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:

$$\approx \frac{u_t(x_j,t_n)\cancel{\Delta t} + \frac{1}{2}U_{tt}(x_j,t_n)(\Delta t)^{\cancel{2}}}{\cancel{\Delta t}} - \frac{u_{xx}\cancel{(\Delta x)^2} + \frac{2}{4!}u_{xxxx}(\Delta x)^{\cancel{4}2}}{\cancel{(\Delta x)^2}}$$

$$= u_t + \frac{1}{2}u_{tt}(\Delta t) - u_{xx} - \frac{2}{4!}u_{xxxx}(\Delta x)^2 + \dots$$

$$= \underbrace{\cancel{u_t - u_{xx}}}_{=0} + \frac{1}{2}u_{tt}\Delta t - \frac{2}{4!}u_{xxxx}(\Delta x)^2 + \dots$$

$$= \frac{1}{2}u_{tt}\Delta t - \frac{2}{4!}u_{xxxx}(\Delta x)^2 + \dots.$$

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:

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

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

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

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

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

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:

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

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

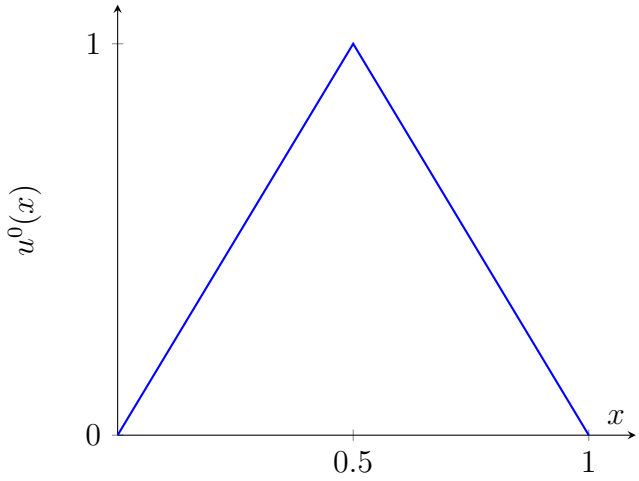
3 September 15th, 2020

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 $\nu \le \frac{1}{2}$, with $\nu = \frac{\Delta t}{(\Delta x)^2}$.

Let's consider the example with initial conditions:

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



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:

$$U_j^n = \lambda^n e^{ik(j\Delta x)}, \quad j = 0, 1, 2, \ldots, N.$$

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 $\nu \leq \frac{1}{2}$.

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 $u_t = u_{xx}$

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 $\nu \leq \frac{1}{2}$.

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:

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

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:

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

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:

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

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 $0 \le i \le J_x$ and $0 \le j \le J_y$, 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:

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

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 $\nu_x + \nu_y \le \frac{1}{2}$.

Proof. Homework. □

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

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

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:

$$U_{ij}^n \sim \lambda^n e^{i(k_x x_i + k_y y_j)}.$$

Substituting, we have:

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

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

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

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.

7 September 29th, 2020

7.1 ADI Method

The **ADI Method** or Alternative Directional Implicit Method has two steps

- Step 1: solve for $u_{ij}^{n+\frac{1}{2}}$ with

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

- Step 2: solve for u_{ij}^{n+1} with

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

In both steps, we only need to solve a tridiagonal system.

Remark 7.1 — Note that Step 1 is explicit in x, where as Step 2 is implicit in y. As such, instead of having to solve a huge system with 5 diagonals (which is not easy to solve), we only need to solve two tridiagonal systems.

If we use δ_x^2 and δ_y^2 to denote the finite difference in x and y, i.e.:

$$\delta_x^2 U = \frac{U_{i+1,j} + U_{i-1,j} - 2U_{i,j}}{(\Delta x)^2}.$$

$$\delta_y^2 U = \frac{U_{i,j+1} + U_{i,j+1} - 2U_{i,j}}{(\Delta y)^2}.$$

and let

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

we can write the ADI method as:

$$\begin{cases} U^{n+1} - U^n = \frac{1}{2}\nu_x \delta_x^2 U^{n+\frac{1}{2}} + \frac{1}{2}\nu_y \delta_y^2 U^n \\ U^{n+1} - U^{n+\frac{1}{2}} = \frac{1}{2}\nu_x \delta_x U^{n+\frac{1}{2}} + \frac{1}{2}\nu_y \delta_y^2 U^{n+\frac{1}{2}} \end{cases}.$$

$$\implies \begin{cases} (1 - \frac{1}{2}\nu_x \delta_x^2) U^{n+\frac{1}{2}} = (1 + \frac{1}{2}\nu_y \delta_y^2) U^n \\ (1 - \frac{1}{2}\nu_y \delta_y^2) U^{n+\frac{1}{2}} = (1 + \frac{1}{2}\nu_x \delta_x^2) U^{n+\frac{1}{2}} \end{cases}.$$

$$U^{n+1} = (1 - \frac{1}{2}\nu_y \delta_y^2)^{-1}(1 + \frac{1}{2}\nu_x \delta_x^2) \underbrace{(1 - \frac{1}{2}\nu_x \delta_x^2)^{-1}(1 + \frac{1}{2}\nu_y \delta_y^2) U^n}_{U^{n+\frac{1}{2}}}.$$

which is in a matrix form.

Remark 7.2 — The truncation error of the ADI method is second order in both time and space.

Proof. Homework. □

To verify the stability, we once again use Von Neumann stability analysis with:

$$U^n \sim \lambda^n \exp(ik_x x_i + jk_y y_j), \quad x_i = i\Delta x, y_j = j\Delta y.$$

Solving, we would get:

$$\lambda = \frac{(1 - 2\nu_x \sin^2(\frac{1}{2}k_x \Delta x))(1 - 2\nu_y \sin^2(\frac{1}{2}k_y \Delta y))}{(1 + 2\nu_x sin^2 \frac{1}{2}k_x \Delta x)(1 + 2\nu_y sin^2(\frac{1}{2}k_y \Delta y))} = \frac{(1-a)(1-b)}{(1+a)(1+b)}.$$

where:

$$a = 2\nu_x \sin^2(\frac{1}{2}k_x \Delta x) \geq 0 \quad b = 2\nu_y \sin^2(\frac{1}{2}k_y \Delta y) \geq 0.$$

We need to show that $|\lambda| \leq 1$, and as such:

$$|\lambda| = \left|\frac{(1-a)(1-b)}{(1+a)(1+b)}\right| = \left|\frac{1-a}{1+a}\right| \cdot \left|\frac{1-b}{1+b}\right|.$$

$$\frac{(1-a)^2}{(1+a)^2} = \frac{1 - 2a + a^2}{1 + 2a + a^2} \leq 1.$$

Thus $|\lambda| \leq 1$ meaning that the ADI method is unconditionally stable.

7.2 Summary of 2D Parabolic (Heat) Equation

- The case for 2D heat equation is quite similar to 1D but with higher computational cost.
- We can still achieve second order accuracy both in time and space by using the ADI method, which combines implicit and explicit methods.
- The analysis is quite similar.

Remark 7.3 — From the analysis of PDE's the solution to parabolic PDEs are smooth. By smooth, this means that it is infinitely differentiable, which is needed for finite difference to work well since Taylor expansion is the basis of finite difference. If the solution is not smooth, then the solution can not be approximated by the Taylor expansion, thus the finite difference methods do not work. We will see that this is the case for hyperbolic equations.

7.3 Hyperbolic Equations in 1D

For the heat equation, if the boundary conditions are fixed at 0, then the whole function will eventually go to zero and reach a steady state. However this is different for hyperbolic equations. Let us consider the simplest hyperbolic equation:

$$\begin{cases} \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \\ u(x,0) = f(x) \end{cases}.$$

This has a solution

$$u(x,t) = f(x-at).$$

Proof. We have:

$$u_t = f' \cdot (-a) \text{ and } u_x = f' \cdot (1).$$

Adding this up, we have:

$$u_t + au_x = -af' + af' = 0.$$

which is the equation. □

Remark 7.4 — This is called the travelling wave solution, since it is the initial condition $f(x)$ shifted by at. The speed is a.

Remark 7.5 — If $a > 0$ the wave will shift to the right, but if $a < 0$ it will shift to the left.

Unlike the heat equation which describes the diffusion process, the hyperbolic (transport) equation describes wave propagation.

7.4 Method of Characteristics

To solve the hyperbolic equation, we can use the method of characteristic. The characteristic line of the equation above is:

$$\frac{dx}{dt} = a \implies x = at + x_0.$$

Moving along the characteristic line, we have:

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} \cdot \frac{dx}{dt} = \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0.$$

This means that u is constant along the characteristic line:

$$u(x,t) = u_0(x_0) = u_0(x - at) = f(x - at) = f(x_0).$$

Generalizing this to multidimensional systems, we have:

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(f(u,v)) = 0 \\ \frac{\partial v}{\partial t} + \frac{\partial}{\partial x}(g(u,v)) = 0 \end{cases}.$$

Which in vector form is:

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} f \\ g \end{bmatrix}_x.$$

We can also write this as:

$$\begin{cases} \frac{\partial}{\partial x}(f(u,v)) = \frac{\partial f}{\partial u}u_x + \frac{\partial f}{\partial v}v_x \\ \frac{\partial}{\partial x}(g(u,v)) = \frac{\partial g}{\partial u}u_x \frac{\partial g}{\partial v}v_x \end{cases} \implies \begin{bmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \\ \frac{\partial g}{\partial u} & \frac{\partial g}{\partial v} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}.$$

As such, we have:

$$\begin{bmatrix} u \\ v \end{bmatrix}_t + A \begin{bmatrix} u \\ v \end{bmatrix}_x \quad A = \begin{bmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \\ \frac{\partial g}{\partial u} & \frac{\partial g}{\partial v} \end{bmatrix}.$$

This is hyperbolic if A has real eigenvalues and a full set of eigenfunctions, which means that A is diagonalizable:

$$A = S^{-1}\Lambda S.$$

Where Λ is a diagonal matrix and S is invertible.

