1 Finite Difference Method for Heat Equation on $\mathbb{R}$

$$\begin{aligned} &u_t(x,t) = k u_{xx}(x,t), \quad -\infty < x < \infty, \quad t > 0 \\ &u(x,0) = f(x), \quad -\infty < x < \infty, \\ &|u(x,t)| < M < \infty \quad \text{for some} \quad M \end{aligned} \tag{1.1}$$

We know that the solution is given by

$$u(x,t) = \frac{1}{\sqrt{4k\pi t}} \int_{-\infty}^{\infty} e^{-(x-y)^2/(4kt)}\, f(y)\, dy. \tag{1.2}$$

On the other hand this integral is not easy to evaluate explicitly and not so easy to approximate. So in these notes we present a numerical method for approximating the solution.

First we consider some numerical approximations to the derivative. If $f \in C^2$ then by Taylor's theorem for some ξ with $x < \xi < (x+h)$ we g=have

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2} f''(\xi)h^2$$

which implies

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{2} f''(\xi)h,$$

or

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h).$$

Here $\ell(h) = \mathcal{O}(h)$ means $\ell(h)/h$ is bounded near $h = 0$.

For $h > 0$ the expression $\dfrac{f(x+h) - f(x)}{h}$ is called a forward-difference approximation. Similarly $\dfrac{f(x) - f(x-h)}{h}$ is called a backward-difference approximation.

We will need even higher order approximations for $f'(x)$ and $f''(x)$. To this end we write more precise Taylor formula approximations:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \frac{f^{(iv)}(\xi_1)}{24}h^4 \tag{1.3}$$

which implies

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(x)}{2}h + \mathcal{O}(h^2)$$

and similarly

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(x)}{6}h^3 + \frac{f^{(iv)}(\xi_2)}{24}h^4 \tag{1.4}$$

which implies

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{f''(x)}{2}h + \mathcal{O}(h^2).$$

Thus for $f'(x)$ we can write

$$2f'(x) = \frac{f(x+h) - f(x)}{h} + \frac{f(x) - f(x-h)}{h} + \frac{f''(x)}{2}h - \frac{f''(x)}{2}h + \mathcal{O}(h^2)$$

or

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2). \tag{1.5}$$

This formula is the center-difference formula which gives a second order approximation.

We now seek a second order approximation formula for $f''(x)$. Namely we will show that

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2). \tag{1.6}$$

To obtain this formula we solve (1.3) and (1.4) for $\frac{f''(x)}{2}h^2$ which gives

$$\frac{f''(x)}{2}h^2 = f(x+h) - f(x) - f'(x)h - \frac{f'''(x)}{6}h^3 - \frac{f^{(iv)}(\xi_1)}{24}h^4$$

and

$$\frac{f''(x)}{2}h^2 = f(x+h) - f(x) + f'(x)h + \frac{f'''(x)}{6}h^3 - \frac{f^{(iv)}(\xi_2)}{24}h^4.$$

Adding these expression and dividing by h^2 gives our desired result (1.6).

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{f^{(iv)}(\xi_1)}{24}h^2 - \frac{f^{(iv)}(\xi_2)}{24}h^2.$$

We now return to the heat equation. Our main assumption here is that the solution satisfies $u(x,t) \to 0$ as $|x| \to \infty$ for all $t > 0$. We notice that this is certainly true if the initial data has compact support, i.e., if the initial temperature is zero for large x.

