Math 6008 Numerical PDEs–Lecture 15
An introduction to Fourier spectral methods

Instructor: Lei Li, INS, Shanghai Jiao Tong University;
Email: leili2010@sjtu.edu.cn

Recall the DFT (离散傅里叶变换) for a sequence $v = (v_0, \cdots, v_{N-1})$ is given by

$$\hat{v}_k = \sum_{n=0}^{N-1} e^{-ikx_n} v_n, \quad k \in \mathbb{Z},$$

where $x_n = \frac{2\pi}{N} n$. (You can regard v as a function on $[0, 2\pi)$ and x_n is a sample point on the interval.) The inverse DFT is given by

$$v_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{ikx_n} \hat{v}_k = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} e^{ikx_n} \hat{v}_k.$$

We know that they can be computed using FFT in a complexity $O(N \log N)$.

1 Fourier spectral differentiation

For general x, we can define a function

$$f(x) = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} e^{ikx} \hat{v}_k,$$

and then we use $f^{(m)}(x_j)$ to approximate the derivatives $v^{(m)}(x_j)$. The above idea generates the following **Fourier differentiation**:

- Given $v = (v_1, \ldots, v_N)$, compute $\hat{v}$.
- Define $\hat{w}_k = (ik)^m \hat{v}_k$, for $k = -N/2+1, \ldots, N/2$ (or $k = 0, \ldots, N/2, -N/2+1, \ldots, -1$ in Matlab.)
- Compute the inverse DFT (inverse FFT) and get w_k, the real part of which is the approximation of the derivative.

Remark 1. *However, the $k = N/2$ mode is a little bit strange. Assume v is a real array. Then,*

$$\hat{v}_{N/2} = \overline{\hat{v}}_{-N/2} = \overline{\hat{v}}_{N/2}.$$

This means $\hat{v}_{N/2}$ is real. This mode will contribute derivative

$$\hat{v}_{N/2}\frac{iN}{2}e^{i\frac{N}{2}x_j}$$

Clearly, this part is imaginary at the given grid points. Taking real part will get rid of this.

Exercise: *Will it matter if we use $f(x) = \frac{1}{N}\sum_{k=0}^{N-1} e^{ikx}\hat{v}_k$ to compute the derivatives? If it matters, which one is better? Choose a periodic smooth function and code up to check out. Can you explain this? (Hint: Use the Aliasing formula in the next lecture.)*

Remark 2. *As a corollary of the Paserval equality, without taking the real part, the discrete integration by parts by Fourier differentiation holds.*

2 Using DFT(FFT) to solve PDEs with periodic boundary condtions

2.1 Poisson equation

The application to solving elliptic problems will be gone over later for fast Poisson solvers. Here, we just give one simple example to illustrate how it works (section 10.4 in the book of Iserles).

Consider

$$-\Delta u = f, \quad -1 \le x, y \le 1,$$

with periodic boundary condition

$$u(-1, y) = u(1, y), \quad u_x(-1, y) = u_x(1, y),$$
$$u(x, -1) = u(x, 1), \quad u_y(x, -1) = u_y(x, 1).$$

Assume that f satisfies

$$\iint f\,dxdy = 0.$$

To solve this, we makes use of the DFT. With the inverse (2D) DFT, we have

$$f(x_i, y_j) = c_N \sum_k \sum_\ell \hat{f}_{k,\ell} e^{i\pi(kx_i + \ell y_j)}.$$

Hence, we have

$$\hat{u}_{k,\ell} = \frac{1}{\pi^2(k^2+\ell^2)} \hat{f}_{k,\ell}, \quad (k,\ell) \neq (0,0).$$

If $k = \ell = 0$, then the mode $\hat{u}_{0,0}$ means the integral of u. We can impose it to be zero since there is a freedom of constant in the solution. Hence, we can solve the Poisson equation with periodic boundary condition by

- Do 2D FFT for f.
- Divide $\hat{f}_{k,\ell}$ by $\pi^2(k^2+\ell^2)$ for $k, \ell = -N/2+1, \cdots, N/2$. We obtain $\hat{u}_{k,\ell}$.
- Set $\hat{u}_{0,0} = 0$.
- Do inverse FFT for $\hat{u}$ and obtain u (taking real parts).

