \Lambda)(Q^H \otimes Q^H). \end{aligned} \tag{20}$$

Note that $(Q \otimes Q)(Q^H \otimes Q^H) = (Q^H \otimes Q^H)(Q \otimes Q) = I$, so by lemma 2.2, we have the 2D eigen decomposition.

$$I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x = (Q^y \otimes Q^x)(I^y \otimes \Lambda^x + \Lambda^y \otimes I^x)(Q^y \otimes Q^x)^H$$

$$I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x = ((Q^y)^H \otimes Q^x)(I^y \otimes \Lambda^x + \Lambda^y \otimes I^x)(Q^y \otimes (Q^x)^H)$$

$$I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x = (Q^y \otimes (Q^x)^H)(I^y \otimes \Lambda^x + \Lambda^y \otimes I^x)((Q^y)^H \otimes Q^x)$$

$$I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x = ((Q^y)^H \otimes (Q^x)^H)(I^y \otimes \Lambda^x + \Lambda^y \otimes I^x)(Q^y \otimes Q^x)$$

This is the separability of 2D DFT.
Since $vec(AXB) = (B^T \otimes A)vec(X)$ and (Q is symmetric), we have

$$\begin{aligned} (Q^H \otimes Q^H)vec(M) &= vec(Q^H M Q^H), \\ (Q \otimes Q)vec(M) &= vec(QMQ). \end{aligned} \tag{21}$$

Then we can compute $Q^H M Q^H$ in this way,

$$Q^H M Q^H = \frac{1}{n}\Phi^H M \Phi^H = \frac{1}{n}\Phi^H \left(\Phi M^H\right)^H \tag{22}$$

and compute QMQ similarly,

$$QMQ = \Phi M(\frac{1}{n}\Phi) = \Phi[(\frac{1}{n}\Phi^H)M^H]^H. \tag{23}$$

Note that Φ is DFT, $\frac{1}{n}\Phi^H$ is iDFT.

4 FFT-based Numerical Methods in Three Dimensions

Let $\Omega = \{x_b < x < x_e, y_b < y < y_e, z_b < z < z_e\}$. We present the case with a Dirichlet boundary condition as $u = g$ on $\partial\Omega$ and discussions on other boundary condition cases simply follow. Similar to solving the two-dimensional system, we denote h_x, h_y, h_z as the spatial step size, and N_x, N_γ, N_z as the number of grid intervals in x, y, z direction, respectively.

Set $u_{i,j,k} = u_{i,j,k}(t) \approx u(t, x_i, y_j, z_k)$ for $0 \le i \le N_x, 0 \le j \le N_y$ and $0 \le k \le N_z$. Denote the unknowns as a three-dimensional array $\mathbf{U} = (u_{i,j,k})_{(N_x-1)\times(N_y-1)\times(N_z-1)}$. Define the vector of $\mathbf{U}$:

$$\mathrm{vec}_2(\mathbf{U}) = \begin{pmatrix} \mathrm{vec}(\mathbf{U}(:,:,1)) \\ \mathrm{vec}(\mathbf{U}(:,:,2)) \\ \vdots \\ \mathrm{vec}(\mathbf{U}(:,:,n_3)) \end{pmatrix}.$$

Then it is easy to check

$$\left(I^z \otimes (I^y \otimes \Delta_h^x + \Delta_h^y \otimes I^x) + \Delta_h^z \otimes (I^y \otimes I^x)\right) \mathrm{vec}\,(\mathbf{U}) \approx \Delta(\mathrm{vec}\mathbf{U}). \tag{24}$$

Then

$$\begin{aligned} I \otimes I \otimes A &= \left(QIQ^H\right) \otimes [(Q \otimes Q)(I \otimes \Lambda)(Q^H \otimes Q^H)] \\ &= (Q \otimes Q \otimes Q)(I \otimes I \otimes \Lambda)(Q^H \otimes Q^H \otimes Q^H). \end{aligned} \tag{25}$$

Then

$$\mathrm{vec}_2^{-1}[(Q \otimes Q \otimes Q)\mathrm{vec}_2(\mathbf{U})](r,s,t) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} u_{i,j,k} \cdot e^{-2\pi\mathrm{i}\left(\frac{ri}{n_1}+\frac{sj}{n_2}+\frac{tk}{n_3}\right)}$$

is a 3D FFT.

