MATH5311 - Advanced Numerical Methods I

Taught by Wang Xiao Ping

Notes by Aaron Wang

Contents

1 September 8th, 2020

1.1 Introduction

1.1.1 Numerical Differentiation

Recall that the derivative is defined as:

$$u(x_0) = \lim_{h\to 0} \frac{u(x_0+h) - u(x_0)}{h}.$$

Thus we can approximate the difference

Definition 1.1. Central Difference

$$u_x(x_i) = \frac{u_{i+1} - u_{i-1}}{2h}.$$

Theorem 1.2
The central difference method is second order accurate.

Proof. Using Taylor Expansion, we have:

$$u_{i+1} = u_i + u_x h + \frac{1}{2} u_{xx}.$$

□

Remark 1.3 — Note that if the derivative is

1.1.2 Numerical Integration

Numerical Integration can be approximation by the **Riemann Sum**

$$\int_a^b f(x)\, dx \approx \sum_{i=1}^{N} f(x_i)\Delta x_i.$$

We can also use the **Trapezoidal Rule**, as:

$$\int_{x_i}^{x_{i+1}} \approx \frac{h}{2}\left[f(x_i) + f(x_{i+1})\right].$$

Remark 1.4 — The above equation is approximating the curve in interval $[x_i, x_{i+1}]$ with a straight line

Thus if we add up over all intevals x_i, we'd get:

$$\int_a^b f(x)\, dx \approx \sum_{i=0}^{N-1} \frac{h}{2}\left[f(x_i) + f(x_{i+1})\right].$$

This is **second order accurate**.

1.1.3 Simpson's Rule

1.2 Solution of ODE's

Say we have an ODE:

$$\begin{cases} \dot{x} = f(x,y) \\ x(0) = x_0 \end{cases}.$$

One simple approximation is the **Euler method**:

$$\dot{x} \approx = \frac{x^{n+1} - x^n}{\Delta t} = f(x^n, t_n).$$

with time step Δt and:

$$x^{n+1} = x^n + \Delta t f(x^n, t_n).$$

This is a first order accurate scheme.

> **Remark 1.5** — There are some disadvantages to this method, as Δt must be chosen carefully to be stable.

The Euler method can be modified to make higher order methods, such as the **Modified Euler**

$$\begin{cases} x^* = x^n + \Delta t f(x^n, t_n) \\ x^{n+1} = x^n + \frac{\Delta t}{2}\left(f(x^n, t_n)\right) + f(x^*, t_{n+1}) \end{cases}.$$

Essentially, x^* is a predictor of the This is second order accurate.

> **Remark 1.6** — There is also a **Runge-Katta Method**, which is a 4th order method.

1.2.1 Solving Linear System

We have a linear system:

$$Ax = b.$$

We can solve this with a variety of methods, such as

- Guassian Eliminations
- LU factorization
- Iterative methods

2 September 10th, 2020

2.1 Outline of the Course

1. Parabolic equations, 1D, 2D
2. Hyperbolic equation (1D)
3. Elliptical equation (2D)
4. Iterative methods for linear systems

2.2 Parabolic Equations in 1D

The model problem is the **heat equation** that describes heat conduction.

Definition 2.1. The **heat equation** in 1D is:

$$u_t = ku_{xx}, \quad t > 0, 0 < x < t$$

with boundary conditions

$$u(0,t) = a, \quad u(1,t) = b$$
$$u(x,0) = u_0(x).$$

For simplicity, let's first consider $k = 1$, i.e. $u_t = u_{xx}$, and boundary conditions $u(0,t) = u(1,t) = 0$ the method will be the same.

This can be interrelated as considering a rod with unit length with some initial temperature distribution, and then observing the temperature over time.

Definition 2.2. The above heat equation has **Dirichlet boundary conditions**, meaning that they are fixed at the boundaries, i.e. specifying the temperature.

Definition 2.3. **Neumann boundary conditions** would be specifying the derivative at the boundaries, i.e.

$$u_x(0) = a, \quad u_x(1) = b.$$

Here a and b would be the heat flux at the boundaries.

2.2.1 Explicit Scheme

Like before let's discretized it by considering the uniform grid in space in time, with:

$$x_j = j\Delta x, \quad t_n = n\Delta t,$$

for $j = 0, 1, \ldots, J, \quad n = 0, 1, , \ldots$ and $\Delta x = \frac{1}{J}$. Thus we seek to approximate:

$$U_j^n \approx u(x_j, t_n).$$

The first approximation would be using an explicit method, with:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} \approx u_t(x_j, t_n).$$

meaning we use the forward difference for the time derivative. For space derivative, we have:

$$\frac{U_{j+1}^n + U_{j-1}^n - 2U_j^n}{(\Delta x)^2} \approx u_{xx}(x_j, t_n).$$

which is derived from the central difference.