2.2 Evolutionary equations

Here, we consider the application of DFT for solving evolutionary problems with periodic boundary conditions.

Consider the following problem:

$$u_t + c(x)u_x = 0, \quad c(x) = \frac{1}{5} + \sin^2(x-1), x \in [0, 2\pi].$$

Suppose the problem is with periodic boundary condition.

For periodic boundary condition, we can use DFT/FFT to perform the spatial derivative u_x instead of finite difference to achieve better accuracy in space. (Later we will see that we do not have the spectral accuracy if it's not of periodic boundary condition)

If we apply forward Euler, then we have the following algorithm:

- Take the Fourier Transform of u^n (values at $t = t^n$) and get $\hat{u}^n$
- Multiply ik and take the inverse Fourier Transform. The real part w is the derivative.
- The value at time $t = t^{n+1}$ is $u^{n+1} = u^n - dt \cdot c \cdot w$.

If we apply the leapfrog time discretization, we have

- Take the Fourier Transform of u^n (values at $t = t^n$) and get $\hat{u}^n$
- Multiply ik and take the inverse Fourier Transform. The real part w is the derivative.
- The value at time $t = t^{n+1}$ is $u^{n+1} = u^{n-1} - 2dt \cdot c \cdot w$.

Code presentation: Now, let us show the results obtained by applying the forward-Euler Fourier and leapfrog Fourier for this example.

For different time steps we see that the forward Euler seems to be unstable while the leapfrog method can be stable with suitable time steps.

Time-stepping in Fourier Spectral method

How do we explain the phenomenon observed in the code simulation? (Read Chapter 10 in Trefethen.)

This is just like the stability theory for FDM. We require the time step times eigenvalues (spectra) of the Fourier difference to be in the stability region of the ODE solver (sometimes, the spectra are not enough, and we need the pseudospectra to satisfy this requirement as well).

For the Fourier difference, the e-values are $ik, |k| \leq N/2$. Formally, using the frozen coefficient method, heuristically, $i\frac{N}{2}\max_x |c(x)|dt$ should fall into the stability region of the ODE solver. For the forward Euler, the stability region is $|z+1| \leq 1$, and this requirement is hard to satisfy (need $dt \to 0$ limit). For the leapfrog, the stability region is $(-i, i)$. As long as $dt \sim O(N^{-1})$, this requirement will be satisfied.

We see that we need the solvers to cover some part of the imaginary axis. Some Runga-Kutta methods will satisfy this requirement. For example, explicit RK-p ($p \geq 3$) is good.

Nonperiodic boundary conditions?

If the equations are not of periodic boundary condition, can we use Fourier pseudo-spectral method to solve?

Now, if we solve the equations using Fourier differentiation, we are extending the solution periodically. However, if the given boundary condition is not periodic, after extension, the solution will usually have crusps at the boundary! This means that the extended periodic function is **not smooth**. By the analysis below, the Fourier differentiation will have big aliasing errors due to the existence of large Fourier modes. Hence, Fourier differentiation often is not used for non-periodic boundary conditions.

For Dirichlet BC, one can use the Chebyshev pseudo-spectral method.

Exercise: Try using the Fourier spectral method to solve the KdV equation on torus.

$$u_t + uu_x = \nu u_{xxx}.$$

You may also consider using the time-splitting approach.

3 Spectral accuracy of Fourier pseudo-spectral method** (Free reading)

The spatial accuracy of Fourier pseudo-spectral method is usually $O(h^m)$ $\forall m > 0$ for smooth functions and $O(c^N)$ for analytic functions. This is known as the spectral accuracy. We aim to understand why we have such spectral accuracy

3.1 Aliasing formula

We now consider the effect of sampling. Suppose we discretize the space with step h and $x_j = jh$. Let's recall the semi-discrete Fourier Transform of v, given by

$$\hat{v}(\xi) = h \sum_{j=-\infty}^{\infty} v_j e^{-i\xi x_j}.$$

Note that this is different from the semi-discrete transform we have used in the von-Neumann analysis. The only difference is that we don't have the extra $\frac{1}{\sqrt{2\pi}}$ here. This extra constant doesn't matter.