Remark 7.6 — If A is a symmetric matrix, it will satisfy this property.

Plugging this into the original equation, we have:

$$Su_t + \Lambda S u_x = 0.$$

With this, we can define a **Riemann invariant**:

$$r = r(u) = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}.$$

such that :

$$\begin{cases} r_t = Su_t \\ r_x = Su_x \end{cases} \implies r_t + \Lambda r_x = 0.$$

Which would allow us to decouple the equation into two transport equations:

$$\begin{cases} r_{1t} + \lambda_1 r_{1x} = 0 \\ r_{2t} + \lambda_2 r_{2x} = 0 \end{cases}.$$

Which leads to two characteristic lines:

$$\frac{dx}{dt} = \lambda_1 \text{ and } \frac{dx}{dt} = \lambda_2.$$

8 October 6th, 2020

8.1 Explicit Scheme for Hyperbolic Equation

Applying the explicit scheme method, we have:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_j^n}{\Delta x} = 0.$$

If we set $\nu = \frac{a\Delta t}{\Delta x}$, we have:

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

Definition 8.1 (CFL Condition)**.** For a convergent scheme, the domain of dependence of the PDE must lie within the domain of dependence of the numerical scheme.

Example 8.2

The scheme above cannot converge for $a > 0$. This is because if $a > 0$, U_j^{n+1} depends on U_j^n, U_{j+1}^n, U_{j+1}^{n-1}, U_{j+2}^{n-1}, etc. However, the characteristic line $x = at + x_0$ does not lie in this domain, thus the scheme does not hold.

Example 8.3

If we instead use a backward difference:

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

Then the characteristic line falls within the domain.

Example 8.4

For $a < 0$, we must have $a\frac{\Delta t}{\Delta x} \leq 1$, in order to have the characteristic line to lie within the domain.

If we attempt to use central difference, we can eliminate the sign condition by using a symmetric scheme in space:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} = 0.$$

Note that for this scheme, CFL condition holds for $|a\frac{\Delta t}{\Delta x}| \leq 1$. However, if we look at this from stability analysis, we have:

$$\frac{\lambda - 1}{\Delta t}\lambda^n e^{ik(j\Delta x)} + \frac{a}{2\Delta x}\lambda^n e^{ik(j\Delta x)}\left(e^{ik\Delta x} - e^{-ik\Delta x}\right) = 0.$$

$$\implies \lambda = 1 - a\frac{\Delta t}{\Delta x} i \sin k\Delta x.$$

which is complex. Thus we have:

$$|\lambda| = \sqrt{1 + a^2\left(\frac{\Delta t}{\Delta x}\right)^2 \sin^2 k\Delta x} > 1.$$

meaning it does not converge. Thus we cannot central difference for an explicit scheme for hyperbolic equation. Instead, we have to use forward or backwards depending on the value of a:

$$U_j^{n+1} = \begin{cases} U_j^n - \nu(U_{j+1}^n - U_j^n) & a < 0 \\ U_j^n - \nu(U_j^n - U_{j-1}^n) & a > 0 \end{cases}.$$

which is called the **upwind scheme**.

Example 8.5

In laymen terms, if the wave is going forward ($a > 0$) we use backward difference, if the wave is going backward we use forward difference.

Analyzing the stability, for $a > 0$, (backward difference) we have:

$$\begin{aligned} \lambda &= 1 - \nu(1 - e^{-ik\Delta x}) \\ &= 1 - \nu(1 - (\cos k\Delta x - i \sin k\Delta x)) \\ &= 1 - \nu(1 - \cos k\Delta x) + i\nu \sin k\Delta x \\ \implies |\lambda|^2 &= (1 - \nu(1 - \cos k\Delta x))^2 + \nu^2 \sin^2 k\Delta x \\ &= 1 - 2\nu(1 - \nu)(1 - \cos k\Delta x) \\ &= 1 - 4\nu(1 - \nu)\sin^2\left(\frac{1}{2}k\Delta x\right) < 1 \quad \text{if } \nu < 1. \end{aligned}$$

This matches the CFL condition.

Remark 8.6 — Since this only uses first order derivatives, this scheme is only first order in time and space.

Previously to get second order accuracy, we used central difference. However, as we showed above, this is not a stable scheme. To achieve this accuracy, we will use quadratic interpolation.

8.2 Lax-Wendroff Scheme

Suppose we want to find the value at P in Figure 1. Using the characteristic line, we have that $U(P) = U(Q)$, and as such, we want to find the value at Q. To do this, we

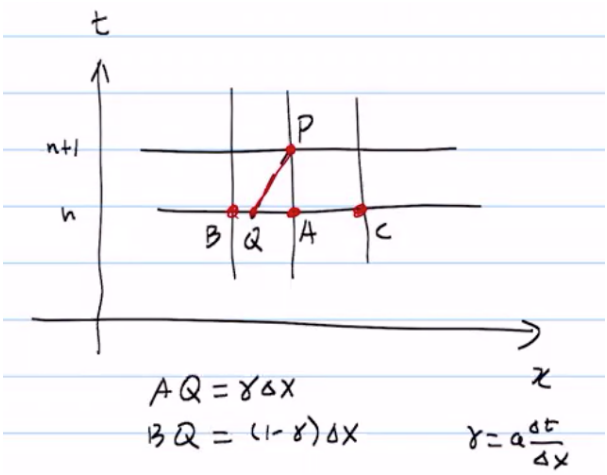


Figure 1

can try linear interpolation with points $B = U^n_{j-1}$, $A = U^n_j$, and $C = U^n_{j+1}$ that lie on the lattice:

$$\begin{aligned} U(Q) &= \nu U(B) + (1-\nu)U(A) \\ \implies U^{n+1}_j &= \nu U^n_{j-1} + (1-\nu)U^n_j. \end{aligned}$$

which is exactly the upwind scheme. If we try quadratic interpolation, we would have: In other words

$$P(x) = U_{j-1}\frac{(x-\Delta x)}{(0-\Delta x)}\cdot\frac{(x-2\Delta x)}{(0-2\Delta x)} + U_j\frac{(x-0)}{(\Delta x-0)}\cdot\frac{(x-2\Delta x)}{(\Delta x-2\Delta x)} + U_{j+1}\frac{(x-0)}{(2\Delta x-0)}\cdot\frac{(x-\Delta x)}{(2\Delta x-\Delta x)}.$$

Plugging in $Q = (1-\nu)\Delta x$, we have:

$$\begin{aligned} P(Q) &= \frac{\nu}{2}(1+\nu)U_{j-1} + (1-\nu)(1+\nu)U_j - \frac{1}{2}\nu(1-\nu)U_{j+1} \\ \implies U^{n+1}_j &= U^n_j + \frac{\nu}{2}(U^n_{j-1}U^n_{j+1}) + \frac{\nu^2}{2}\left(U^n_{j-1} - 2U^n_j + U^n_{j+1}\right). \end{aligned}$$

Remember we need the characteristic line to lie within the scheme to satisfy the CFL condition, meaning we need $|\nu| \leq 1$. This scheme is called the **Lax-Wendroff method**.

Remark 8.7 — Note that this scheme is similar to the central difference but with a higher order correction term.

Remark 8.8 — The above scheme is second order in space. This can be verified by calculating the truncation error.

Checking the stability analysis, we would have:

$$\begin{aligned}
\lambda &= (1 - 2\nu^2 \sin^2 \frac{k\Delta x}{2})^2 + \nu^2 \sin^2 k\Delta x \\
|\lambda|^2 &= (1 - 2\nu^2 \sin^2 \frac{k\Delta x}{2})^2 + \nu^2 \sin^2 k\Delta x \\
&= 1 - 4\nu^2 \sin^2 \frac{k\Delta x}{2} + 4\nu^4 \sin^4 \frac{k\Delta x}{2} + 4\nu^2 \sin^2 \frac{k\Delta x}{2} \cos^2 \frac{k\Delta x}{2} \\
&= 1 - 4\nu^2(1-\nu^2) \sin^4 \left(\frac{1}{2} k\Delta x\right) \leq 1 \quad \text{if } \nu \leq 1
\end{aligned}$$

.

This means the scheme is stable for $|\nu| \leq 1$ (CFL condition).

To summarize, the design of numerical methods for hyperbolic equations is very different that for parabolic equations, as the issues we have to consider are very different.

9 October 8th, 2020

9.1 Upwind and Lax-Wendroff Analysis

The solutions to the upwind and Lax-Wendroff methods have different properties. For example:

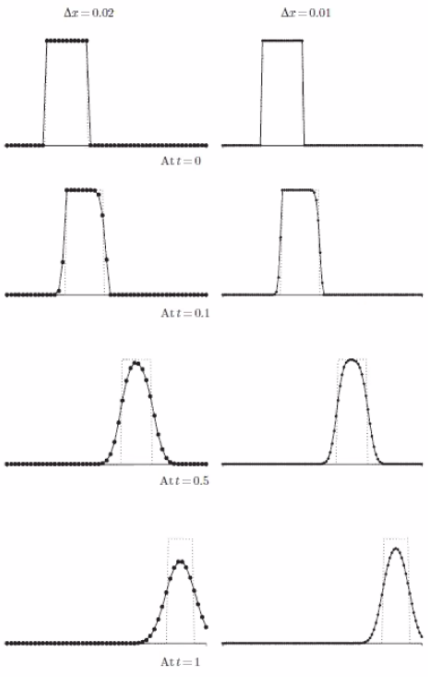


Fig. 4.6. Linear advection by the upwind method: problem (4.33), (4.34).