Remark 2.4 — Remembering Taylor expansion, this is because:

$$u_{i+1} + u_{i-1} = 2u_i + u_{xx}h^2 + \frac{2}{4!}u^{(4)}h^4 + \ldots.$$

$$\implies \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = u_{xx} + O(h^2).$$

Using this, we can discretize the PDE with:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2}.$$

Now that we have a discrete finite difference formula, we can solve it for U_j^n:

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{(\Delta x)^2}\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right).$$

Using the notation $\nu = \frac{\Delta t}{(\Delta x)^2}$, we have:

$$U_j^{n+1} = (1-2\nu)U_j^n + \nu\left(U_{j+1}^n + U_{j-1}^n\right) \tag{1}$$

This is an **explicit scheme**, as each value at time level t_{n+1} can be independently calculated from the values at time t_n. As such, we can start with $n = 0$ and calculate for each next value of n step by step.

2.2.2 Error Analysis for Explicit Scheme

Since we are using U_j^n to approximate $u(x_j, t_n)$, if we replace U_j^n by the exact solution, the truncation error would be:

$$T_j^n = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} - \frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)}{(\Delta x)^2} \tag{2}$$

Since this is smooth, we can use the taylor expansion, giving us:

$$\begin{aligned}
&\approx \frac{u_t(x_j, t_n)\cancel{\Delta t} + \frac{1}{2}U_{tt}(x_j, t_n)(\Delta t)^{\cancel{2}}}{\cancel{\Delta t}} - \frac{u_{xx}\cancel{(\Delta x)^2} + \frac{2}{4!}u_{xxxx}(\Delta x)^{\cancel{4}2}}{\cancel{(\Delta x)^2}} \\
&= u_t + \frac{1}{2}u_{tt}(\Delta t) - u_{xx} - \frac{2}{4!}u_{xxxx}(\Delta x)^2 + \dots \\
&= \underbrace{\cancel{u_t - u_{xx}}}_{=0} + \frac{1}{2}u_{tt}\Delta t - \frac{2}{4!}u_{xxxx}(\Delta x)^2 + \dots \\
&= \frac{1}{2}u_{tt}\Delta t - \frac{2}{4!}u_{xxxx}(\Delta x)^2 + \dots.
\end{aligned}$$

Note that:

$$T^n \to 0, \quad \text{as } \Delta t \to 0, \quad (\Delta x) \to 0.$$

Since the truncation error goes to zero, this scheme is consistent. Note that this scheme is first order accurate in time and second order accurate in space. Although this scheme is consistent, this is only a necessary condition for convergence.

Definition 2.5. The scheme is **convergent** if as $\Delta t \to 0, \quad \Delta x \to 0$, $\forall (x^*, t^*)$: $x_j \to x^*$ and $t_n \to t^*$ implies $U_j^t \to u(x^*, t^*)$

Let $e_j^n = U_j^n - u(x_j, t_n)$, meaning that convergence is equivelent to $e_j^n \to 0$ as $\Delta t \to 0$ and $\Delta x \to 0$. Rearranging 1 and 2. We have:

$$e_j^{n+1} = e_j^n + \nu\left(e_{j+1}^n - 2e_j^n + e_{j-1}^n\right) + T_j^n \Delta t$$
$$\implies e_j^{n+1} = (1-2\nu)e_j^n + \nu e_{j+1}^n + \nu e_{j-1}^n + T_j^n \Delta t$$
$$\implies |e_j^{n+1}| \leq |1-2\nu||e_j^n| + |\nu||e_{j+1}^n| + |\nu||e_{j-1}^n| + |T_j^n|\Delta t$$
$$\implies |e_j^{n+1}| \leq |1-2\nu|E^n + |\nu|E^n + |\nu|E^n + |T_j^n|\Delta t$$

where $E^n = \max_j\{|e_j^n|\}$. If $\nu \leq \frac{1}{2}$, we have:

$$\max_j\{|e_j^{n+1}\} \leq |1-2\nu|E^n + 2|\nu|E^n + \tilde{T}\Delta t = E^n + \tilde{T}\Delta t.$$

where $\tilde{T} = \max_{j,n}\{|T_j^n|\}$. Now we have:

$$\begin{aligned} E^{n+1} &\leq E^n + \tilde{T}\Delta t \leq (E^{n-1} + \tilde{T}\Delta t) + \tilde{T}\Delta t \\ &\leq E^{n-1} + 2\tilde{T}\Delta t \\ &\leq \dots \\ &\leq \underbrace{\cancel{E^0}}_{=0} + n\tilde{T}\Delta t \\ &= t_F \tilde{T} \end{aligned}$$