Theorem 1. *Suppose $u \in L^2(\mathbb{R})$ and has a first derivative with bounded variation. Let $v_j = u(x_j)$. $\hat{v}(\xi)$ is the semi-discrete Fourier transform of v and $\hat{u}(\xi)$ is the Fourier Transform of u. Then, for any $\xi \in [-\pi/h, \pi/h)$, we have*

$$\hat{v}(\xi) = \sum_{j=-\infty}^{\infty} \hat{u}(\xi + 2\pi j/h).$$

Comment: This aliasing formula is related to the Nyquist sampling theorem in signaling processing.

证明. This formula follows easily from the Poisson summation formula

$$2\pi \sum_{n=-\infty}^{\infty} \varphi(2\pi n) = \sum_{n=-\infty}^{\infty} \hat{\varphi}(n)$$

Now, let's define $\varphi(x) = \frac{h}{2\pi} u(\frac{xh}{2\pi}) e^{-i\xi\frac{xh}{2\pi}}$, we find that $\hat{\varphi}(\eta) = \hat{u}(\xi + 2\pi\eta/h)$. The Poisson summation formula then yields the desired result. □

We now consider the aliasing formula for functions defined on the torus. Recall that the DFT is defined by

$$\hat{v}_k = \sum_{n=1}^{N} v_n e^{-ikx_n}$$

Theorem 2. *Suppose u is a good periodic function on $[0, 2\pi)$ such that its Fourier series converges to u. Let $v_j = u(x_j)$. Then, the DFT of v and Fouerier series of u are related by*

$$\hat{v}_k = N \sum_{m=-\infty}^{\infty} \hat{u}_{k+mN}, k = -N/2+1, \ldots, N/2.$$

A different scaling for DFT is used compared with semi-discrete Fourier Transform and the Fourier series for functions on the torus. That's why we have the extra $N = 2\pi/h$ here.

证明. If the Fourier series converges pointwise, then, we have

$$\hat{v}_k = \sum_j u(x_j)e^{-ikx_j} = \sum_j (\sum_p \hat{u}_p e^{ipx_j})e^{-ikx_j} = \sum_p \hat{u}_p \sum_j e^{i(p-k)jh}$$

Clearly, if $p - k = mN$, then the second sum is N; otherwise, the sum is zero. Hence, we have

$$\hat{v}_k = N \sum_m \hat{u}_{k+mN}.$$

□

3.2 Smoothness of a function and the decay of its Fourier Transform

We first consider the Fourier Transform of a function defined on the whole axis.

Theorem 3. *Suppose $u(x) \in L^2(\mathbb{R})$ and $\hat{u}(\xi) = \int u(x)e^{-i\xi x}dx$ is its Fourier transform. Then,*

- *If u has $p-1$ continuous derivatives and are in L^2 for some $p \geq 1$ and a pth derivative of bounded variation, then*

$$\hat{u}(\xi) = O(|\xi|^{-p-1}), |\xi| \to \infty.$$

- *If u has infinitely many continuous derivatives in L^2, then*

$$\hat{u}(\xi) = O(|\xi|^{-m}), \forall m > 0, \text{ as } |\xi| \to \infty.$$

- *If u can be analytically extended to $|Im(z)| < a$ such that there exits a constant C independent of $y \in (-a, a)$, $\|u(\cdot + iy)\|_{L^2} \leq C$, then*

$$e^{a|\xi|}\hat{u}(\xi) \in L^2.$$

Similarly, for a function defined on the torus $[0, 2\pi)$, we introduce the coefficients of Fourier series $\hat{f}_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}dx$ so that $f(x) = \sum_k \hat{f}_k e^{ikx}$.