(a) Upwind Method

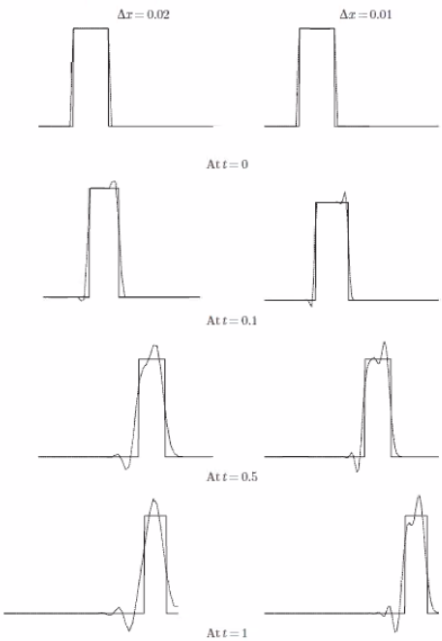


Fig. 4.7. Linear advection by the Lax–Wendroff method: problem (4.33), (4.34).

(b) Lax-Wendroff Method

Figure 2: Modified Equation Analysis

Remark 9.1 — The upwind does not have osciliation, but has smoothing and decaying effect. For the Lax-Wendroff, the result has an oscillation.

Let us look at upwind scheme for $a < 0$, we have:

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

Using Taylor expansion, we have:

$$u(x, t + \Delta t) = u(x, t) + u_t(x, t)\Delta t + \frac{1}{2}u_{tt}(x, t)(\Delta t)^2 + \frac{1}{6}u_{ttt}(\Delta t)^3$$
$$u(x + \Delta x, t) = u(x, t) + u_x(x, t)\Delta x + \frac{1}{2}u_{xx}(x, t)(\Delta x)^2 + \frac{1}{6}u_{xxx}(\Delta x)^3$$

.

Thus, we have:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} \approx u_t + \frac{1}{2}u_{tt}(\Delta t) + \dots.$$
$$\frac{U_{j+1}^n - U_j^n}{\Delta x} \approx u_x + \frac{1}{2}u_{xx}(\Delta x) + \dots.$$

Since $u_{tt} = -au_{tx}$, we have:

$$u_t = -au_x \qquad u_{tt} = a^2u_{xx} \qquad u_{ttt} = -a^3u_{xxx}.$$

Thus:

$$\frac{U^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_j^n}{\Delta x} \approx u_t + au_x + \frac{1}{2}\left(\Delta t a^2 + \Delta x a\right)u_{xx} = 0.$$

In other words, we are approximating:

$$u_t + au_x = \epsilon u_{xx}.$$

for some

$$\epsilon = -\frac{a\Delta x}{2}\left(a\frac{\Delta t}{\Delta x} + 1\right) > 0.$$

Remark 9.2 — Note that this is similar to a parabolic equation with $u_t = \epsilon u_{xx}$. In other words, the effect of the leading order error is diffusion or damping, which is why the upwind has a diffusion behavior. Even though ϵ can be very small, it is non-zero, thus the diffusion effect is always there.

To see why, let's use Fourier analysis. Assume the solution of form:

$$u(x, t) = e^{i(kx - \omega t)}.$$

This is a wave traveling with a certain speed. We have:

$$u_t = -i\omega e^{i(kx-\omega t)}$$
$$u_x = ike^{i(kx-\omega t)}$$
$$u_{xx} = (ik)^2 e^{i(kx-\omega t)}.$$

Plugging this into the equation, we have:

$$e^{i(kx-\omega t)}\left[-i\omega + aik\right] = e^{i(kx-\omega t)}[-\epsilon k^2].$$

Thus, we have:

$$\omega = ak - i\epsilon k^2.$$

This is a dispersion relation, since it is a relation between the frequency in time and the wave length. Plugging this into the exact solution, we have:

$$u(x,t) = e^{i(kx-akt+i\epsilon k^2 t)} = e^{ik(x-at)}e^{-\epsilon k^2 t}.$$

Note that the first factor is a traveling wave with speed a. The second term decays to 0 exponentially fast if $\epsilon > 0$, which is why the upwind method also has a decaying effect.

Remark 9.3 — This is called **modified equation analysis**, as we introduce error when we use finite difference. Therefore, instead of solving the original equation, we are solving a modified equation with some error. The form of this modified equation reveals the behavior of the numerical solution.

Looking at Lax-Wendroff, the modified equation is:

$$u_t + au_x = \frac{a}{6}(\Delta x)^2(\nu^2 - 1)u_{xxx} = \epsilon u_{xxx}.$$

This gives the leading order error a dispersion effect.

If we apply the same concept, we have $u_{xxx} = -ik^3 e^{i(kx-\omega t)}$. Plugging into the modified equation, we have:

$$\omega = ak + \epsilon k^3.$$

If we plug this into the exact solution, we would have a wave speed of $a + \epsilon k^2$ for $a > 0, \nu < 1, \epsilon < 0$. Note that this does not decay to 0 as in the case of the upwind scheme.

Remark 9.4 — $\epsilon < 0$ because $a > 0$ to satisfy CFL condition.

This means that the numerical wave speed will always be slower than the actual wave speed. The higher a wave number, the lower the wave speed. Because the speed of the higher frequency wave is slower than the original wave speed a, the oscillation is always behind.

Remark 9.5 — The reason why we don't see the oscillation in the exact solution is because the waves of each frequency travel together at the same speed. However, for the numerical scheme the speeds are different depending on the frequency.

10 October 15th, 2020

10.1 Box Scheme

From last time, we looked at the modified equations for the upwind scheme and the Lax-Wendroff schemes and saw that they showed different properties because of the leading

order error term. For upwind, this was u_{xx}, giving it a dispersive effect, while for Lax-Wendroff this was u_{xxx}, giving us an oscillatory effect.

Let us now consider the **Box Scheme**:

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

Remark 10.1 — This is essentially taking the average of the time and space derivatives.

If we let $\nu=\frac{\Delta t}{\Delta x}$, we have:

$$(1-a\nu)U_j^{n+1}+(1+a\nu)U_{j+1}^{n+1}=(1+a\nu)U_j^n+(1-a\beta)U_{j+1}^n.$$

Assume for $j=0$, U_j^{n+1} is given for the boundary condition. If this is the case, then we have:

$$(1+a\nu)U_{j+1}^{n+1}=-(1-a\nu)U_j^{n+1}+(1+a\nu)U_j^n+(1-a\beta)U_{j+1}^n.$$

allowing us to solve for U_{j+1}^{n+1} recursively. If the boundary condition is given on the right boundary, i.e. U_N^{n+1} is given, then we can solve recursively from the right, as:

$$(1-a\nu)U_j^{n+1}=-(1+a\nu)U_{j+1}^{n+1}+(1+a\nu)U_j^n+(1-a\beta)U_{j+1}^n.$$

Remark 10.2 — Note that for a transport equation to be solved, we need to specify the left boundary condition $a>0$ or the right boundary condition if $a<0$.

For the stability, let us consider:

$$U_j^n\sim\lambda^n e^{ikj\Delta x}.$$

Plugging this into the equation, we have:

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

$$\implies (1-a\nu)\lambda-(1+a\nu)\lambda e^{ik\Delta x}+(1+a\nu)+(1-a\nu)e^{ik\Delta x}$$

$$\implies \lambda=\frac{(1+a\nu)+(1-a\nu)e^{ik\Delta x}}{(1+a\nu)e^{ik\Delta x}+(1-a\nu)}.$$

If we let $a_1=1+a\nu$ and $a_2=1-a\nu$, we have:

$$\lambda=\frac{a_1+a_2e^{ik\Delta x}}{a_1e^{ik\Delta x}+a_2}.$$

$$\implies |\lambda|=\left|\frac{a_1+a_2\cos x+ia_2\sin x}{a_1\cos x+ia\sin x+a_2}\right|=\frac{(a_1+a_2\cos x)^2+a_2^2\sin^2 x}{(a_1\cos x+a_2)^2+a_1^2\sin^2 x}.$$

$$=\frac{a_1^2+2a_1a_2\cos x+a_2^2}{a_1^2+2a_1a_2\cos x+a_2^2}=1.$$

and thus the box scheme is unconditionally stable.

Remark 10.3 — The box scheme is also second order in time and space.

Remark 10.4 — The box scheme has a phase advance error if $|a\nu| \leq 1$ and a phase lag error if $|a\nu| > 1$.

10.2 Leap Frog Scheme

The leap-frog scheme takes the central difference in both time and space. The equation we want to solve is:

$$u_t + f(a)_x = 0.$$

or

$$u_t + au_x = 0.$$

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

This has a CFL condition of $|\nu| < 1$ where $\nu = a\frac{\Delta t}{\Delta x}$.

Remark 10.5 — The leap frog scheme is second order in time and space, as it is using the central difference.

For the stability, we have:

$$\frac{1}{2\Delta t}(\lambda^{n+1} - \lambda^{n-1})e^{ikj\Delta x} + a\frac{1}{\Delta x}(e^{ik\Delta x} - e^{-ik\Delta x})\lambda^n e^{ikj\Delta x} = 0$$

$$\implies \left(\lambda - \frac{1}{\lambda}\right) + a\frac{\Delta t}{\Delta x}(2i\sin k\Delta x) = 0$$

$$\implies \lambda^2 + (2\nu i \sin k\Delta x)\lambda - 1 = 0$$

$$\implies \lambda = \frac{-2\nu i \sin k\Delta x \pm \sqrt{-4\nu^2 \sin^2 k\Delta x + 4}}{2} = -\nu i \sin k\Delta x \pm \sqrt{1 - \nu^2 \sin^2 k\Delta x}$$

$$\implies |\lambda| = \nu^2 \sin^2 k\Delta x + 1 - \nu^2 \sin^2 k\Delta x = 1.$$

Thus the leap frog is stable if $|\nu| \leq 1$.