Thus for a strict solution we can apply (1.6) to write

$$u_{xx} = \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} + \mathcal{O}(h^2) \equiv J(x,t) + \mathcal{O}(h^2).$$

This implies

$$u_t(x,t) \approx \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2}. \tag{1.7}$$

At this point let us recall a numerical integration procedure called the mid-point (or trapezoid) rule which gives

$$\int_{t_0}^{t_0+\Delta t} g(t)\,dt = \frac{\Delta t}{2}(g(t_0) + g(t_0 + \Delta t)) + \mathcal{O}((\Delta t)^2).$$

This formula follows from the Taylor formula once again. For $t_0 < t < t_1 = t_0 + \Delta t$

$$g(t) = g(t_0) + g'(t_0)(t - t_0) + \frac{g''(t_0)}{2}(t - t_0)^2 + \cdots$$

we can also write

$$g(t) = g(t_1) + g'(t_1)(t - t_1) + \frac{g''(t_1)}{2}(t - t_1)^2 + \cdots$$

Plugging each of these into the integral we obtain

$$\begin{aligned}\int_{t_0}^{t_1} g(t)\,dt &= \left[g(t_0)t + g'(t_0)\frac{(t-t_0)^2}{2} + \frac{g''(t_0)}{2}\frac{(t-t_0)^3}{3} + \cdots\right]\Bigg|_{t_0}^{t_1} \\ &= g(t_0)(\Delta t) + g'(t_0)\frac{(\Delta t)^2}{2} + \frac{g''(t_0)}{2}\frac{(\Delta t)^3}{3} + \cdots\end{aligned}$$

and for the second form we have

$$\begin{aligned}\int_{t_0}^{t_1} g(t)\,dt &= \left[g(t_1)t + g'(t_1)\frac{(t-t_1)^2}{2} + \frac{g''(t_1)}{2}\frac{(t-t_1)^3}{3} + \cdots\right]\Bigg|_{t_0}^{t_1} \\ &= g(t_1)(\Delta t) - g'(t_1)\frac{(\Delta t)^2}{2} + \frac{g''(t_1)}{2}\frac{(\Delta t)^3}{3} + \cdots\end{aligned}$$

Take the average of these last two formulas to obtain

$$\int_{t_0}^{t_1} g(t)\,dt = \frac{(\Delta t)}{2}[g(t_0) + g(t_1)] + [g''(t_0) + g''(t_1)]\,\frac{(\Delta t)^3}{12} + \cdots$$

Thus we conclude

$$\int_{t_0}^{t_1} g(t)\,dt = \frac{(\Delta t)}{2}[g(t_0) + g(t_1)] + \mathcal{O}((\Delta t)^3).$$

So integrating (1.7) from t to $(t + \Delta t)$ we have

$$u(x, t + \Delta t) \approx u(x, t) + \frac{(\Delta t)}{2}\left(J(x, t) + J(x, t + \Delta t)\right). \tag{1.8}$$

We can write (1.8) as

$$u(x, t + \Delta t) - \frac{\Delta t}{2}J(x, t + \Delta t) = u(x, t) + \frac{\Delta t}{2}J(x, t) \tag{1.9}$$

.

Now we discretize the interval $-L \le x \le L$ by

$$x_j = jh, \quad j = -NL, \cdots, NK, \quad h = 1/N$$

and introduce the discrete time steps $t_k = k(\Delta t), \quad k = 0, 1, 2, \cdots$. Let us introduce the notation $u_{j,k}$ to be an approximate value of $u(x_j, t_k)$. There are two special cases, $j = -NL$ and $j = NL$, in which we need to make some further assumptions in our formula for $J(x, t)$.

In these cases we make use the assumption that $u(x,t)$ is going to zero and $|x| \to \infty$ and replace the terms $u(x_{-NL-1},t)$ and $u(x_{NL+1},t)$ by zero. So with this approximation we have

$$J(x_{-NL},t_k) = \frac{u(x_{-NL+1},t_k) - 2u(x_{-NL},t_k) + u(x_{-NL-1},t_k)}{h^2} \approx \frac{u(x_{-NL+1},t_k) - 2u(x_{-NL},t_k)}{h^2}$$

and

$$J(x_{NL},t_k) = \frac{u(x_{NL+1},t_k) - 2u(x_{NL},t_k) + u(x_{NL-1},t_k)}{h^2} \approx \frac{-2u(x_{-NL},t_k) + u(x_{NL-1},t_k)}{h^2}.$$