$$\begin{aligned} & I^z \otimes I^y \otimes \Delta_h^x + I^z \otimes \Delta_h^y \otimes I^x + \Delta_h^z \otimes I^y \otimes I^x \\ &= \left(Q^z I^z (Q^z)^H\right) \left[(Q^y \otimes Q^x)(I^y \otimes \Lambda^x + \Lambda^y \otimes I^x)(Q^y \otimes Q^x)^H\right] \\ &\quad + \Delta_h^z \otimes I^y \otimes I^x \\ &= (Q^z \otimes Q^y \otimes Q^x)(I^z \otimes I^y \otimes \Lambda^x + I^z \otimes \Lambda^y \otimes I^x)(Q^z \otimes Q^y \otimes Q^x)^H \\ &\quad + (Q^z \otimes Q^y \otimes Q^x)(\Lambda^z \otimes I^y \otimes I^x)(Q^z \otimes Q^y \otimes Q^x)^H \end{aligned} \tag{26}$$

$$
\begin{aligned}
&(Q^z \otimes Q^y \otimes Q^x)\mathrm{vec}_2(\mathbf{U}) \\
&= (Q^z \otimes (Q^y \otimes Q^x))\mathrm{vec}\left(\mathrm{vec}(\mathbf{U}(:,:,1)), \mathrm{vec}(\mathbf{U}(:,:,2)), \ldots, \mathrm{vec}(\mathbf{U}(:,:,n_3))\right) \\
&= \mathrm{vec}\left\{(Q^y \otimes Q^x)\left(\mathrm{vec}(\mathbf{U}(:,:,1)), \mathrm{vec}(\mathbf{U}(:,:,2)), \ldots, \mathrm{vec}(\mathbf{U}(:,:,n_3))\right) Q^z\right\} \\
&= \mathrm{vec}\left\{(Q^y \otimes Q^x)\left[(Q^z)^H\left(\mathrm{vec}(\mathbf{U}(:,:,1)), \mathrm{vec}(\mathbf{U}(:,:,2)), \ldots, \mathrm{vec}(\mathbf{U}(:,:,n_3))\right)^H\right]^H\right\}
\end{aligned} \tag{27}
$$

Maybe the matrix form is not the best.

5 Circular Matrix and FFT

Definition 5.1. *A circular matrix is like*

$$
A = \begin{pmatrix}
a_0 & a_1 & a_2 & \cdots & a_{n-1} \\
a_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\
a_{n-2} & a_{n-1} & a_0 & \cdots & a_{n-3} \\
\vdots & \vdots & \vdots & & \vdots \\
a_1 & a_2 & a_3 & \cdots & a_0
\end{pmatrix}. \tag{28}
$$

A special circular matrix is

$$
J = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
1 & 0 & 0 & \cdots & 0
\end{pmatrix}. \tag{29}
$$

It is easy to know that J^k is also a circular matrix.

$$
J^k = \begin{pmatrix} 0 & I_{n-k} \\ I_k & 0 \end{pmatrix} (1 \leq k \leq n). \tag{30}
$$

Thus, $A = a_0 I + a_1 J + ... + a_{n-1} J^{n-1}$. If we define a polynomial $g(x) := a_0 + a_1 x + a_2 x^2 + ... + a_{n-1} x^{n-1}$, then $A = g(J)$.

It is easy to know that the eigenvalues of J are principal roots of order n of uinty $w^k = e^{-\frac{k}{n} 2\pi \mathrm{i}} (k = 0, 1, 2, ..., n-1)$ because $|\lambda I - J| = \lambda^n - 1$. And it is easy to compute that the corresponding eigenvectors are

$$
\vec{\phi_k} = (1, w^k, w^{2k}, ..., w^{(n-1)k})^T, k = 0, 1, 2, ..., n-1. \tag{31}
$$

Therefore, we have the next throrem:

Theorem 5.2. *The eigen-pairs of the circular matrix A is*

$$\begin{aligned} &\lambda_k = g(w^k), && k = 0, 1, 2, ..., n-1, \\ &\vec{\phi_k} = (1, w^k, w^{2k}, ..., w^{(n-1)k})^T, && k = 0, 1, 2, ..., n-1. \end{aligned} \tag{32}$$

Note that $\vec{\phi_k}(k = 0, 1, 2, ..., n-1)$ form the matrix of discrete Fourier transformation (DFT).

Definition 5.3. *For a continuous Fourier transformation* $F(w) = \int_{-\infty}^{+\infty} f(t)e^{-iwt}dt$, *its discrete transformation is*

$$F(w_n) = \sum_{m=0}^{N-1} f(t_m)\, e^{-i\frac{2\pi mn}{N}}, \tag{33}$$

and the inverse discrete Fourier transformation is

$$f(t_m) = \frac{1}{N} \sum_{n=0}^{N-1} F(w_n)\, e^{i\frac{2\pi mn}{N}}. \tag{34}$$

By this definition, the DFT can be written as

$$\begin{bmatrix} F(t_0) \\ F(t_1) \\ \vdots \\ F(t_{N-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N^1 & w_N^2 & \cdots & w_N^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & w_N^{(n-1)} & w_N^{2(n-1)} & \cdots & w_N^{(n-1)^2} \end{bmatrix} \begin{bmatrix} f(w_0) \\ f(w_1) \\ \vdots \\ f(w_{N-1}) \end{bmatrix}, \tag{35}$$

where $w \coloneqq e^{-i\frac{2\pi}{n}}$ is the principal n-th root of unity.