10.3 Leap-Frog Scheme for Wave Equation

Now let's consider the leap frog scheme for:

$$u_{tt} - a^2 u_{xx} = 0.$$

Remark 10.6 — This second order equation can be converted into a system of first order equation. Let $u_t = -av_x$. We have:

$$u_{tt} = -av_{tx} = a^2 u_{xx}.$$

Thus we can rewrite it as

$$\begin{cases} u_t + av_x = 0 \\ v_t + au_x = 0 \end{cases}.$$

To solve this, let U_j^n be defined on the grid points and $V_{j+\frac{1}{2}}^{n+\frac{1}{2}}$ be defined on the center of the grid squares. With this, we can apply the leap frog scheme:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{V_{j+\frac{1}{2}}^{n+\frac{1}{2}} - V_{j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} = 0.$$

$$\frac{V_{j+\frac{1}{2}}^{n+\frac{3}{2}} - V_{j+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta t} + a\frac{U_{j+1}^{n+1} - U_j^{n+1}}{\Delta x} = 0.$$

Remark 10.7 — This also has second order time and space accuracy, since it is using central difference.

For the stability, let us construct a Fourier solution:

$$(U^n, V^{n+\frac{1}{2}}) = \lambda^n e^{ik\Delta x}(\hat{u}, \hat{v}).$$

where $\hat{u}$, $\hat{v}$ are constants. This gives us:

$$\frac{\hat{u}(\lambda^{n+1}e^{ikj\Delta x} - \lambda^n e^{ikj\Delta x})}{\Delta t} + a\frac{\hat{v}(\lambda^n e^{ik(j+\frac{1}{2})\Delta x} - \lambda^n e^{ik(j-\frac{1}{2})\Delta x})}{\Delta x} = 0.$$

$$\frac{\hat{v}(\lambda^{n+1}e^{ik(j+\frac{1}{2})\Delta x} - \lambda^n e^{ik(j+\frac{1}{2})\Delta x})}{\Delta t} + a\frac{\hat{u}(\lambda^{n+1}e^{ik(j+1)\Delta x} - \lambda^{n+1} e^{ikj\Delta x})}{\Delta x} = 0.$$

$$\implies \begin{cases} \hat{u}(\lambda - 1) + \nu\hat{v}(2i\sin k\frac{\Delta x}{2}) = 0 \\ \hat{v}(\lambda - 1)e^{ik\frac{\Delta x}{2}} + \nu\hat{u}\lambda(e^{ik\Delta x} - 1) = 0 \end{cases}.$$

$$\implies \begin{bmatrix} \lambda - 1 & 2i\sin\frac{k\Delta x}{2} \\ 2i\lambda\nu\sin\frac{k\Delta x}{2} & \lambda - 1 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = 0.$$

for nontrivial solutions, this requires

$$(\lambda - 1)^2 + \lambda 4\nu^2 \sin^2\frac{k\Delta x}{2} = 0.$$

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

solving this would give us $|\lambda| = 1$ which is stable.

Remark 10.8 — We can also do direct discretization of the scheme which is second order in time and space and is stable.

11 October 20th, 2020

11.1 Elliptic Equation

Let us consider a simple elliptic equation:

$$\begin{cases} -u'' = f \\ u(0) = u(1) = 0 \end{cases}.$$

This can be approximated by central difference

$$-\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} = f_j.$$

where h is the step size this can be expressed in the form of a tridiagonal linear system. In 2D, we have a similar construction with:

$$\frac{U_{i+1,j} - 2U_{i,j} + U_{j-1,j}}{(\Delta x)^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{j,j-1}}{(\Delta y)^2} = f_{i,j}.$$

This would give us a system of equation $AU = F$, with $(M-1) \times (N-1)$ unknowns and A being a matrix with 5 diagonals, similar to the 2D hyperbolic case.

Remark 11.1 — How to solve this linear system efficiently will be discussed in next semester.

11.2 Finite Element Method

Finite different methods need the domain to be a rectangular mesh, making it not effective for arbitrary domains. This is where **finite element method** comes in. The approach of this finite element method is very different from finite difference. In finite difference, we are discretizing the derivative. However, in finite element, we are discretizing the solution space.

Let us consider the same equation as before:

$$\begin{cases} \Delta u = f \\ u\big|_{\partial\Omega} = 0 \end{cases}.$$

Let us define a **functional**:

$$I(v) = \int_\Omega (\frac{1}{2}|\nabla v|^2 + fv)\ dxdy.$$

where ∇v is the **gradient** of v, i.e. $\nabla v = (\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y})$.

Remark 11.2 — This is a functional because the parameter v is a function.

We define this over the function space:

$$H_0^1(\Omega) = \{v | v, \nabla v \in L^2(\Omega),\ v\big|_{\partial\Omega} = 0\}.$$

Remark 11.3 — $H_0^1(\Omega)$ is set of functions that are square integrable (in L^2) and satisfy the boundary condition. Superscript 1 means that the first derivative is square integrable, and subscript 0 means it is zero at the boundary.

We would like to find the minimum over the function space:

$$\min_{v \in H_0^1(\Omega)} I(v).$$

In order to find the minimum, we want to find the derivative and set it to zero. This is a necessary condition.

Let's assume that function v_0 is a minimum of $I(v)$, meaning that:

$$I(v_0 + \epsilon u) \geq I(v_0), \quad \forall \epsilon,\ u \in H_0^1(\Omega).$$

Now we create a function:

$$J(\epsilon) = I(v_0 + \epsilon u).$$

where $\epsilon = 0$ is a minimum of $J(\epsilon)$, i.e. $J(\epsilon) \geq J(0)$ for all ϵ. This means that:

$$\left.\frac{dJ}{d\epsilon}\right|_{\epsilon=0} = 0.$$

From the definition of I, we have:

$$J(\epsilon) = I(v_0 + \epsilon u) = \int_\Omega \frac{1}{2}|\nabla v_0 + \epsilon \nabla u|^2 + f \cdot (v_0 + \epsilon u) dx dy.$$

Expanding, we have:

$$\int_\Omega \frac{1}{2}|\nabla v_0|^2 + \epsilon \nabla v_0 \nabla u + \frac{1}{2}\epsilon |\nabla u|^2 + \int_\Omega f(v_0) + \epsilon \int_\Omega f u.$$

Taking the derivative and evaluating at zero, we have:

$$\left.\frac{dJ}{d\epsilon}\right|_{\epsilon=0} = \int_\Omega \nabla v_0 \nabla u + \int_\Omega f u = 0.$$

From the divergence theorem:

$$\int_\Omega \operatorname{div} \vec{F}(x)\ dx = \int_{\partial\Omega} \vec{F} \cdot \vec{n}\ ds.$$

Taking $\vec{F} = v \nabla u$, we have:

$$\operatorname{div} \vec{F} = \nabla v \nabla u + v \Delta u.$$

Thus, we have:

$$\int_\Omega \operatorname{div} \vec{F}(x)\ dx = \int_\Omega \nabla v \nabla u + \int_\Omega v \Delta u = \int_{\partial\Omega} \vec{F} \cdot \vec{n}\ ds = 0.$$

because we are integrating over the boundary which is zero. Thus, we have:

$$\begin{aligned}\left.\frac{dJ}{d\epsilon}\right|_{\epsilon=0} &= \int_\Omega \nabla v_0 \nabla u + \int_\Omega fu = 0 \\ &= -\int_\Omega \Delta v_0 u + \int_\Omega fu = 0 \\ &= -\int_\Omega (\Delta v_0 - f)u \; dxdy = 0.\end{aligned}$$

Since u is arbitrary, we must have:

$$\Delta v_0 - f = 0.$$

As such:

$$\begin{cases}\Delta v_0 = f \\ v_0\big|_{\partial\Omega} = 0\end{cases} \iff \int_\Omega \nabla v_0 \nabla u + \int_\Omega fu = 0 \quad \text{for all } u \in H_0^1(\Omega).$$

Definition 11.4. The right hand side is a **weak formulation** of the original PDE.

Remark 11.5 — If we have a minimum of the functional, then we have a solution to desired equation.

Remark 11.6 — This is called the weak formulation because we only involve the first derivative.

Let us assume there is a basis $\{\phi_1, \phi_2, \dots \phi_N\}$ for $H_0^1(\Omega)$. Let us consider a finite dimensional subspace of $H_0^1(\Omega)$ (thus discretize the function space):

$$S_N = \{\phi_1, \phi_2, \dots, \phi_N\} \subset H_0^1(\Omega).$$

For any $v_0 \in S_N$, we can express it in terms of the basis functions:

$$v_0 = \sum_{i=1}^N v_i \phi_i.$$

Since the weak formulation must hold true for any $u \in S_N$, we can choose $u = \phi_j$:

$$\int_\Omega \nabla v_0 \nabla \phi_j + \int_\Omega f\phi_j = 0, \qquad \text{for } j = 1, 2, \dots, N.$$

Expanding v_0, we have:

$$\sum_i \int_\Omega v_i \nabla\phi_i \nabla\phi_j + \int_\Omega f\phi_j = 0.$$

Giving us a linear system $A\vec{v} = \vec{b}$ where:

$$A = (a_{ij}), \quad a_{ij} = \int_\Omega \nabla\phi_i \nabla\phi_j.$$

$$\vec{b} = (b_j), \quad b_j = \int_\Omega f\phi_j.$$

$$\vec{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}.$$

12 October 22th, 2020

12.1 Finite Element Method Cont.

Recall that the finite element method transform the PDE into a equivalent weak formulation:

$$\int_\Omega \nabla v_0 \nabla u + \int_\Omega fu = 0 \quad \text{for all } u \in H_0^1(\Omega).$$

With this, we can discretize the function space $H_0^1(\Omega)$ by expressing it in terms of its basis and truncating it to only N basis functions. If we find a solution $v_0 \in S_N$ which satisfies this weak formulation, we would have:

$$\sum_i \int_\Omega v_i \nabla \phi_i \nabla \phi_j + \int_\Omega f\phi_j = 0.$$

Giving us a linear system $A\vec{v} = \vec{b}$ where:

$$A = (a_{ij}), \quad a_{ij} = \int_\Omega \nabla\phi_i \nabla\phi_j.$$

$$\vec{b} = (b_j), \quad b_j = \int_\Omega f\phi_j.$$

$$\vec{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}.$$

Remark 12.1 — The resulting linear system of equations depends on the choice of the basis functions and the subspace S_N.