For all j satisfying $-NL < j < NL$ we have

$$J(x_j,t_k) = \frac{u(x_{j+1},t_k) - 2u(x_j,t_k) + u(x_{j-1},t_k)}{h^2}.$$

Then we can write the formula (1.8) in a matrix notation with

$$\boldsymbol{U}_k = \begin{bmatrix} u_{-NL} \\ \vdots \\ u_{-1,k} \\ u_{0,k} \\ u_{1,k} \\ \vdots \\ u_{NL,k} \end{bmatrix}^\top, \quad M = \begin{bmatrix} -2 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & & \ddots & \vdots \\ 0 & 1 & -2 & & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & & 1 \\ 0 & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix}$$

With this (1.9) can be written as

$$\left(I_{2NL+1} - \frac{\Delta t}{2} M\right) \boldsymbol{U}_k = \left(I_{2NL+1} + \frac{\Delta t}{2} M\right) \boldsymbol{U}_{k-1}.$$

Let us finally define

$$\mathcal{A}_\pm = \left(I_{2NL+1} \pm \frac{\Delta t}{2} M\right), \qquad \mathcal{A} = (\mathcal{A}_-)^{-1}\mathcal{A}_+$$

then we obtain the time marching scheme called the Crank-Nichols Method

$$\boldsymbol{U}_k = \mathcal{A}\boldsymbol{U}_{k-1}, \quad k = 1, 2, \cdots, \tag{1.10}$$

where

$$\boldsymbol{U}_0 = \left[f(x_{-NL}), \quad \cdots, \quad f(x_{-1}), \quad f(x_0), \quad f(x_1), \quad \cdots, \quad f(x_{NL})\right]^\top.$$