Theorem 4. *Suppose $u \in C^{(p-1)}([0, 2\pi], \mathbb{C})$, $u^{(j)}(0) = u^{(j)}(2\pi)$ for $j = 0, 1, \ldots, p-1$ and that $u^{(p)}$ is piecewise continuous. Then,*

$$\hat{u}(k) = \frac{\gamma_k}{|k|^p}, \forall k \in \mathbb{N} \quad \sum_k |\gamma_k|^2 < \infty.$$

The proof of the Fourier series case is very straightforward if one notices that we can differentiate term by term.

We are not going to show the proof of the theorems here. The proofs can be found in any Fourier Analysis book. To understand them physically, just remember that a smooth function changes slowly and the high wavenumber modes must be small since they correspond to rapid oscillatory waves.

We are not going to show the proof of the theorems here. The proofs can be found in any Fourier Analysis book.

To understand them physically, just remember that a smooth function changes slowly and the high wavenumber modes must be small since they correspond to rapid oscillatory waves. Hence, **smooth functions gives rapidly decaying Fourier transforms, and rapidly decaying functions gives smooth Fourier transforms.**

3.3 The spectral accuracy

Using the aliasing formula and the decay of Fourier coefficients, one may obtain the following.

For the functions defined on $\mathbb{R}$, we have

Corollary 1. *With the same notations, we have the following claims for any $\xi \in [-\pi/h, \pi/h)$:*

- *If u has $p-1$ continuous derivatives and are in L^2 for some $p \geq 1$ and a pth derivative of bounded variation, then*

$$|\hat{u}(\xi) - \hat{v}(\xi)| = O(h^{p+1}).$$

- *If u has infinitely many continuous derivatives in L^2, then*

$$|\hat{u}(\xi) - \hat{v}(\xi)| = O(h^m), \forall m > 0.$$

- *If u can be analytically extended to $|Im(z)| < a$ such that there exits a constant C independent of $y \in (-a, a)$, $\|u(\cdot + iy)\|_{L^2} \leq C$, then*

$$|\hat{u}(\xi) - \hat{v}(\xi)| = O(e^{-\pi(a-\epsilon)/h}), \forall \epsilon > 0.$$

证明. We can estimate that

$$|\hat{u}(\xi) - \hat{v}(\xi)| \leq \sum_{j \neq 0} |\hat{u}(\xi + 2\pi j/h)| \sim \frac{h}{2\pi} \int_{|\xi| > \pi/h} |\hat{u}(\xi)| d\xi$$

In the first case, we have $Ch \int_{|\xi|>\pi/h} \frac{1}{|\xi|^{p+1}} d\xi = O(h^{p+1})$. The other cases follow similarly. □

For the periodic case, we then have the following corollary:

Corollary 2. *Suppose $u \in C^{(p-1)}([0, 2\pi], \mathbb{C})$, $u^{(j)}(0) = u^{(j)}(2\pi)$ for $j = 0, 1, \ldots, p-1$ and that $u^{(p)}$ is piecewise continuous. Let $\hat{v}_k$ be the DFT of its sampling and $\hat{u}_k$ be its Fourier series coefficients. Then,*

$$|\hat{u}(k) - \frac{1}{N}\hat{v}_k| = O(N^{-p}) = O(h^p).$$

Now, we turn to the Fourier differentiation. For the semi-discrete Fourier transform, suppose $\hat{w}(\xi) = (ik)^\nu \hat{v}(\xi)$ for $\xi \in [-\pi/h, \pi/h)$ and $v_j = u(x_j)$. Then, we have the following theorem:

Theorem 5. *For any $\xi \in [-\pi/h, \pi/h)$:*

- *If u has $p-1$ continuous derivatives and are in L^2 for some $p \geq \nu+1$ and a pth derivative of bounded variation, then*

$$|w_j - u^{(\nu)}(x_j)| = O(h^{p-\nu}).$$

- *If u has infinitely many continuous derivatives in L^2, then*

$$|w_j - u^{(\nu)}(x_j)| = O(h^m), \forall m > 0.$$

- *If u can be analytically extended to $|Im(z)| < a$ such that there exits a constant C independent of $y \in (-a, a)$, $\|u(\cdot + iy)\|_{L^2} \leq C$, then*