where $t_F = n\Delta t$ which is a constant. Since $\tilde{T} \to 0$ as $\Delta x \to 0$ and $\Delta t \to 0$ because it is consistent, we have that E^n goes to zero. Note that this means that we have the condition that:

$$\nu = \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \implies 2\Delta t \leq (\Delta x)^2,$$

meaning that our scheme is convergent provided that $\nu \leq \frac{1}{2}$. Next class we will consider $\nu > \frac{1}{2}$ and show that it will not converge.

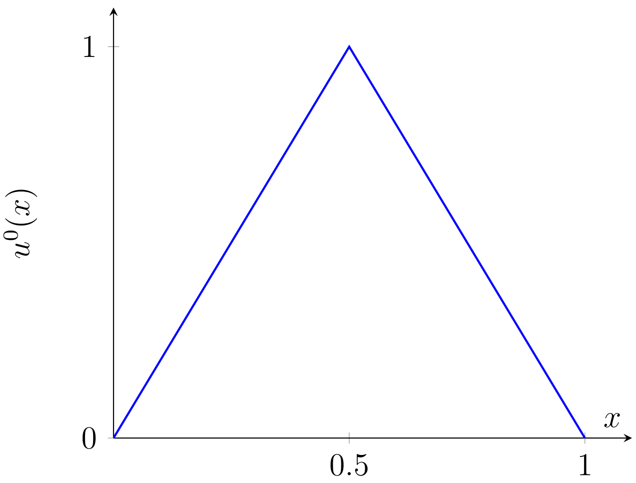
3 September 15th, 2020

3.1 More on the Explicit Method for 1D Heat Equation

Last time we showed an explicit method for evaluating u_t and u_{xx} for the 1D heat equation. It is a explicit scheme because each value at time level t_{n+1} can be independently calculated from values at time level t_n. We also showed that the scheme converges for $\nu \leq \frac{1}{2}$, with $\nu = \frac{\Delta t}{(\Delta x)^2}$.

Let's consider the example with initial conditions:

$$u^0(x) = \begin{cases} 2x, & 0 \leq x \leq \frac{1}{2} \\ 2-2x, & \frac{1}{2} \leq x \leq 1 \end{cases} .$$



Let's assume that the grid size is $\Delta x = 0.05$, i.e. there are 20 grid points in the space dimension. From the condition, we have:

$$\Delta t \leq \frac{1}{2}(\Delta x)^2 = 0.0125.$$

From this, let's consider the cases:

$$\Delta t = 0.0012 \implies \nu < \frac{1}{2} \text{ and } \Delta t = 0.0013 \implies \nu > \frac{1}{2}.$$

Running some MATLAB simulation reveals that U_j^n converges to 0 if $t = 0.0012$, but it does not for $t = 0.0013$. Thus, the condition $\nu \leq \frac{1}{2}$ is a stability condition.

3.2 Fourier Analysis of Error

Any smooth function can be expanded into a Fourier series:

$$f(x,t) = \sum_{n=-\infty}^{+\infty} a_n(t)e^{inx},$$

for some complex function $a_n(t)$ Where:

$$e^{inx} = \cos(nx) + i\sin(nx).$$

Thus we can write our equation as:

$$U_j^n = \lambda^n e^{ik(j\Delta x)}, \quad j = 0, 2, \ldots, N.$$

Remember our numerical explicit scheme is:

$$U_j^{n+1} = U_j^n + \nu(U_{j+1}^n - 2U_j^n + U_{j-1}^n), \quad \nu = \frac{\Delta t}{(\Delta x)^2}.$$

$$\implies \lambda^{n+1}e^{ik(j\Delta x)} = \lambda^n e^{ik(j\Delta x)} + \nu\lambda^n\left(e^{ik(j+1)\Delta x} - 2e^{ikj\Delta x} + e^{ik(j-1)\Delta x}\right).$$

$$\implies \lambda = 1 + \nu\left(e^{ik\Delta x} - 2 + e^{-ik\Delta x}\right)$$

Using the fact that $e^{ik\Delta x} + e^{-ik\Delta x} = 2\cos k\Delta x$, we have:

$$\implies \lambda = 1 + \nu\left(2\cos(k\Delta x) - 2\right).$$

Using the double angle formula: $2(\cos k\Delta x - 1) = 2(-2\sin^2\frac{k\Delta x}{2})$, we have:

$$\lambda = 1 - 4\nu\sin^2\frac{k\Delta x}{2}.$$

For the solution to be well behaved, we need $|\lambda| < 1$, since otherwise $|\lambda|^n$ will grow to infinity, i.e.:

$$\left|1 - 4\nu\sin^2\frac{k\Delta x}{2}\right| \leq 1.$$

$$\implies 1 - 4\nu\sin^2\frac{k\Delta x}{2} \geq -1.$$

$$\nu\sin^2\frac{k\Delta x}{2} \leq \frac{1}{2}.$$

which holds if $\nu \leq \frac{1}{2}$.