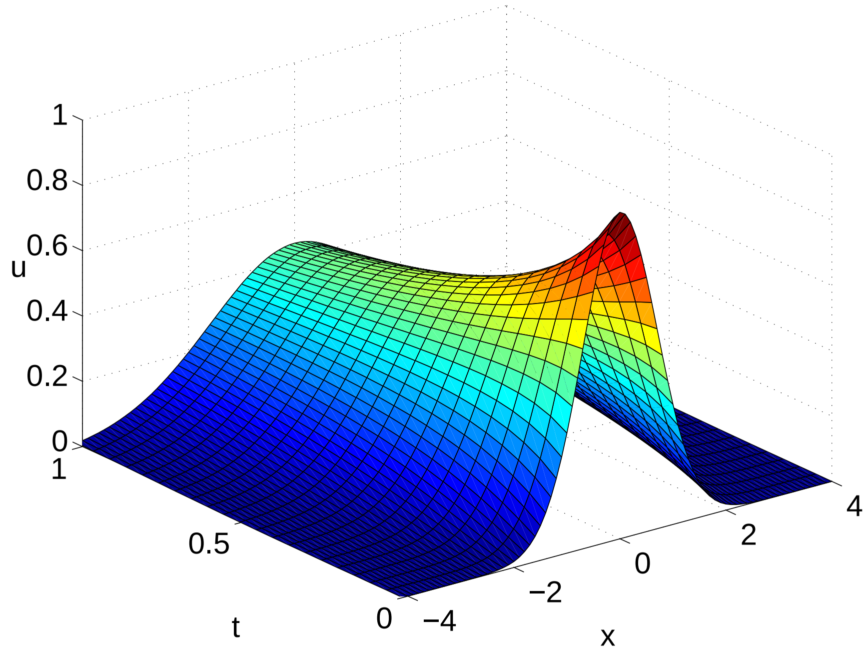
In the homework we saw that

$$u(x,t) = \frac{1}{(1+4t)^{1/2}} \exp\left(-\frac{x^2}{(1+4t)}\right)$$

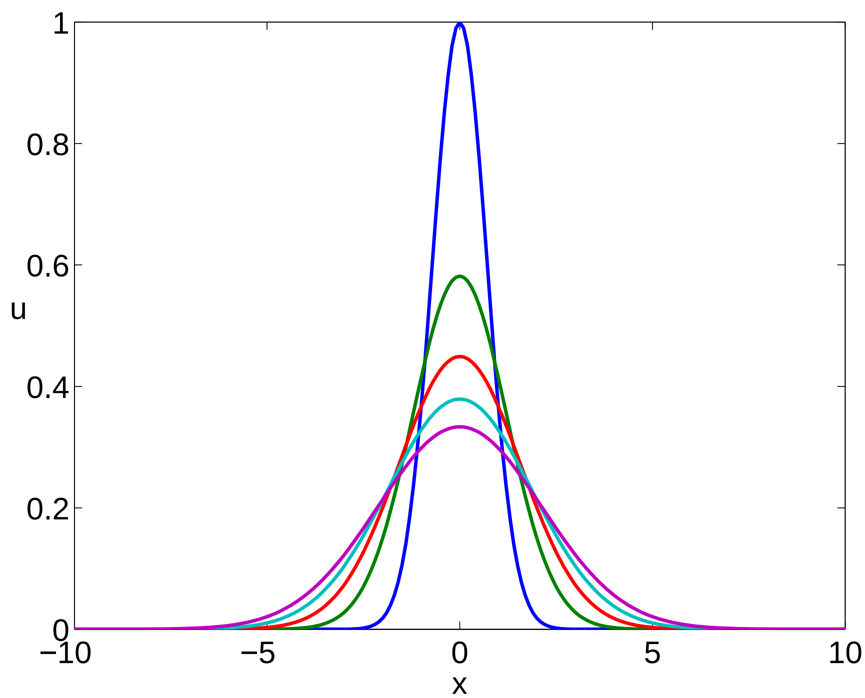
is a solution to the heat equation with $k = 1$ for the initial condition

$$u(x,0) = f(x) = e^{-x^2}.$$

The following figures correspond to our Finite Difference Method with $L = 10$ and $N = 10$. The maximum error for this example at the grid points is on the order of .001.



Numerical Solution Surface



Exact and Numerical Solutions: $t = 0, 1/2, 1, 3/2, 2$

Matlab Program to Solve IVP Heat Equation

```
L=10;
N=10;
h=1/N;
dt=.01;
T=5;
t=0:dt:T;
lt=length(t);
x=(-N*L:N*L)*h;

 f=exp(-x.^2);

M=-2*eye(2*N*L+1)+diag(ones(2*N*L,1),1)+diag(ones(2*N*L,1),-1);

Am=eye(2*N*L+1)-dt/(2*h^2)*M;
Ap=eye(2*N*L+1)+dt/(2*h^2)*M;
Lam = Am\Ap;
 U(:,1)=f;

for j=2:lt
 U(:,j) = Lam*U(:,j-1);
end
ind=[50 100 150 201];

for j=1:4
    ff= 1/(1+4*t(ind(j)))^(1/2)*exp(-x.^2/(1+4*t(ind(j))));
    error(j)=max(abs(U(:,ind(j))-ff'));
end

error

xplot= x([6*N:(14*N+1)]);
tplot=t([1:5:105]);

 figure
set(0,'DefaultAxesFontSize', 18)
surf(xplot,tplot,U([6*N:(14*N+1)],[1:5:105])')
axis([-4,4,0,1,0,1])
xlabel('x')
ylabel('t')
zlabel('u','rotation',0)
print -deps2c sol_heat_box_1.eps

figure
set(0,'DefaultAxesFontSize', 18)
plot(x,U(:,[1, 50, 100, 150, 201]),'LineWidth',2)
xlabel('x')
ylabel('u','rotation',0)
print -deps2c curves_heat_box_1.eps
```