$$|w_j - u^{(\nu)}(x_j)| = O(e^{-\pi(a-\epsilon)/h}), \forall \epsilon > 0.$$

Similarly, we have

Theorem 6. *Suppose $u \in C^{(p-1)}([0, 2\pi], \mathbb{C})$, $u^{(j)}(0) = u^{(j)}(2\pi)$ for $j = 0, 1, \ldots, p-1$ and that $u^{(p)}$ is piecewise continuous. Let $\hat{v}_k$ be the DFT of its sampling and $\hat{u}_k$ be the Fourier series coefficients. Suppose $\hat{w} = (ik)^\nu \hat{v}$ and $\nu \leq p-1$. Then,*

$$|w_j - u^{(\nu)}(x_j)| = O(h^{p-\nu}).$$

The proof follows easily from the aliasing formulas and the decay of fourier coefficients for smooth functions. We would like to omit here.

4 Discrete Sine and Cosine transform

The DFT is useful for periodic boundary conditions. If we have Dirichlet boundary condition for $x \in [0, 1]$. Then, we have $u_0 = u_N = 0$, we may consider the discrete sine transform

$$s_k = \sum_{j=0}^{N-1} u_j \sin(\frac{j\pi k}{N}),$$

corresponding to the discrete sampling of the function $\sin(k\pi x)$ with period 1 and step size $h = 1/N$ (or equivalently, sampling of $\sin(kx)$ with period π and step size $h = \pi/N$).

The sine transform can be computed fast using FFT. In fact, with the extended data by defining

$$f_{2N-j} = -f_j, \quad j = 0, \cdots, N-1,$$

then the DFT of the new data equals $2is_k$. The DFT can be computed fast!

For Neumann boundary conditions on $[0, 1]$, one may use the discrete cosine transform (DCT). The discrete consine transform sometimes is better done using the grid $x_{j+1/2}$

$$c_k = \sum_{j=0}^{N-1} u_j \cos\left(\frac{(j+1/2)\pi k}{N}\right).$$

Note that the fast consine transform can also be computed similarly using DFT.

5 Fast Poisson solvers

As we have seen, if the Poisson equation is equipped with periodic boundary conditions, the solution can be found using FFT and IFFT fast using the fact that the Fourier modes are the eigenfunctions. How about Dirichlet and Neumann boundary conditions? Can the five-point and nine-point schemes be solved fast?

Different from the spectral method above where we use the spectral expansion to do the numerical differentiation, here we use the spectral expansion to solve some FDM and some discrete schemes. In some sense, this is also spectral method, but for solving some linear systems.

5.1 Some special cases

Consider first the following:

$$\begin{cases} -\Delta u = f, \quad x \in \Omega = (0,1)^2, \\ u = 0 \quad x \in \partial\Omega. \end{cases}$$

We discretize the equations using the five-point scheme:

$$-\Delta_h u_{k\ell} = f_{k\ell}.$$

Note that this is an FDM scheme not with periodic boundary condition. Can we use the DFT to solve this?

Recall that the eigenfunctions of the discrete Laplacian here are given by

$$v^{\alpha\beta}_{k,\ell} = \sin\left(\frac{k\alpha\pi}{m+1}\right)\sin\left(\frac{\ell\beta\pi}{m+1}\right)$$

and the eigenvalues are

$$\lambda_{\alpha\beta} = \frac{4}{h^2}\left\{\sin^2(\frac{\alpha\pi}{2(m+1)}) + \sin^2(\frac{\beta\pi}{2(m+1)})\right\}.$$

Hence, to solve this, we may expand u and f in terms of the eigenfunctions:

$$u = \sum c_{\alpha\beta} v^{\alpha\beta}, \quad f = \sum f_{\alpha\beta} v^{\alpha\beta}.$$

Since $-\Delta_h u = \sum_{\alpha\beta} c_{\alpha\beta}\lambda_{\alpha\beta} v^{\alpha\beta}$, we find

$$c_{\alpha\beta}\lambda_{\alpha\beta} = f_{\alpha\beta}.$$

Hence, we find $c_{\alpha\beta}$. This means that if we can find the coefficients $f_{\alpha\beta}$

and do the sum fast, we can solve this equation fast. This is actually possible as we can use FFT to do the 2D sine transform!