To reiterate, in order to ensure that the amplitude (λ) does not grow, we require $\nu \leq \frac{1}{2}$.

THere is a condition called

The method is stable if there exists a constant K independent of k ,s.t.:

$$|\lambda^n| \leq K \text{ for } n\Delta t \leq t_F.$$

Definition 3.1 (Von Neumann Stability Condition)**.** The method is stable if:

$$|\lambda k| \leq 1 + c\Delta t,$$

for some constant c and for all k.

Proof. Note that:

$$|\lambda|^n \leq (1 + c\Delta t)^n \leq \left(1 + \frac{ct_F}{n}\right)^n \leq e^{ct_F} = K.$$

Thus it is a necessary and sufficient condition for the convergence of a consistent difference scheme.G □

Essentially the Von Neumann condition is investigating the condition for converging amplitude of the Fourier series of the numerical scheme. As such, to find the stability, just assume the scheme has the form:

$$U_j^n = \lambda^n e^{ik(j\Delta x)}.$$

3.3 Implicit Scheme for $n_t = u_{xx}$

For this we use backward time difference:

$$\frac{U^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2}, \quad j = 1, 2, \ldots, N+1.$$

This is an implicit scheme since it involves 3 unknown values of U on the new level $n+1$. This gives us $N-1$ equations and $N-1$ unknowns:

$$(1+2\nu)U_j^{n+1} - \nu U_{j+1}^{n+1} - \nu U_{j-1}^{n+1} = U_j^n, \quad j = 1, 2, \ldots, N-1.$$

This givues a tridagonal matrix:

$$AU = b.$$

with :

$$U = \begin{bmatrix} U_1^{n+1} \\ \vdots \\ U_{N-1}^{n+1} \end{bmatrix}, \quad b = \begin{bmatrix} U_1^n \\ \vdots \\ U_{N-1}^n \end{bmatrix}, \quad A = \begin{bmatrix} 1+2\nu & -\nu & 0 & \\ -\nu & 1+2\nu & \ddots & \\ 0 & \ddots & \ddots & -\nu \\ & & -\nu & 1+2\nu \end{bmatrix}.$$

4 September 17th, 2020

4.1 Implicit Scheme for $u_t = u_{xx}$

Recall that the implicit scheme is:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2}.$$

Note that when compared to the explicit scheme, the implicit scheme involves 3 unknown values of U on the new level $n+1$. This is in contrast to the explicit scheme, for which the values of U_j^{n+1} only depend on U^n. Thus there are $N-1$ unknowns: $U_1^{n+1}, U_2^{n+1}, \ldots, U_{N-1}^{n+1}$, and $N-1$ equations:

$$(1+2\gamma)U_j^{n+1} - \gamma U_{j-1}^{n+1} - \gamma U_{j-1}^{n+1} = U_j^n.$$

This can be expressed as a linear system $AU = b$, with A being tridiagonal.

The simplest way to solve this linear system is Gaussian elimination, which for a tridiagonal matrix is similar to Thomas algorithm which solves the equation:

$$-a_j U_{j-1} + b_j U_j - c_j U_{j+1} = d_j, \quad j = 1, \ldots, N-1.$$

While assuming diagonally dominance:

$$a_j > 0, b_j > 0, c_j > 0, \quad b_j > a_j + c_j.$$

Remark 4.1 — This diagonal dominance is to ensure there is a solution (not singular).

4.2 Stability Analysis for Implicit Scheme

Recall we are considering the equation:

$$\begin{cases} u_t = u_{xx} \\ u(0,t) = u(1,t) = 0 \\ u(x,0) = u_0(x) \end{cases}.$$

Assuming we can do separation of variables, we have:

$$u(x,t) = Z(x) \cdot T(t).$$

Taking the Fourier series of the original equation, we have

$$u(x,t) = \sum_{k=1}^{\infty} a_k(t) \sin k\pi x$$

$$\implies \sum_{k=1}^{\infty} a_k(t) \sin k\pi x = -\sum_{k=1}^{\infty} a_k(t)(k\pi)^2 \sin k\pi x.$$

Since $\sin k\pi x$ forms a basis, the coefficients must match, giving us:

$$a_k'(t) = -(k\pi)^2 a_k(t)$$

$$\implies a_k(t) = a_0 e^{-(k\pi)^2 t}.$$

Note that the evolution of a_k is independent of other values of k. Thus in order to study how amplitude evolves with time, we don't need to look at the whole series, only how the amplitude decays with k. Thus for an exact solution of $u_t = u_{xx}$, we know that the amplitude decays exponentially fast.