Remark 12.2 — Note that the solution is not a discretized solution since we are solving for the coefficients of the basis set. Also, note that this is an approximate solution because we did truncation on the number of basis functions.

With this, we would like to investigate the convergence of this method and the construction of basis functions.

Let us consider the 1D elliptical PDE:

$$\begin{cases} -u'' = f & 0 \le x \le 1 \\ u(0) = u(1) = 0 \end{cases}.$$

This has the corresponding weak formulation:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } \int_0^1 u'v' \, dx = \int_0^1 fv \, dx, \quad \forall v \in H_0^1(\Omega).$$

We have:

$$H_0^1(\Omega) = \{v | v, v' \in L^2, v(0) = v(1) = 0\}.$$

Now we would like to find a N dimensional subspace of this function space. To do this, let us consider S_N to be the set of continuous piecewise linear functions that vanishes at $x = 0, 1$.

Remark 12.3 — Piecewise linear means that the function is linear within the subinterval $[x_i, x_{i+1}]$.

Let us define a "hat" function at each grid point x_j:

$$\phi_j(x) = \begin{cases} 0 & x < x_{j-1} \vee x > x_{j+1} \\ \frac{x - x_{j-1}}{x_j - x_{j-1}} & x \in [x_{j-1}, x_j] \\ \frac{x_{j+1} - x}{x_j - x_{j+1}} & x \in [x_j, x_{j+1}] \end{cases}.$$

Claim 12.4. $\{\phi_1, \phi_2, \ldots, \phi_N\}$ forms a basis of S_N.

Proof. To show that this is a basis, we need to show that they are linearly independent and that they are complete.

Independence: Suppose $c_1\phi_1 + c_2\phi_2 + \ldots + c_N\phi_N = 0$. Consider at point x_j:

$$c_1\phi_1 + c_2\phi_2 + \ldots + c_N\phi_N = c_j\phi_j(x_j) = 0 \implies c_j = 0.$$

this is true for all $j = 1, 2, \ldots, N$, meaning that $\{\phi_j\}$ is linearly independent.

Complete: Let $v \in S_N$. Take $c_j = v(x_j)$:

$$\sum c_j\phi_j = \sum v(x_j)\phi_j(x) = v(x).$$

Since they are equal at all x_i and linear in $[x_i, x_{i+1}]$, they are equal everywhere.

□

With these hat functions, we can write down the linear system:

$$A\vec{u} = \vec{b}.$$

where

$$a_{ij} = \int_0^1 \phi_i'\phi_j' \, dx, \quad \vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}, \quad b_j = \int_0^1 f\phi_j \, dx.$$

Let us assume that we have a uniform grid $\{x_i\}$, where $x_i = i \cdot h$ and $h = \frac{1}{N}$. With this, we have:

$$\phi_i(x) = \begin{cases} 0 & x \le x_{i-1} \\ \frac{x - x_{x-1}}{h} & x_{i-1} \le x \le x_i \\ -\frac{x - x_{x+1}}{h} & x_i \le x \le x_{i+1} \\ 0 & x \ge x_{i+1} \end{cases} \implies \phi_i'(x) = \begin{cases} 0 & x \le x_{i-1} \\ \frac{1}{h} & x_{i-1} \le x \le x_i \\ -\frac{1}{h} & x_i \le x \le x_{i+1} \\ 0 & x \ge x_{i+1} \end{cases}.$$

Thus, we have:

$$a_{ij} = \int_0^1 \phi_i'\phi_j' \, dx.$$

Note that this is non-zero only when there is overlap between ϕ_i and ϕ_j. This only happens between ϕ_i and ϕ_{i+1}, giving us:

$$a_{i,i+1} = \int_0^1 \phi_i' \phi_{i+1}' \, dx = \int_{x_i}^{x_{j+1}} \phi_i' \phi_{i+1}' \, dx = \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h}\right)\left(\frac{1}{h}\right) dx = -\frac{1}{h^2} \cdot h = -\frac{1}{h}$$

$$a_{i,i-1} = \int_0^1 \phi_i' \phi_{i-1}' \, dx = \int_{x_{i-1}}^{x_j} \phi_i' \phi_{i-1}' \, dx = \int_{x_{i-1}}^{x_i} \left(\frac{1}{h}\right)\left(-\frac{1}{h}\right) dx = -\frac{1}{h^2} \cdot h = -\frac{1}{h}.$$

$$a_{i,i} = \int_0^1 \phi_i' \phi_i' \, dx = \int_{x_{i-1}}^{x_{j+1}} \phi_i' \phi_i' \, dx = \int_{x_{i-1}}^{x_{i+1}} \left(\frac{1}{h}\right)\left(\frac{1}{h}\right) dx = \frac{1}{h^2} \cdot 2h = \frac{2}{h}.$$

Thus, we have:

$$A = \frac{1}{h}\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}.$$

For $\vec{b}$, we have $f_i \sim f(x_i)$:

$$b_i = \int_0^1 f\phi_i \, dx = \int_{x_{i-1}}^{x_{i+1}} f_i \phi_i \, dx = f_i \int_{x_{i-1}}^{x_{i+1}} \phi_i \, dx = f_i \frac{1}{2}(2h) = f_i h.$$

Thus, we have:

$$-\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{bmatrix}.$$

Remark 12.5 — This is very similar to the finite different scheme we saw before because we assume a uniform grid and use the hat functions for the basis functions. Since this is the same as the finite different method, it converges and is second order in both time and space.

Remark 12.6 — The order of accuracy depends on S_N, as if we instead used a quadratic piecewise function, we would have a different order error term.

Note that our finite element solution is not differentiable in the classical sense because there are corners. This leads to the idea of a weak derivative which will be introduced in the next class.

13 October 27th, 2020

13.1 Weak Derivative

From last time, we have that the general approach to solving a PDE using finite element methods is:

1. Find the weak formulation of the equation
2. Define the solution space
3. Find the N-dimensional subspace
4. Reduce the weak formulation by a linear system

Recall that the weak formulation is to find $u \in S_N$ s.t.:

$$\int_0^1 u'v' = \int_0^1 fv \, dx \quad \forall v \in S_N.$$

In our previous example, we had S_N being continuous piecewise linear functions. However these piecewise linear functions are not differentiable at the corners, meaning we cannot interpret derivatives in the point wise sense. To do this, we will introduce the concept of a weak derivative.

Definition 13.1 (weak derivative)**.** w is a **weak derivative** of u if w is integrable and:

$$\int_{-\infty}^{+\infty} uv' = -\int_{-\infty}^{+\infty} wv \, dx, \quad \forall v \in C_0^\infty(\mathbb{R}).$$

Remark 13.2 — If u, v are differentiable in the classical sense, assuming zero boundary conditions, then:

$$\int_{-\infty}^{+\infty} u'v \, dx = uv\Big|_{-\infty}^{+\infty} - \int_{-\infty}^{+\infty} uv' \, dx = -\int_{-\infty}^{+\infty} uv' \, dx.$$

This means that if u is differentiable, then $w = u'$.

Example 13.3 (Weak derivative of hat function)
Consider the hat function:

$$u(x) = \begin{cases} 0 & x < -1 \text{ or } x > 1 \\ 1 + x & -1 \leq x < 1 \\ 1 - x & 0 \leq x \leq 1 \end{cases}.$$

Plugging this in, we have:

$$\begin{aligned} \int_{-\infty}^{+\infty} uv' \, dx &= \int_{-1}^{0} (1+x)v' \, dx + \int_{0}^{1} (1-x)v' \, dx \\ &= (1+x)v\Big|_{-1}^{0} - \int_{-1}^{0} 1 \cdot v \, dx + (1-x)v\Big|_{0}^{1} + \int_{0}^{1} 1 \cdot v \, dx \\ &= v(0) - \int_{-1}^{0} v \, dx - v(0) + \int_{0}^{1} v \, dx \\ &= -\int_{-\infty}^{+\infty} wv \, dx. \end{aligned}$$

where:

$$w(x) = \begin{cases} 0 & x < -1 \text{ or } x > 1 \\ +1 & -1 \leq x < 1 \\ -1 & 0 \leq x \leq 1 \end{cases}.$$

As such, the hat function is weakly differentiable.

Example 13.4 (Piecewise smooth function that is not weakly differentiable)
Consider the step function:

$$u(x) = \begin{cases} 1 & x < 0 \\ -1 & x \geq 0 \end{cases}.$$

We have:

$$\int_{-\infty}^{+\infty} uv' \, dx = \int_{-\infty}^{0} v' \, dx - \int_{0}^{\infty} v' \, dx = v(0) - (-v(0)) = 2v(0) = \int_{-\infty}^{+\infty} wv \, dx.$$

In this situation, w is the delta function at 0. However, it is not integrable in the classical sense. As such, u is not weakly differentiable.

Theorem 13.5
If f is piecewise smooth and continuous, and f' is integrable in each Ω_i, then f is weakly differentiable.

