Numerical Analysis MTH614

Spring 2012, Korea University

Fourier transform

Solving the heat equation

In the previous lectures, we have studied the finite difference method. Now we consider the Fourier transform method. We also focus on the one dimensional diffusion equation.

Suppose that $\partial_t u(x,t) = u_{xx}(x,t),\ 0 < x < 2\pi,\ t > 0.$ (1)

The equation has to be supplemented by an initial condition $u(x,0) = u_0(x)$ and periodic boundary conditions $u(0,t) = u(2\pi,t)$.

To solve the equation, we use separation of variables. Write $u(x,y) = v(x)w(t)$ and plug this into (1), then we obtain

$$w'(t)v(x) = v''(x)w(t) \tag{2}$$

$$\frac{w'(t)}{w(x)} = \frac{v''(x)}{v(t)}, \tag{3}$$

where we assumed $w(t) \neq 0 \text{ and } v(x) \neq 0.$ Since left hand side depends only on t but the right hand side only on x, we can say that both must be constant.

It turns out that this constant must be non positive, that is $-k^2 > 0$.

$$v''(x) = -k^2 v(x)(x) = c_1 \cos(kx) + c_2 \sin(kx) \tag{4}$$

with constants $c_{1,2} \in \mathbf{R}$,

In same manner,

$$w'(t) = -k^2 w(t) \to w(t) = e^{-k^2 t} w(0). \tag{5}$$

Moreover, to fit the boundary conditions such that $w(2\pi, t) = w(0, t)$, we need $k \in \mathbf{N}$.

Therefore, $u(x,t) = e^{-k^2 t}(c_1 \cos(kx) + c_2 \sin(kx))$ (6)
are solutions of the heat equation, and its linearity deduces that

$$u(x,t) = \sum_{k \in \mathbf{Z}} c_k e^{-k^2 t} e^{ikx}. \tag{7}$$

Discrete Fourier Transform

The discrete Fourier transform is frequently used for computing with finite collections of points. We consider this by the heat equation as follows.

There is the heat equation same as previous but a different domain

$$\begin{aligned} u_t(x,t) &= u_{xx}(x,t) \text{ on } \Omega = (0,\ 2), \\ u(x,0) &= f(x) \text{ on } \Omega = (0,\ 1), \\ u(x,0) &= -f(x-1) \text{ on } \Omega = (1,\ 2). \end{aligned} \tag{8}$$

That means we have periodic boundary conditions over the domain with discrete points

$$h = 2/(N_x+1) \quad \text{and} \quad x_p = \frac{2p}{N_x+1}, \quad p = 0, \cdots, N_x. \tag{9}$$

In the case of the Implicit scheme,

$$\frac{u_p^{n+1} - u_p^n}{\Delta t} = \frac{u_{p+1}^{n+1} - 2u_p^{n+1} + u_{p-1}^{n+1}}{h^2},$$

$$\frac{u_p^{n+1}}{\Delta t} - \frac{u_{p+1}^{n+1} - 2u_p^{n+1} + u_{p-1}^{n+1}}{h^2} = \frac{u_p^n}{\Delta t}. \tag{10}$$

The other hand, we have one dimensional discrete Fourier transform such as

$$\hat{u}_j = \frac{1}{N_x + 1} \sum_{p=0}^{N_x} u_p e^{-i\pi j x_p}, \tag{11}$$

$$u_p = \sum_{j=-k_0}^{k_0+1} \hat{u}_j e^{i\pi j x_p}, \tag{12}$$

where $k_0 = (N_x - 1)/2,\ x_{p+1} = x_p + h,\ x_{p-1} = x_p - h$. (13)

Plug (3) into (1), then we obtain

$$\hat{u}_j^{n+1} \left(\frac{1}{\Delta t} - \frac{e^{i\pi hj} - 2 + e^{-i\pi hj}}{h^2} \right) = \frac{\hat{u}_j^n}{\Delta t} \ . \tag{14}$$

In other words,

$$\hat{u}_j^{n+1} \left(\frac{1}{\Delta t} - \frac{2\cos(\pi hj) - 2}{h^2} \right) = \frac{\hat{u}_j^n}{\Delta t}. \tag{15}$$

Solving for $\hat{u_j^{n+1}}$,

$$\hat{u}_j^{n+1} = \frac{\hat{u}_j^n}{1 - \Delta t(2\cos(\pi h j) - 2)/h^2}. \tag{16}$$