For the discretized case, we want to see how the numeric scheme propagates the Fourier mode. Thus we let:

$$U_j^n = \lambda^n e^{ik(j\Delta x)}.$$

Plugging into the numerical implicit scheme, we have:

$$(1+2\nu)\lambda^{n+1} e^{ik(j\Delta x)} - \nu\lambda^{n+1} e^{ik(k+1)\Delta x} - \nu\lambda^{n+1} e^{ik(j-1)\Delta x} = \lambda^n e^{ikj\Delta x}.$$

$$\implies \lambda\left[(1+2\nu) - \nu e^{ik\Delta x} - \nu e^{-ik\Delta x}\right] = 1.$$

$$\implies \lambda\left(1 + 2\nu - 2\nu\cos k\Delta x\right) = 1.$$

$$\implies \lambda\left(1 + 4\nu\sin^2\frac{k\Delta x}{2}\right) = 1.$$

$$\implies \lambda = \frac{1}{1 + 4\nu\sin^2\frac{k\Delta x}{2}} < 1.$$

Thus this implicit scheme unconditionally stable, meaning there is no condition on ν. Remember that for the explicit scheme, we needed the condition $\nu \le \frac{1}{2}$.

4.3 The θ-Method

Recall we have learned two schemes:

- Explicit Scheme:
$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2}.$$
- Implicit Scheme:
$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2}.$$

Both schemes have first order error in time t and second order in space. This can be seen the truncation error T_j^n using taylor expansion.

Definition 4.2. The ***θ*-method** is a weighted average of explicit and implicit scheme. For the heat equation this is:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = (1-\theta)\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2} + \theta\frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2}, \quad 0 \le \theta \le 1.$$

Remark 4.3 — If $\theta = 0$, we have a explicit scheme, and if $\theta = 1$ we have the implicit scheme, both with 1st order in time and 2nd order in space.

However, if use $\theta = \frac{1}{2}$, we have 2nd order in time and space. This is because there is some cancellation when $\theta = \frac{1}{2}$. For any other values of θ, this will not be true. To calculate the truncation error for the θ method, we expand terms at $(x_j, t_{n+\frac{1}{2}})$:

$$u(x_j, t_n) = u(x_j, t_{n+\frac{1}{2}}) - u_t(\frac{1}{2}\Delta t) + \frac{1}{2}u_{tt}\left(-\frac{1}{2}\Delta t\right)^2$$
$$u(x_j, t_n) = u(x_j, t_{n+\frac{1}{2}}) - u_t(\frac{1}{2}\Delta t) + \frac{1}{2}u_{tt}\left(-\frac{1}{2}\Delta t\right)^2$$

This gives truncation error:

$$T_j^{n+\frac{1}{2}} = \underbrace{(u_t - u_{xx})}_{=0} + \left[(\frac{1}{2} - \theta)\Delta t u_{xxt} - \frac{1}{12}(\Delta x)^2 u_{xxxx}\right] + \frac{1}{4!}\left(\frac{1}{2} - \theta\right)\Delta t u_{xxxxt}(\Delta x)^2$$
$$+ O(\Delta t)^2 + O((\Delta x)^2)$$

Note that when $\theta = \frac{1}{2}$, the truncation error is second order in both time and space. This is called the Czzrank-Nicolson scheme. Now the natural question is what is the stability of the this θ-method. We have:

- $0 \le \theta \le \frac{1}{2}$: stable $\iff$ $\nu < \frac{1}{2}(1-2\theta)^{-1}$
- $\frac{1}{2} \le \theta \le 1$: stable for all ν

Thus the Crank-Nicolson scheme is unconditionally stable.

5 September 22nd, 2020

5.1 Three Time Level Scheme

Recalling the θ-method, when $\theta = \frac{1}{2}$, note that both the left and right hand sides are symmetric with respect to $n + \frac{1}{2}$, making it similar to the central difference. This is the intuition for why it has second order with respect to both time and space.