Proof. Let f be smooth on Ω_i, $[a, b] = \bigcup_i \Omega_i$. We define g by $g\big|_{\Omega_i}$ which is the classical

derivative of f on Ω_i. Now we will show that w is the weak derivative of f. By definition, we need to show:

$$\int_a^b gv\ dx = -\int_a^b fv'\ dx, \quad v \in C_0^\infty(a,b).$$

To do this, we have:

$$\begin{aligned}
\int_a^b gv\ dx &= \sum_i \int_{\Omega_i} gv\ dx = \sum_i \int_{\Omega_i} f'v\ dx = \sum_i \int_{x_i}^{x_{i+1}} f'v\ dx \\
&= \sum_i fv\Big|_{x_i}^{x_{i+1}} - \sum_i \int_{x_i}^{x_{i+1}} fv'\ dx \\
&= \underbrace{\sum_i (fv(x_{i+1}) - fv(x_i))}_{=0 \text{ since } v \text{ is continuous}} - \sum_i \int_{x_i}^{x_{i+1}} fv'\ dx \\
&= -\int_a^b fv'\ dx
\end{aligned}$$

.

as desired. Thus f is weakly differentiable with a weak derivative g. □

We can generalize this to higher space dimensions:

Definition 13.6 (weak partial derivative)**.** If ω_i is integrable over Ω and:

$$\int_\Omega \omega_i v\ dx = -\int_\Omega u\frac{\partial v}{\partial x_i}\ dx \quad \forall v \in C_0^\infty(\Omega).$$

then w_i is the **weak partial derivative** of u with respect to x_i.

We can also define higher order derivatives:

Definition 13.7. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ where α_j are non-negative integers. Let us define $|\alpha| = \sum_{j=1}^n \alpha_j$, then w_α, integrable in Ω is the multivariable weak derivative of u if:

$$\int_\Omega u\partial^\alpha v\ d\Omega = (-1)^{|\alpha|}\int_\Omega w_\alpha v\ d\Omega, \quad \forall v \in C_0^\infty(\Omega).$$

with:

$$w_\alpha = \frac{\partial^{|\alpha|}u}{\partial^{\alpha_1}x_1\partial^{\alpha_2}x_2\ldots\partial^{\alpha_n}x_n}.$$

Remark 13.8 — Consider the case for 2nd order derivative. Integration by parts gives us:

$$\int_{-\infty}^{+\infty} u''v\ dx = -\int_{-\infty}^{+\infty} u'v'\ dx = \int_{-\infty}^{+\infty} uv''\ dx.$$

Then if there exists a w that is integrable such that:

$$\int_{-\infty}^{+\infty} wv\ dx = \int_{-\infty}^{+\infty} uv''\ dx.$$

then u has a second order weak derivative equal to w.

Example 13.9
The hat function is not second order weakly differentiable, since it's first order weak derivative is a step function, which is not weakly differentiable.

Remark 13.10 — A piecewise quadratic function is not necessarily second order weakly differentiable, as you need the additional constraint that the first order derivative at the boundaries are the same.

Definition 13.11. Let us define

$$L^2(\Omega) = \{u : \text{ integrable}, \int_\Omega u^2 \, dx < +\infty\}, \quad \|u\|_{L^2} = \left(\int_\Omega u^2 \, dx\right)^{\frac{1}{2}}.$$

Then the **Soboler space** is:

$$H^1(\Omega) = \{u : u \in L^2, \quad u \text{ has weak first order derivate which are in } L^2(\Omega)\}.$$

$$\|u\|_{H^1(\Omega)} = \left(\int_\Omega u^2 + \sum_{i=1}^n (\partial x_i u)^2 \, dx\right).$$

Remark 13.12 — The solution space for FEM is a Soboler space.

Some of the properties of $H^1(\Omega)$ include:

- Functions in $H^1(a,b)$ are continuous. $(\Omega = (a,b))$
- f is piecewise smooth and f is continuous implies that $f \in H^1(\Omega)$
- Let $C^1(\Omega) = \{f$ which are continuously differentiable in $\Omega\}$, then $C^1(\Omega)$ is dense in $H^1(\Omega)$

13.1.1 Treatment of Boundary Conditions

For now we have only been considering zero boundary conditions. If instead we have a inhomogeneous Dirichlet boundary condition, e.d.

$$\begin{cases} -u'' = f \\ u(0) = a \\ u(1) = b \end{cases}.$$

Then we would have:

$$u(x) = u_H + u_I, \quad u_I = a + (b-a)x.$$

where u_H is zero at the boundary. With this, we have:

$$\begin{aligned} u_H &= u(x) - u_I \\ -u_H'' &= -u'' + u_I'' = -u'' = f \\ u_H(0) &= u(0) - u_I(0) = a - a = 0 \\ u_H(1) &= u(1) - u_I(1) = b - b = 0. \end{aligned}$$

Thus we would have:

$$\begin{cases} -u_H'' = f \\ u_H(0) = u_H(1) = 0 \end{cases}.$$

14 October 29th, 2020

14.1 Neumann Boundary Conditions for FEM

Recall for finite different, to account for Neumann Boundary conditions, we would introduce ghost points. For FEM, the functional is the same:

$$E(u) = \int_0^1 (u'^2 - 2uf)\ dx$$

but with no boundary conditionals imposed explicitly, we would have the solution space:

$$H^1(\Omega) = \{u | u, u' \in L^2(\Omega)\}.$$

Remark 14.1 — For the previous case we would require $u(0) = u(1) = 0$. As such, this is a much bigger solution space than the Dirichlet boundary conditions.

If we once again try and minimize this functional:

$$\min_{u \in H^1(\Omega)} E(u),$$

assuming u is a minimizer:

$$\begin{aligned} \frac{d}{d\epsilon} E(u + \epsilon v)\bigg|_{\epsilon=0} &= 0, \forall v \in H^1(\Omega) \\ \implies E(u + \epsilon v) &= \int_0^1 (u' + \epsilon v')^2 - 2(u + \epsilon v) f \\ &= \int_0^1 (u'^2 + 2\epsilon u'v' + \epsilon^2 v'^2 - 2uf - 2\epsilon v f)\ dx \\ \implies \frac{d}{d\epsilon} E(u + \epsilon v)\bigg|_{\epsilon=0} &= 2\int_0^1 u'v' - 2\int_0^1 vf = 0. \end{aligned}$$

Thus, this gives us the weak formulation:

$$\int_0^1 u'v'\ dx - \int_0^1 vf\ dx = 0, \forall v \in H^1(\Omega).$$

Note that this is the same as before. If we do integration by parts, we would get:

$$v'u\Big|_0^1 - \int_0^1 u''v - \int_0^1 vf = 0.$$

for any v. Let first restrict $v \in H_0^1(\Omega)$, i.e. $v\big|_{\partial\Omega} = 0$, we would get:

$$\int_0^1 (-u'' - f)v \, dx = 0 \implies -u'' - f = 0.$$

which is the PDE. Alternatively, if we consider arbitrary v, if u is the minimizer that satisfies the boundary condition, we must have:

$$u'v\Big|_0^1 = 0 \implies u'(0) = 0, u'(1) = 0.$$

This means that for a solution space that does not restrict the boundary condition, the minimizer of the functional satisfies the Neumann boundary condition. Thus, this is called the **natural boundary condition**.
Now that we have the weak formulation, we have to discretize the solution space. Let us consider:

$$S_N = \{\text{piecewise linear funciton space}\} \subset H^1(\Omega).$$

with the hat functions $\{\phi_j(x)\}, j = 0, 1, 2, \dots N$.

Remark 14.2 — We add two more hat functions ϕ_0 and ϕ_N, with:

$$\phi_0(x) = \begin{cases} \frac{x - x_0}{x_1 - x_0} & x_0 \le x \le x_1 \\ 0 & \text{otherwise} \end{cases}.$$

$$\phi_N(x) = \begin{cases} \frac{x_N - x}{x_N - x_{N-1}} & x_{N-1} \le x \le x_N \\ 0 & \text{otherwise} \end{cases}.$$

With this, we want to find $u \in S_N$ such that:

$$\int_0^1 (u'\phi_j' - f\phi_j) \, dx = 0, \forall j = 0, 1, 2, \dots, N.$$

$$\implies \int_0^1 u_i \phi_i' \phi_j' - \int_0^1 f\phi_j = 0.$$

$$\implies \sum_i a_{ij} u_i - b_j = 0 \implies Ax = b.$$

where:

$$a_{ij} = \int_0^1 \phi_i' \phi_j' \, dx, \quad b_j \int_0^1 f\phi_j, \quad x = \begin{bmatrix} u_0 \\ u_1 \\ u_N \end{bmatrix}.$$

Remark 14.3 — Homework: show taht $\{\phi_j, j = 0, 1, \ldots, N\}$ forms a basis of S_N, compute a_{ij} for uniform grid on $[0, 1]$.

14.2 FEM for Polygon Domain

One benefit of FEM is that we are not limited to rectangular domains like finite difference methods. Let us consider an arbitrary polygon domain. With this, we triangulate the polygon, turning it into triangles, with interior nodes being the vertices between traingles. With this, we define basis functions at each interior node:

$$\phi_i(x_j) = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}.$$

and is piecwise linear in each triangle. This is the 2D generalization of the hat function, called the **tent functions**.

Remark 14.4 — If we have an arbitrary domain, we can approximate it with a polygon domain by adding points on the boundary.

Remark 14.5 — Piecewise linear functions in this context defines a plan, meaning that the tent function is "gluing" together many planes.

Remark 14.6 — We can show that $\{\phi_i, i = 1, 2, \ldots N\}$ forms a basis of S_N.

As such, we can apply the weak formulation in the same way as before.