As we all know, the computation can be fasten by FFT. Here is a simple discussion. The starting point is to examine the block structure of an even-order DFT matrix after its columns are reordered so that the odd-indexed columns come first. Consider the case

$$F_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 & w^4 & w^5 & w^6 & w^7 \\ 1 & w^2 & w^4 & w^6 & 1 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w & w^4 & w^7 & w^2 & w^5 \\ 1 & w^4 & 1 & w^4 & 1 & w^4 & 1 & w^4 \\ 1 & w^5 & w^2 & w^7 & w^4 & w & w^6 & w^3 \\ 1 & w^6 & w^4 & w^2 & 1 & w^6 & w^4 & w^2 \\ 1 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w \end{bmatrix} \quad (w = w_8)$$

(Note that w_8 is a root of unity so that high powers simplify.) If $cols = [1,3,5,7,2,4,6,8]$, then

$$F_8(:,cols) = \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w^2 & w^4 & w^6 & w & w^3 & w^5 & w^7 \\ 1 & w^4 & 1 & w^4 & w^2 & w^6 & w^2 & w^6 \\ 1 & w^6 & w^4 & w^2 & w^3 & w & w^7 & w^5 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & w^2 & w^4 & w^6 & -w & -w^3 & -w^5 & -w^7 \\ 1 & w^4 & 1 & w^4 & -w^2 & -w^6 & -w^2 & -w^6 \\ 1 & w^6 & w^4 & w^2 & -w^3 & -w & -w^7 & -w^5 \end{array}\right].$$

The lines through the matrix are there to help us think of $F_8(:,cols)$ as a 2-by-2 matrix with 4 -by- 4 blocks. Noting that $w^2 = w_8^2 = w_4$, we see that

$$F_8(:,cols) = \left[\begin{array}{c|c} F_4 & \Omega_4 F_4 \\ \hline F_4 & -\Omega_4 F_4 \end{array}\right],$$

where $\Omega_4 = \mathrm{diag}\,(1, w_8, w_8^2, w_8^3)$. It follows that if $x \in \mathbb{R}^8$, then

$$F_8 x = F_8(:,\ \mathrm{cols}\ ) \cdot x(\ \mathrm{cols}\ ) =$$

$$\left[\begin{array}{c|c} F_4 & \Omega_4 F_4 \\ \hline F_4 & -\Omega_4 F_4 \end{array}\right] \left[\begin{array}{c} x(1:2:8) \\ \hline x(2:2:8) \end{array}\right] = \left[\begin{array}{c|c} I_4 & \Omega_4 \\ \hline I_4 & -\Omega_4 \end{array}\right] \left[\begin{array}{c} F_4 x(1:2:8) \\ \hline F_4 x(2:2:8) \end{array}\right]$$

Thus, if $N = 2^k$, then the complexity of k-step is $T(k) = 2T(\frac{k}{2}) + O(N)$, so the all complexity is $O(N \log_2 N)$ We can use FFT to compute the eigen decomposition of circular matrice. Firstly, let us see a lemma:

Lemma 5.4. *For principal roots of order n of uinty $w^k = e^{-ik/n} (k = 0,1,2,...,n-1)$, if $w^k \neq 1$, then*

$$1 + w^k + w^{2k} + \cdots + w^{(n-1)k} = 0.$$

It is very simple to prove because

$$0 = (w^k)^n - 1 = (w^k - 1)(1 + w^k + w^{2k} + \cdots + w^{(n-1)k}).$$

Using this lemma, we have

Corollary 5.5. *$\vec{\phi_k} = (1, w^k, w^{2k}, ..., w^{(n-1)k})^T, (k = 0,1,2,...,n-1)$ are conjugate orthogonal.*

Because

$$\vec{\phi_k}^H \vec{\phi_l} = 1 + w^{l-k} + w^{2(l-k)} + \cdots + w^{(n-1)(l-k)} = n\delta_{k,l}.$$

Theorem 5.6. *If we note the DFT matrix as Φ, then the eigen decomposition of the circular matrix C is*

$$C = \frac{1}{\sqrt{n}}\Phi \cdot \Lambda \cdot \frac{1}{\sqrt{n}}\Phi^H = \frac{1}{n}\Phi\Lambda\Phi^H.$$

References

[1] Lili Ju, Jian Zhang, Liyong Zhu, and Qiang Du. Fast Explicit Integration Factor Methods for Semilinear Parabolic Equations. *Journal of Scientific Computing*, 62(2):431–455, 2014. (document)

[2] J. Makhoul. A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):27–34, 1980. 2.2

[3] A. Wiegmann. Fast poisson, fast helmholtz and fast linear elastostatic solvers on rectangular parallelepipeds. Paper LBNL-43565, Lawrence Berkeley National Laboratory, 1999. 2.3