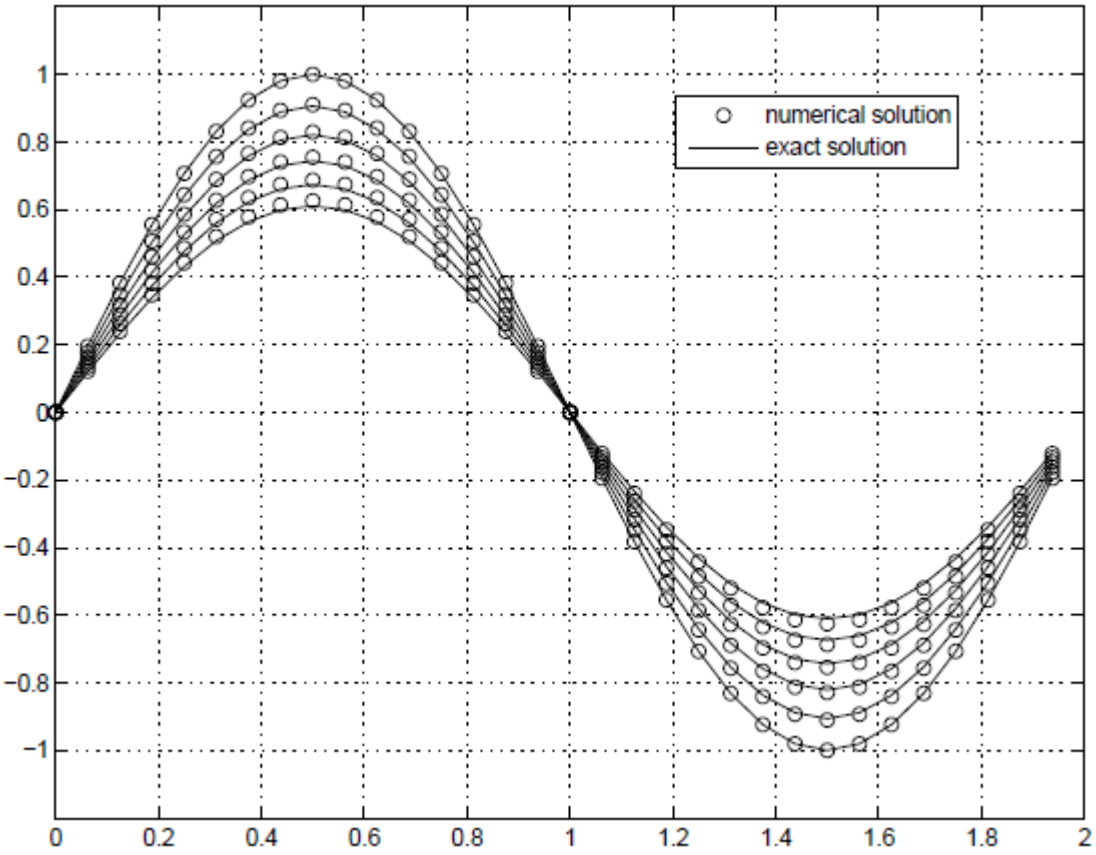
For example, we implement MATLAB code with implicit scheme.