How about the Neumann boundary condition?

$$\begin{cases} -\Delta u = f, \quad x \in \Omega = (0,1)^2, \\ \frac{\partial u}{\partial n} = 0 \quad x \in \partial\Omega. \end{cases}$$

Now, we instead use the cosine functions. However, it is found that locating the solutions in the center of the cells is more convenient (if not, still doable but more involved). Consider the five-point scheme. The eigenfunctions can be written as

$$v_{k,\ell}^{\alpha\beta} = \cos(\alpha\pi x_{k+1/2})\cos(\beta\pi y_{\ell+1/2}),$$

where $x_{k+1/2} = (k+\frac{1}{2})h = \frac{k+1/2}{m+1}$. One can show that it is indeed an eigenfunction. Then, the equation can be solved in a similar fashine.

The matrix formulation for the 5-point Poisson

Let us now check what the 2D sine transform is doing for the Dirichlet boundary condition in the disguise of the matrix formulation. Consider the 5-point method.

Recall that the 5-point method is $AU = F$ where $A = \frac{1}{h^2}(I \otimes A_1 + A_1 \otimes I)$ where I_1 is the $m \times m$ identity matrix and

$$A_1 = \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}_{m\times m}.$$

If we define the matrix

$$U_m = \begin{pmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & & \vdots \\ u_{m1} & \cdots & u_{mm} \end{pmatrix}_{m\times m},$$

the equations are

$$U_m A_1 + A_1 U_m = B$$

for some B.

Note that A_1 matrix is a so-called TST (Toeplitz, symmetric and tridiagonal) matrix. This corresponds to the discretization of the 1D second order derivative with Dirichlet boundary condition. The eigenvalues are given by the discrete sine functions. Hence, one can construct an orthogonal matrix Q using the eigenfunctions by

$$Q_{j,\alpha} = \sqrt{\frac{2}{m+1}} \sin(\frac{j\alpha\pi}{m+1}).$$

Then, it is easy to see that $Q = Q^T = Q^{-1}$. Also, $Q^T A_1 Q = D$, where D is the diagonal matrix with the eigenvalues on the diagonal.

Clearly,

- $Q^T v$ for a column vector v can be interpretated as the sine transform of v (up to the factor $\sqrt{\frac{2}{m+1}}$).
- For a row vector a, aQ is also the sine transform of a.

Using the eigendecomposition $Q^T A_1 Q = D$, one has

$$UVDV^{-1} + VDV^{-1}U = B.$$

Denote $Q^T U Q$ and $Q^T B Q$ by $\tilde{U}$ and $\tilde{B}$ respectively. Then,

$$\tilde{U}D + D\tilde{U} = \tilde{B}.$$

So

$$\tilde{B}_{ij} = (\tilde{U}D + D\tilde{U})_{ij} = \tilde{U}_{ij}d_j + d_i\tilde{U}_{ij}.$$

The equations are diagonalized directly. That in fact, the quantity $Q^T U Q$ is doing the 2D sine transform.

5.2 The fast Poisson solvers in the general form

Now, we aim to see how we can generalize the matrix formulation above for 5-point to general schemes, and obtain the fast Poisson solvers. This is then applicable to the 9-point scheme.

Discretizing the Poisson Equation usually yields the so-called TST matrices

$$A = \begin{pmatrix} \alpha & \beta & & \\ \beta & \ddots & \ddots & \\ & \ddots & \ddots & \beta \\ & & \beta & \alpha \end{pmatrix}_{m\times m}.$$

We have seen that (see also section 12.2 of the book by Iserles) the eigenvalues are given by

$$\lambda_j = \alpha + 2\beta\cos(\frac{j\pi}{m+1})$$

and the eigenvectors are $v_j = \sin(k\pi\frac{j}{m+1})$. Since the eigenvectors are all the same, hence **all TST matrices of the same size commute**.