For the previous schemes, we use forward difference to expand u_t. Another way to express this is using the central difference instead:

$$\frac{U_j^{n+1} - U^{n-1})j}{2\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2}.$$

Note that both sides are symmetric with respect to n. This scheme is second order in both time and space, and is an explicit scheme. Note that it is a three-time level scheme, meaning we need the values at $n-1$ and n, and after that we can get the values for $n+1$.

Investigating the stability using Von Neumann stability analysis, we have:

$$U_j^n = \lambda^n e^{ik(j\Delta x)} \implies \frac{\lambda - \lambda^{-1}}{2\Delta t} = \frac{-4\sin^2(\frac{1}{2}k\Delta x)}{(\Delta x)^2}.$$

$$\implies \lambda^2 + 8\lambda\mu \sin^2(\frac{1}{2}k\Delta x) - 1 = 0.$$

This has two roots: λ_1, λ_2 with $\lambda_1 \cdot \lambda_2 = -1$. This means that $|\lambda_1| > 1$ or $|\lambda_2| > 1$, meaning that the scheme is always unstable. As such, this three level scheme is explicit and second order in both time and space, it is unstable.

Remark 5.1 — Note that since $n+1$ requires n and $n-1$, we'd have to use the forward difference explicit scheme for the first iteration, and then we can use the three time level scheme.

As a summary, we've investigated the following schemes:

- **Explicit Scheme**
  - Advantage: simple and easy to implement
  - Disadvantage: has a stability constraint of $\mu = \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$.
- **Implicit Scheme**
  - Advantage: unconditionally stable, meaning there is no restriction on Δt
  - Disadvantage: needs to solve a linear system at each time step

Remark 5.2 — In high dimensions, e.g. $d = 2, 3$ then implicit scheme is preferred, as it has a lower computational cost.

5.2 Boundary Conditions

Recall we are looking at the 1D heat equation:

$$u_t = u_{xx}, \quad 0 \le x \le 1.$$

Currently, we are solving with zero boundary conditions, i.e.:

$$u(0,t) = u(1,t) = 0.$$

and with initial conditions:

$$u(x,0) = u_0(x).$$

Recall for the finite difference methods before, we only calculate for the interior points, $j = 1, 2, \ldots, N-1$. This is because for $j = 0$ and $j = N$, the solution is known (since it is the boundary condition).

Note that we do use the boundary data for $j = 1$ and $j = N-1$, for example:

$$\frac{U_1^{n+1} - U_1^n}{\Delta t} = \frac{U_2^n + 2U_1^n + \cancel{U_0^n}}{(\Delta x)^2}.$$

If we instead consider the case where the boundary conditions are non zero but constant, i.e.:

$$u(0,t) = a, \quad u(1,t) = b, \quad a, b > 0.$$

Then, we can still calculate it, e.g.:

$$\frac{U_1^{n+1} - U_1^n}{\Delta t} = \frac{U_2^n + 2U_1^n + a}{(\Delta x)^2}.$$

Definition 5.3. The above boundary conditions are **Dirichlet boundary conditions**, as the function value is specified at the boundaries.

Definition 5.4. We can also have the **Neumann boundary conditions**, where we specify the derivative of the function at the boundaries

Example 5.5

An example of Neumann boundary condition would be

$$u_x(0,t) = 0, \quad u_x(1,t) = 0.$$

Remark 5.6 — The physical representation of Neumann boundary condition would be the heat flux, for example if $u_x(0,t) = u_x(1,t) = 0$, then there is no heat entering or exiting, meaning it is insulating.

Let us consider the insulating case with the explicit scheme:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2}.$$

In particular, consider $j = 1$, where:

$$\frac{U_1^{n+1} - U_1^n}{\Delta t} = \frac{U_2^n + 2U_1^n + U_0^n}{(\Delta x)^2}.$$

When $n = 0$, U_0^n is known (just the initial condition), but U_0^n is not known for $n = 1, 2, \ldots$ To fix this, we can use finite difference to discretize the boundary conditions.

To use central difference at the boundaries, we can introduce a ghost point at $x_{-1} = -\Delta x$ and $x_{N+1} = 1 + \Delta x$, meaning that at $x = 0$:

$$x_0 = 0 \approx \frac{U_1 - U_{-1}}{2\Delta x}.$$

which has second order accuracy. Meanwhile at $x = 1$, we have:

$$u_x = 0 \approx \frac{U_{N+1} - U_{N-1}}{2\Delta x}.$$

Now instead of only evaluating interior points, we also evaluate at the boundary points.