Initial condition →

$$\hat{u}_j = \frac{1}{N_x + 1} \sum_{p=0}^{N_x} u_p e^{-i\pi j x_p},$$

$$u_p = \sum_{j=-k_0}^{k_0+1} \hat{u}_j e^{i\pi j x_p},$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%% dft_heat1d.m %%%%%%%%%%%%%%%%%%%%%%%%%%
clear; clc; a=2; L=31; k0=(L-1)/2;
x=linspace(0,a*L/(L+1),L+1); h=x(2)-x(1); dt=0.01;
f=sin(pi*x); figure(1); clf; plot(x,f,'ko',x,f,'k'); hold
for k=1:5
for j=1:L+1
   un(j)=1/(L+1)*sum(f.*exp(-i*2*pi*(j-1-k0)*x/a)) ...
          /(1-dt*(2*cos(pi*h*(j-1-k0))-2)/h^2);
end
unp=0*x;
for s=1:L+1
   unp=unp+un(s)*exp(i*2*pi*(s-1-k0)*x/a);
end
plot(x,real(unp),'ko',x,sin(pi*x)*exp(-pi^2*dt*k),'k')
f=unp;
end
axis([0 a -1.2 1.2]);
legend('numerical solution','exact solution',2);grid
```


numerical solution
exact solution
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Fast Fourier Transform

Fast Fourier transform(FFT) is a faster version of the Discrete Fourier Transform(DFT). The FFT utilizes some clever algorithms to do the same thing as the DTF.

We see the below.

Let f and $\hat{f}$ be periodic, and the transform can be rewritten as

$$\begin{aligned} \tilde{f}_k &= \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}, k = 0, \cdots, N-1 \\ f_j &= \sum_{k=0}^{N-1} \tilde{f}_k e^{ikx_j}, j = 0, \cdots, N-1. \end{aligned} \tag{17}$$

The transformation matrix F_N of the discrete Fourier transform is

$$\mathcal{F}_N = \left(\omega^{ij}\right)_{\substack{0 \le i < N \\ 0 \le j < N}}, \ \omega = e^{-2\pi\sqrt{-1}/N}. \tag{18}$$

Let us suppose N is even. Define $I_N f$,the trigonometric interpolant of f at $x_0, \ldots, x_{N-1}$ by

$$I_N f(x) = \frac{1+e^{iNx/2}}{2} I^e_{N/2} f(x) + \frac{1-e^{iNx/2}}{2} I^o_{N/2} f(x - 2\pi/N). \tag{19}$$

Here, $I^e_{N/2} f$ and $I^o_{N/2} f$ be the trigonometric interpolant of f at even and odd points. This is because

$$e^{iNx_j/2} = \begin{cases} 1 & \text{for } j \text{ is even} \\ -1 & \text{for } j \text{ is odd} \end{cases} \tag{20}$$

Now we define by

$$\begin{aligned} I^e_{N/2} f &= \sum_{k=0}^{N/2-1} A_k E_k(x) \\ I^o_{N/2} f &= \sum_{k=0}^{N/2-1} B_k E_k(x) \\ I_N f &= \sum_{k=0}^{N-1} C_k E_k(x), \end{aligned}$$

Then,

$$C_k = \begin{cases} \frac{1}{2}(A_k + \omega_N^k B_k) & \text{for } 0 \le k < N/2 \\ \frac{1}{2}(A_{k-N/2} - \omega_N^k B_{k-N/2}) & \text{for } N/2 \le k < N. \end{cases} \quad (21)$$

Therefore, F_N can be splitted into

$$\mathcal{F}_N = Q_N \begin{bmatrix} \mathcal{F}_{N/2} & 0 \\ 0 & \mathcal{F}_{N/2} \end{bmatrix} P_N \ . \quad (22)$$

Here P_N is a permutation matrix which maps