If we discretize the 2D poisson equation with five-point or nine-point methods, we will get **block TST matrices**:

$$A = \begin{bmatrix} S & T & & \\ T & \ddots & \ddots & \\ & \ddots & \ddots & T \\ & & T & S \end{bmatrix}.$$

For example, the 5-point matrix is (ignoring the $1/h^2$ factor)

$$K_2 = A_1 \otimes I_1 + I_1 \otimes A_1$$

so that $T = -I$ and $S = 2I + A_1$.

We denote again

$$u_j = (u_{1j}, u_{2j}, \cdots, u_{mj})^T.$$

Then, the equations can be written as

$$Tu_{j-1} + Su_j + Tu_{j+1} = b_j.$$

Using the eigendecomposition, $TQ = QD_T$, $SQ = QD_S$, one then has

$$D_T Q^T u_{j-1} + D_S Q^T u_j + D_T Q^T u_{j+1} = Q^T b_j.$$

Defining $y_j = Q^T u_j$, $c_j = Q^T b_j$, one then has

$$D_T y_{j-1} + D_S y_j + D_T y_{j+1} = c_j.$$

This means that if we do sine transform for the vectors u_j (or in the x direction), one will then have the system of equations above.

This system of equations is interesting because if we write out the components

$$y_j = (y_{1j}, y_{2j}, \cdots, y_{mj})^T,$$

we will have

$$\lambda_i^T y_{i,j-1} + \lambda_i^S y_{i,j-1} + \lambda_i^T y_{i,j+1} = c_{i,j}.$$

This means if we fix i, then we will have an $m\times m$ system, with the coefficient being TST:

$$\Gamma_i = \begin{pmatrix} \lambda_i^S & \lambda_i^T & & \\ \lambda_i^T & \ddots & \ddots & \\ & \ddots & \ddots & \lambda_i^T \\ & & \lambda_i^T & \lambda_i^S \end{pmatrix}_{m\times m}.$$

What are the rows of Y? They are the solutions with y variable changing! This means for 2D problem, if we do sine transform in x direction, then the discrete schemes decouples to m 1D problems in the y direction!

In principle, one can do the sine transform again for y direction to decouple again into diagonal systems to solve. However, since the matrix is tridiagonal, solving directly using LU decomposition takes only $\sim 3m$. Hence, this is cheaper compared to the sine transform. Hence, doing the sine transform once is enough and maybe slightly cheaper for general block TST systems (compared with the pseudo-spectral method in both directions)! [For the special case of 2D poisson, the pseudo-spectral method in both directions is clean in coding and it might be preferred as well.]

Remark 3. *As we have seen, $D_T y_{j-1} + D_S y_j + D_T y_{j+1} = c_j$ corresponds to m 1D equations if we consider the rows of $Y = [y_1, \cdots, y_m]$. In the*

5-point method, $D_T y_{j-1} + D_S y_j + D_T y_{j+1} = c_j$ *corresponds to* $D_T = -I$, $D_S = 2I + D$. *Hence,* $\Gamma_j = A_1 + d_j I$ *so that the equations can be reformulated as a compact matrix equation*

$$YA_1 + DY = C,$$

and the equations for the rows are easily seen. We then multiply Q *on the right, let* $\tilde{U} = YQ$*, and then have*

$$\tilde{U}D + D\tilde{U} = \tilde{B},$$

which is the same as in the previous section.

For general T matrix, Γ_j does not have such decomposition so there is no such compact form to write the m 1D equations together into a matrix equation. *However, as we have explained, these* m *1D equations are decoupled and of tridiagonal structure. This means though we do not have the compact matrix equation, the fast Poisson solver for general block TST system has the same complexity and does the same thing.*

5.3 Complexity of fast poisson solver

With fast sine transform (FST), the multiplication between Q (or Q^{-1}) with a vector could be calculated in $O(m \log m)$ so that the matrix multiplication between Q and A can be done in $O(m^2 \log m)$. Therefore, the cost of the algorithm above, which is called the Fast Poisson Solver, is $O(N \log N)$ in total, where $N = m^2$.