In particular when $j = 0$, we have:

$$\frac{U_0^{n+1} - U_0^n}{\Delta t} = \frac{U_1^n - U_0^n + U_{-1}^n}{(\Delta x)^2}.$$

With left boundary condition:

$$\frac{U_1^{n+1} - U_{-1}^{n+1}}{2\Delta x} = 0.$$

Thus, we first update the solution

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{(\Delta x)^2}\left(U_{j+1}^n - 2U_j^n + U_{j+1}^n\right), \quad j = 0, 1, , \ldots, N.$$

then we update the boundaries:

$$U_{-1}^{n+1} = U_1^{n+1}.$$
$$U_{N+1}^{n+1} = U_{N-1}^{n+1}.$$

We can do the same thing for the implicit scheme too. In summary, for Neumann boundary we need to introduce ghost points in order to discretize the boundary condition to maintain the second order accuracy.

6 September 24th, 2020

6.1 Parabolic Equation in Two and Three Dimensions

Now let's consider the heat equation in 2 dimensions:

$$\begin{cases} u_t = \sigma(u_{xx} + u_{yy}) = \sigma\Delta u & \\ u|_{\partial\Omega} = f_0 & \text{(boundary condition)} \\ u(x, 0) = u_0(x) & \text{(initial temperature distribution)} \end{cases}.$$

Remark 6.1 — Δu is the Laplacian of u, and is $u_{xx} + u_{yy}$

To solve it numerically, we once again discretize it:

$$x_i = i\Delta x, \quad y_j = j\Delta y$$

for $0 \leq i \leq J_x$ and $0 \leq j \leq J_y$, with the step sizes Δx and Δy being the size of the corresponding domain divided by the number of grid points J_x and J_y. We once again have finite difference:

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} \approx u_t(x_i, u_j, t_n).$$

$$\frac{U_{i+1,j}^n + U_{i-1,j} - 2U_{i,j}^n}{(\Delta x)^2} \approx u_{xx}(x_i, y_j, t_n).$$

$$\frac{U_{i,j+1}^n + U_{i,j-1} - 2U_{i,j}^n}{(\Delta y)^2} \approx u_{yy}(x_i, y_j, t_n).$$

Plugging into the heat equation, we have explicit scheme:

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} = \sigma\left(\frac{U_{i+1,j}^n + U_{i-1,j} - 2U_{i,j}^n}{(\Delta x)^2} + \frac{U_{i,j+1}^n + U_{i,j-1} - 2U_{i,j}^n}{(\Delta y)^2}\right).$$

$$\implies U_{i,j}^{n+1} = \sigma\frac{\Delta t}{(\Delta x)^2}\left(U_{i+1,j}^n + U_{i-1,j}^n - 2U_{i,j}^n\right) + \sigma\frac{\Delta t}{(\Delta y)^2}\left(U_{i,j+1}^n + U_{i,j-1}^n - 2U_{i,j}^n\right) + U_{i,j}^n.$$

Since this is explicit, we expect some condition on

$$\nu_x = \frac{\Delta t}{(\Delta x)^2}, \text{ and } \nu_y = \frac{\Delta t}{(\Delta y)^2}.$$

Note that this is also a consistent scheme.

Now calculating the Truncation error by Taylor expansion, we have:

$$T(x,t) = \frac{1}{2}\Delta t u_{xx} - \frac{1}{12}\sigma\left[(\Delta x)^2 u_{xxxx} + (\Delta y)^2 u_{yyyy}\right] + \ldots.$$

Note that $T_{ij}^n \to 0$ as $\Delta t \to 0$ and $(\Delta x, \Delta y) \to (0,0)$. In addition, it is first order in time, and second order in space.

Theorem 6.2

The explicit scheme converges under the condition $\nu_x + \nu_y \leq \frac{1}{2}$.

Proof. Homework. □

Remark 6.3 — Note that if we let $\Delta x = \Delta y$, the above condition would be:

$$\frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2} \leq \frac{1}{2} \implies 2\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \implies \Delta t \leq \frac{1}{4}(\Delta x)^2.$$

This means that in 2D, the condition for convergence is even more restricted than the 1D case. Because of this, in higher dimensions, we don't want to use explicit schemes.