$$(f_0, f_1, \cdots, f_{N-1})^t \mapsto (f_0, f_2, \cdots, f_{N-2}, f_1, f_3, \cdots, f_{N-1})^t;$$

The matrix Q_N is defined as

$$Q_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix}, \ I : \text{identity matrix}, D_{N/2} = \text{diag}(1, \omega, \cdots, \omega^{N/2-1})$$

This means that use each n-point Fourier transform as a sum of two n/2-point Fourier Transform.

To approximate solution of the heat equation, we use a representation of u by orthogonal eigenfunctions of the operator.

Then the representation is $u(x) = F^{-1}(\hat{u}_k)(x) = \sum_{k \in Z} \hat{u}_k e^{ikx}$, (23)

and the Fourier coefficients are given by $\hat{u}_k = (Fu)_k = \frac{1}{2\pi} \int_\Omega u(x) e^{-ikx} dx.$ (24)

If we set the Laplace operator over the domain $\Omega = (0, 2\pi)$ with periodic boundary conditions, the Fourier modes are given by $\triangle e^{ikx} = -k^2 e^{ikx}$ (25)

Thus, for the equation $\partial_t u(x,t) = u_{xx}(x,t)$ we have the following setup.

$$\frac{1}{h}(\hat{u}_k^{m+1} - \hat{u}_k^m) \approx \partial_t \hat{u}_k(t_m) \approx -k^2[(1-\theta)\hat{u}_k^m + \theta \hat{u}_k^{m+1}] \text{ where } \theta \in [0,1]. \quad (26)$$

Solving for $\hat{u}_k^{m+1}$ yields

$$\hat{u}_k^{m+1} = \mu_k \hat{u}_k^m \text{ with so called multipliers } \mu_k = \frac{1-(1-\theta)hk^2}{1+h\theta k^2}. \quad (27)$$

The observation of $\mu_k = \dfrac{1-(1-\theta)hk^2}{1+h\theta k^2}$ is that $|\mu_k| \leq 1$ for $\theta \geq 1/2$ such that the solution can never grow in time, and the scheme is implicit with $\theta = 1,$ which excludes oscillations in time of the Fourier coefficients.

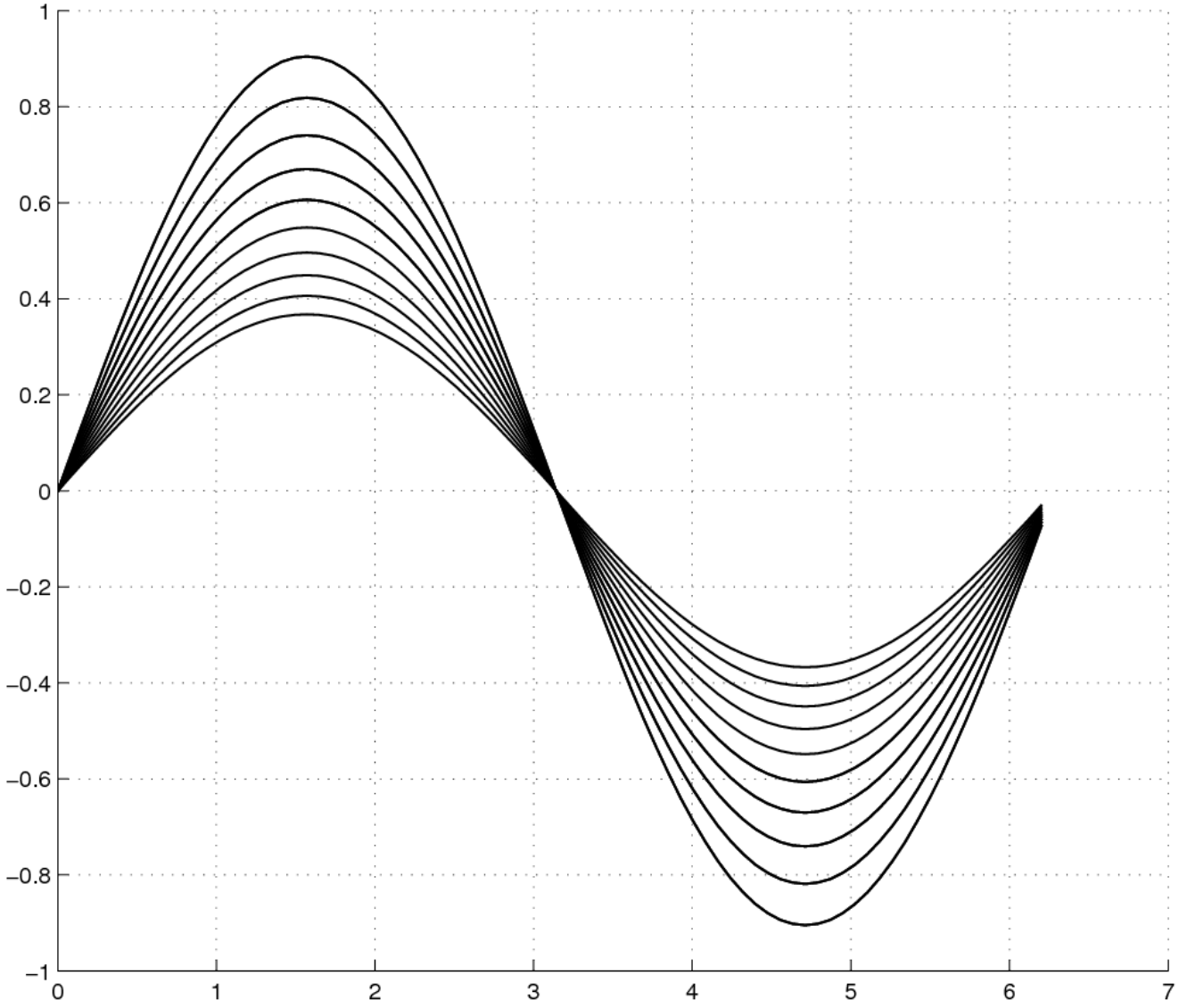
For the numerics, we truncate the Fourier series of u to approximate

$$u(x,t) = \sum_{k<n/2} \hat{u}_k(t) e^{ikx} \text{ with suitable } n.$$

In MATLAB, let $u = (u_1, u_2, \ldots, u_n)$ be an n-vector, n even, to be read as a sampling of a 2π periodic function u on a domain of length 2π. Then command d=fft(u) produces a complex n-vector $(d_1, \ldots, d_n)$. And also we use MATLAB command fftshift which rearrange index to fit the FFT.

```
clc;clear;
n=20;
dx=2*pi/n;
x=0:dx:2*pi-dx;theta=0.5;h=0.1;
pstep=11;tmax=h*(pstep-1);
t=0:h:h*(pstep-1);ua=zeros(pstep,n);
nv=fftshift(-n/2:1:n/2-1);
mu=(1-h*(1-theta)*nv.^2)./(1+theta*nv.^2*h);
u0=sin(x);
ua(1,:)=u0;
u=u0;
un=0*u0;
uf=fft(u);
for i=2:pstep
    uf=mu.*uf;
    u=ifft(uf);
    ua(i,:)=u;
     hold on
    plot(x,ua(i,:),'-');
end
```

$$\hat{u}_k^{m+1} = \mu_k \hat{u}_k^m \text{ with } \mu_k = \frac{1-(1-\theta)hk^2}{1+h\theta k^2}.$$


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
1
2
3
4
5
6
7