15 November 3th, 2020

15.1 Error Analysis of Finite Element Methods

For finite different methods, we can get the error analysis using the Taylor expansion. However, for FEM, it is a bit more complicated. Recall that the truncation error is when we truncate the infinitely dimensional solution space $H_0^1(\Omega)$ into a finite-dimensional subspace $S_N \subset H_0^1(\Omega)$. Recall that for the problem:

$$\begin{cases} -\Delta u = f \\ u\big|_{\partial\Omega} = 0 \end{cases}.$$

we have an equivalent weak formulation of finding $u \in H_0^1(\Omega)$ such that:

$$\int_\Omega \nabla u \nabla v \, dx = \int_\Omega fv, \quad \forall v \in H_0^1(\Omega).$$

Let us define a **bi-linear form**:

$$a(u, v) = \int_\Omega \nabla u \nabla v \, dx.$$

Remark 15.1 — Bi linear form means that $a(u,v)$ is linear in both u and v.

Let us also define a norm:

$$\|v\|_E = \sqrt{a(v,v)} = \left(\int_\Omega (\nabla v)^2 \, dx\right)^{\frac{1}{2}}.$$

Let us consider a finite dimensional subspace $S_N \subset H_0^1(\Omega)$, meaning that the finite element solution u_N is such function that satisfies:

$$a(u_N, v) = (f, v), \quad \forall v \in S_N.$$

where $(u,v) = \int_\Omega uv$ is the inner product. For the exact solution u, we would have:

$$\begin{aligned} a(u,v) &= (f,v) \quad \forall v \in S_N. \\ \implies a(u - u_N, v) &= 0 \quad \forall v \in S_N \subset H_0^1(\Omega). \end{aligned}$$

Let us consider the error, i.e. the difference between u and u_N:

$$\begin{aligned} \|u - u_N\|_E^2 &= a(u - u_N, u - u_N) \\ &= a(u - u_N, u - v + v - u_N), \quad \forall v \in S_N \\ &= a(u - u_N, u - v) + a(u - u_N, v - u_N) \\ &= a(u - u_N, u - v) \qquad (\text{since } v - u_N \in S_N) \\ &\le \|u - u_N\|_E \|u - v\|_E, \quad \forall v \in S_N \quad (\text{Cauchy Schwartz inequality}). \end{aligned}$$

As such, we have:

$$\|u - u_N\|_E \le \|u - v\|_E \quad \forall v \in S_N.$$

This means that out of all solutions in S_N, u_N is the best approximation to the exact solution.

To estimate the error, we will use a duality argument. Let w be a solution of the dual problem:

$$\begin{cases} -\Delta w = u - u_N \\ w(0) = w(1) = 0 \end{cases}.$$

Let us consider the L^2 norm:

$$\|v`\|_{L^2} = \left(\int_\Omega |v|^2 \, dx\right)^{\frac{1}{2}}.$$

we have:

$$\begin{aligned} \|u - u_N\|_{L^2}^2 &= \int |u - u_N|^2 \, dx \\ &= (u - u_N, u - u_N) \\ &= (u - u_N, -w'') \\ &= ((u - u_N)', w') \\ &= a(u - u_N, w) \\ &= a(u - u_N, w - v + v), \quad \forall v \in S_N \\ &= (u - u_N, w - v) + a(u - u_N, v) \\ &\le \|u - u_N\|_E \|w - v\|_E. \end{aligned}$$

As such, we have:

$$\|u-u_N\|_2 \leq \frac{\|u-u_N\|_E\|w-v\|_E}{\|u-u_N\|_2} = \frac{\|u-u_N\|_E\|w-v\|_E}{\|w\|_2}, \quad \forall v \in S_N.$$

If we have the approximation assumption:

$$\int_{w\in S_N} \|w-v\|_E \leq \epsilon\|w''\|_2 \implies \|u-u_N\|_2 \leq \epsilon\|u-u_N\|_E.$$

Remark 15.2 — This approximation assumption is that any second order differentiable function can be approximated by a function in S_N.

Applying the approximation assumption again, we have:

$$\|u-u_N\|_E \leq \epsilon\|u-v\|_E \leq \epsilon\|u''\|_2.$$

$$\implies \|u-u_N\|_2 \leq \epsilon^2\|u''\|_2 = \epsilon^2\|f\|_2.$$

As such, it is second order accurate in L^2 norm.

Consider $0 = x_0 < x_1 < x_2 < \ldots < x_N = 1$ a partition of $[0,1]$. Let S_N be the continuous piecewise linear function space with hat basis functions $\phi_1, \ldots, \phi_{N-1}$. Take any function $u \in H_0^1(\Omega)$. Let $u_I(x) = \sum_{i=1}^{N-1} u(x_i)\phi_i(x)$.

Theorem 15.3

$$h = \max_i(x_{i+1}-x_i) \implies \|u-u_I\|_E \leq ch\|u''\|_2.$$

Proof.

$$\|u-u_I\|_E^2 = \int_0^1 (u'-u_I')^2\,dx = \sum_{j=1}^{N}\int_{x_{j-1}}^{x_j}(u'-u_I')^2\,dx.$$

Let $e(x) = u(x) - u_I(0)$, with $e(x_j) = 0$. For $[x_{j-1}, x_j]$:

$$\exists \xi \in (x_{j-1}, x_j), \text{ such that } e'(\xi) = 0, e'(y) = \int_\xi^y e''(x)\,dx.$$

from the fundamental theorem of calculus. As such, we have:

$$|e'(y)| \leq \int_\xi^y |e''(x)|\,dx \leq \left(\int_\xi^y dx\right)^{\frac{1}{2}}\left(\int_\xi^y |e''(x)|^2\,dx\right)^{\frac{1}{2}}.$$

$$= (y-\xi)^{\frac{1}{2}}\left(\int_\xi^y |e''(x)|^2\,dx\right)^{\frac{1}{2}} \leq (y-\xi)^{\frac{1}{2}}\left(\int_{x_{j-1}}^{x_j} |e''(x)|^2\,dx\right)^{\frac{1}{2}}.$$

Thus, we have:

$$\begin{aligned}\int_{x_{j-1}}^{x_j} |e'(y)|\ dy &\leq \int_{x_{j-1}}^{x_j} (y-\xi))\left(\int_{x_{j-1}}^{x_j} |e''(x)|^2\ dx\right)\ dy \\ &= \frac{1}{2}(y-\xi)^2\Big|_{x_{j-1}}^{x_j} \int_{x_{j-1}}^{x_j} |e''(x)|^2\ dx \\ &\leq c(x_j - x_{j-1})^2 \int_{x_{j-1}}^{x_j} |e''(y)|^2\ dx \quad \text{for some constant } c \\ &\leq ch^2 \int_{x_{j-1}}^{x_j} (u'')^2\ dx \quad \text{since } e'' = u''.\end{aligned}$$

Thus, we have:

$$\left(\int_0^1 (u' - u_I')^2\ dy\right)^{\frac{1}{2}} \leq ch\left(\int_0^1 (u'')^2\ dx\right)^{\frac{1}{2}}.$$
$$\implies \|u - u_I\|_E \leq ch\|u''\|_2.$$

If we choose $v = w_I \in S_N$ to be the interpolation function of w, we would have:

$$\|w - v\|_E = \|w - w_I\|_E \leq ch\|w''\|_2.$$

□

As such, from before, we would have:

$$\|u - u_N\|_2 \leq ch\|u - u_N\|_E \leq ch^2\|u''\|_2 = ch^2\|f\|_2.$$

As such, this finite element scheme is second order accurate.

Remark 15.4 — We can increase this accuracy by choosing different basis functions, e.g. piecewise quadratic functions.

16 November 10th, 2020

16.1 Iterative Methods to Solve Linear Systems

Recall that a linear system is of the form:

$$Ax = b, \quad A = (a_{ij}), \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}.$$

Where A is a $n \times n$ matrix. The standard approach is **Gaussian Elimination**. Recall that this is done by applying row operations to:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} & b_1 \\ a_{21} & a_{22} & & & \vdots \\ \vdots & & \ddots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} & b_N \end{bmatrix}.$$

into an upper triangular matrix, and then doing back-substitution. This can be done for any matrix, even non-square matrices. However, this method is slow, as it has a computational cost of $O(N^2)$, making it not practice for many calculations.

Example 16.1

Let us assume we are solving the laplacian on unit cube $[0,1]^3$:

$$\begin{cases} \Delta u = f = u_{xx} + u_{yy} + u_{zz} \\ u\big|_{\partial\Omega} = 0 \end{cases}$$

with 100 points in each dimension. As such, $N = 100^3$, meaning that the cost on the order 10^{12}.

As such, we need to develop methods to solving linear system that are more efficient. One such type of methods are iterative methods. Let us assume we want to solve the equation:

$$f(x) = x^5 + x^4 + 3x^2 + 1 = 0.$$

To solve for the roots of this equation, we can use the **Newton's Method**, for which we have an initial guess x_0. If $f(x_0) = y_0 = 0$, we are done. Otherwise, we use the Taylor expansion at the point by considering the tangent line at (x_0, y_0), and taking it's x-intercept, with:

$$y = f(x_0) + f'(x_0)(x - x_0) = 0 \implies x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

We repeat this process with:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

If $x_n \to x^*$, then:

$$x^* = x^* - \frac{f(x^{\cdot *})}{f'(x^*)} \implies f(x^*) = 0.$$

meaning it converges to a root of $f(x)$.

Remark 16.2 — For the above case, we assume $f'(x^*) \neq 0$, this is a reasonable assumption. We can also slightly change this method to account for this situation.

Remark 16.3 — Newton's method depends on the initial guess x_0. In addition, it only converges (if it does) to one root, where as $f(x)$ above has 5 roots.