Let's now consider this scheme using Von Neumann stability analysis, for $\sigma = 1$, we have:

$$\implies U_{i,j}^{n+1} = U_{ij}^n + \frac{\Delta t}{(\Delta x)^2}\left(U_{i+1,j}^n + U_{i-1,j}^n - 2U_{i,j}^n\right) + \frac{\Delta t}{(\Delta y)^2}\left(U_{i,j+1}^n + U_{i,j-1}^n - 2U_{i,j}^n\right).$$

We have:

$$U_{ij}^n \sim \lambda^n e^{ik_x x_i + jk_j y_j}.$$

Substituting, we have:

$$\begin{aligned}\lambda^{n+1}e^{ik_x x_i + jk_y y_j} = \lambda^n e^{i(k_x x_i + k_y y_j)} &+ \nu_x \lambda^n \left(e^{ik_x x_{i+1}} - e^{ik_x x_i} + e^{ik_x x_{i-1}}\right) e^{ik_y y_j} \\ &+ \nu_y \lambda^n \left(e^{ik_y y_{j+1}} - 2e^{ik_y y_j} + e^{ik_y y_{j-1}}\right) e^{ik_x x_i}.\end{aligned}$$

$$= \lambda^n e^{ik_x x_i + jk_y y_j} + \nu_x \lambda^n \left(e^{ik_x \Delta x} - 2 + e^{-ik_x \Delta x}\right) e^{ik_x x_i + ik_y y_j} + \nu_y \lambda^n \left(e^{ik_y \Delta y} - 2 + e^{-ik_y \Delta y}\right) e^{ik_x x_i + jk_y y_j}.$$

$$\implies \lambda = 1 + \nu_x\left(-4\sin^2\frac{k_x \Delta x}{2}\right) + \nu_y\left(-4\sin^2\frac{k_y \Delta y}{2}\right).$$

Let us consider the case where $\Delta x = \Delta y$, meaning that $\nu_x = \nu_y$ and $k_x = k_y$, thus we have:

$$\lambda = 1 + 2\nu_x\left(-4\sin^2\frac{k_x \Delta x}{2}\right) = 1 - 8\nu_x \sin^2\frac{k_x\Delta x}{2} \leq 1.$$

Since we want $|\lambda| \leq 1$, we need:

$$1 - 8\nu_x \sin^2\frac{k_x \Delta x}{2} \geq -1 \implies \nu_x \sin^2\frac{k_x\Delta x}{2} \leq \frac{1}{4}.$$

If $\Delta x \neq \Delta y$, we can still do it, but just a bit more complicated.

6.2 Implicit Scheme

The implicit scheme of the 2D heat equation is of the form:

$$\implies U_{i,j}^{n+1} = U_{ij}^n + \frac{\Delta t}{(\Delta x)^2}\left(U_{i+1,j}^{n+1} + U_{i-1,j}^{n+1} - 2U_{i,j}^{n+1}\right) + \frac{\Delta t}{(\Delta y)^2}\left(U_{i,j+1}^{n+1} + U_{i,j-1}^{n+1} - 2U_{i,j}^{n+1}\right).$$

To solve this we need to solve a linear system. Since we discretize $i = 1, 2, \ldots, J_x - 1$ and $j = 1, 2, \ldots, J_y - 1$, we have a total of $(J_x - 1) \times (J_y - 1) = M$ equations, with the same number of unknowns. If we write this linear system in a matrix form, we have:

$$Ax = b$$

Where

$$A = \begin{bmatrix} D & L & & \\ L & D & \ddots & \\ & \ddots & \ddots & L \\ & & L & D \end{bmatrix}_{M\times M}$$

With

$$A = \begin{bmatrix} 1 + 2\nu_x + 2\nu_y & -\nu_x & & \\ -\nu_x & 1 + 2\nu_x + 2\nu_y & \ddots & \\ & \ddots & \ddots & -\nu_x \\ & & -\nu_x & 1 + 2\nu_x + 2\nu_y \end{bmatrix}$$

and

$$L = \begin{bmatrix} -\nu_y & & & \\ & \ddots & & \\ & & \ddots & \\ & & & -\nu_y \end{bmatrix}$$

Remark 6.4 — A is a symmetric-positive definite matrix.

Remark 6.5 — The form of matrix A, D, and L will depend on how you order x.

Rearrange the implicit scheme, we have:

$$-\nu_y U_{i,j-1}^{n+1} - \nu_x U_{i-1,j}^{n+1} + (1 - 2\nu_x - 2\nu_y) U_{i,j}^{n+1} - \nu_x U_{i+1,j}^{n+1} - \nu_y U_{i,j+1}^{n+1} = U_{i,j}^{n}.$$

To order x, we will first order them by y and then x, i.e.:

$$x = \begin{bmatrix} U_{11} \\ U_{21} \\ U_{31} \\ \vdots \\ U_{12} \\ U_{22} \\ \vdots \end{bmatrix}.$$

Note that x is a vector of lengths $(J_x - 1) \times (J_y - 1)$, as we only consider the interior points. Similar to the 1D case, the benefit of this implicit scheme is stability.

Remark 6.6 — If we use gaussian elimination to solve this linear system, the computation cost would be $O((M \times M)^2)$, which is unfeasible, especially for large M.