Remark 16.4 — Iterative methods generally have the following approach:

- We have a problem that is difficult to solve or too expensive
- We attempt to solve an approximate problem that is easy (e.g. tangent line for Newton's method)

- We repeatedly solve this simpler problem

Let us consider:

$$Ax = b.$$

where:

$$A = \begin{bmatrix} d_1 & & * \\ & \ddots & \\ * & & d_n \end{bmatrix} = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix} - \begin{bmatrix} 0 & & 0 \\ & \ddots & \\ -* & & 0 \end{bmatrix} - \begin{bmatrix} 0 & & -* \\ & \ddots & \\ 0 & & 0 \end{bmatrix} = D - L - U.$$

Remark 16.5 — D is a diagonal matrix, L is a lower triangular matrix, and U is an upper triangular matrix

As such, we have:

$$\begin{aligned} Ax = Dx - Lx - Ux &= b \\ \implies Dx &= b + Lx + Ux \end{aligned}$$

Given an initial guess x_0, we have:

.

$$\begin{aligned} Dx &= b + Lx_0 + Ux_0 \\ x_1 &= D^{-1}(b + Lx_0 + Ux_0) \\ \implies x_{n+1} &= D^{-1}(b + Lx_n + Ux_n). \end{aligned}$$

This method is called the **Jacobi iteration**, and it has $O(N)$ computation cost, since we only have compute the diagonal for D^{-1} and back-substitution.

Remark 16.6 — Note that D is a diagonal matrix, meaning that D^{-1} is very easy to solve:

$$D^{-1} = \begin{bmatrix} \frac{1}{d_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{d_n} \end{bmatrix}.$$

If $x_n \to x_*$, we have:

$$\begin{aligned} Dx_* &= D^{-1}(b + Lx_* + Ux_*) \\ \implies Ax_* &= Dx_* - Lx_* - Ux_* = b. \end{aligned}$$

meaning that x_* is a solution to $Ax = b$.

However, we still have to investigate its convergence, since if it requires $O(N)$ iterations to converge, then it is no better than Gaussian Elimination.

If A is upper triangular, then we can solve the problem using back-substitution in $O(N)$, meaning we only need $A = D - U$. As such, we have:

$$\begin{aligned} Ax = Dx - Lx - Ux &= b \\ Dx - Ux &= b + L_x \\ (D-U)x &= b + Lx. \end{aligned}$$

Given an initial guess x_0, we have:

$$\begin{aligned} (D-U)x_1 &= b + Lx_0 \\ \implies x_1 &= (D-U)^{-1}(b + Lx_0) \\ \implies x_{n+1} &= (D-U)^{-1}(d + Lx_n). \end{aligned}$$

This method is called the **Gauss-Seidel Iteration**, and also has $O(N)$ cost per iteration.

Remark 16.7 — Note that we can solve $(D-U)^{-1}$ efficiently.

Remark 16.8 — There is a different version where we put U on the right hand side instead of L, giving us:

$$x_{n+1} = (D-L)^{-1}(b + Ux_n).$$

There is another version called **SOR iteration** where instead of having $D - L$, we have $D - \omega L$ for $0 < \omega < 2$. This gives u:

$$\begin{aligned} Dx_{n+1} &= (1-\omega)Dx_n + \omega(b + Lx_{n+1} + Ux_n) \\ \implies (D - \omega L)x_{n+1} &= (1-\omega)Dx_n + \omega(b + Ux_n) \\ \implies x_{n+1} &= (D-\omega L)^{-1}\left[(1-\omega)Dx_n + \omega(b + Ux_n)\right]. \end{aligned}$$

This converges to a solution, as:

$$\begin{aligned} x_* &= (D-\omega L)^{-1}\left[(1-\omega)Dx_* + \omega(b + Ux_*)\right] \\ Dx_* - \omega L_* x_* &= (D - \omega D)x_* + \omega b + \omega U x_* \\ \implies \omega(D - L - U)x_* &= \omega b \\ \implies Ax_* &= b. \end{aligned}$$

Remark 16.9 — If $\omega = 1$, this would reduce to Gauss-Seidel.

We need to study two issues:

1. convergence, to see if it will return a solution
2. rate of convergence, as if it requires more than $O(N)$ iterations, then it is worse than Gaussian elimination.

Note that these iterative methods all have the following form:

$$x_{n+1} = Gx_n + C.$$

Example 16.10

For the above methods, we have:

1. Jacobi Iteration: $G = D^{-1}(L+U)$, $C = D^{-1}b$
2. Gauss-Seidel: $G = (D-L)^{-1}U$ or $G = (D-U)^{-1}L$
3. SOR: $G = (D-\omega L)^{-1}((1-\omega)D + \omega U)$

We know that the exact solution satisfies:

$$x^* = Gx^* + C.$$

and the iterative solution satisfies:

$$x_{n+1} = Gx_n + C.$$

Subtracting, we have:

$$x_{n+1} - x^* = G(x_n - x^*).$$

Denoting the difference as the error $e_n = x_n - x^*$, we have:

$$e_{n+1} = Ge_n = G^2 e_{n+1} = \ldots = G^{n+1}e_0.$$

Where $e_0 = x_0 - x^*$. As such, we want to see if $x_{n+1} = G^{n+1}e_0$ tends to zero.

Let us consider the simple case where G is diagonalizable, i.d. $G = T^{-1}\Lambda T$, where $\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}$, we have:

$$G^{n+1} = T^{-1}\Lambda^{n+1}T, \quad \Lambda^{n+1} = \begin{bmatrix} \lambda_1^{n+1} & & \\ & \ddots & \\ & & \lambda_m^{n+1} \end{bmatrix} \to 0.$$

Lemma 16.11

The iteration converges as $n \to +\infty$ for all starting vector $x_0 \iff \rho(G) < 1$, where:

$$\rho(G) = \max_i |\lambda_i(G)|,$$

or in other words the norm of all eigenvalues $|\lambda_i| < 1$.

Definition 16.12. $\rho(G)$ is the **spectral radius** of G.

Remark 16.13 — For the diagonalization of G, Λ is a diagonal matrix of all of the eigenvalues, and T is all of the eigenvectors.

Returning to the three iterations, let's study the convergence using Lemma 16.11.

Theorem 16.14
The Jacobi iteration converges for an irreducibly diagonally dominant matrix.

Definition 16.15. $A = (A_{ij})$ is **diagonally dominant** if:

$$|A_{\ell\ell}| \geq \sum_{m \neq \ell} |A_{\ell m}|, \qquad \forall \ell.$$

or in other words, the norm of the diagonal entry is larger than the sum of the off diagonal entries in the row.

Definition 16.16. $A = (A_{ij})$ is **strictly diagonally dominant** if:

$$|A_{\ell\ell}| > \sum_{m \neq \ell} |A_{\ell m}|, \qquad \forall \ell.$$

Definition 16.17. $A = (A_{ij})$ is **irreducibly diagonally dominant** if A is diagonally dominant and the strict inequality holds for at least one row ℓ.

Example 16.18
Consider the identity matrix I. Note that it is strictly diagonally dominant for all rows. Thus it is irreducibly diagonally dominant.

Example 16.19
Consider the central difference matrix for $-u'' = f$:

$$A = -\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 \end{bmatrix}.$$

Note that $|A_{\ell\ell}| = 2$, where as $\sum_{m \neq \ell} |A_{\ell m}| = 2$. for all $\ell \in \{2, \ldots, n-1\}$ However for $\ell = 1$ or n, we would the strict inequality. Thus it is irreducibly diagonally dominant.

17 November 12th, 2020

17.1 Iterative Methods to Solve Linear Systems

Last time, we introduced three methods, Jacobi, Gauss-Seidel, and SOR. Let us consider the Jacobi iteration. We want to show that it is irreducibly diagonally dominant matrix.

Proof. Assume A is irreducibly diagonally dominant. Recall that the Jacobi iteration is of the form

$$G = D^{-1}(L + U).$$

We want to show that $\rho(G) < 1$. Assume λ is an eigenvalue of G. We need to now show that $|\lambda| < 1$. We have:

$$\begin{aligned} Gv &= \lambda v \\ \implies D^{-1}(L+U)v &= \lambda v \\ \implies (L+U)v &= \lambda D v \\ \implies (\lambda D - L - U)v &= 0. \end{aligned}$$

Writing down each row explicitly, we have:

$$\lambda A_{kk} v_k + \sum_{\ell \neq k} A_{k\ell} v_\ell = 0 \quad \forall k.$$

$$\lambda A_{kk} v_k = -\sum_{\ell \neq k} A_{k\ell} v_\ell \quad \forall k.$$

Taking absolute value on both sides, we have:

$$|\lambda||A_{kk}||v_k| = |\sum_{\ell \neq k} A_{k\ell} v_\ell| \leq \sum_{\ell \neq k} |A_{k\ell}||v_\ell|.$$

If we choose $|v_k| = \max_i |v_i| \neq 0$, we would have:

$$|\lambda| \leq \sum_{\ell \neq k} \frac{|A_{k\ell}|}{|A_{kk}|} \frac{|v_\ell|}{|v_k|} \leq \frac{1}{|A_{kk}|} \sum_{\ell \neq k} |A_{k\ell}| \leq 1.$$

The last inequality is because we assume that A is diagonally dominant.

If $|\lambda| = 1$, we would have:

$$1 \leq \sum_{\ell \neq k} \frac{|A_{k\ell}|}{|A_{kk}|} \frac{|v_\ell|}{|v_k|} \leq 1 \implies |v_\ell| = |v_k|, \forall \ell.$$

Now for all k, we have:

$$\sum_{\ell \neq k} |A_{k\ell}| = |A_{kk}|.$$

which is a contradiction to the irreducibility of A. As such $|\lambda| < 1$ for all eigenvalues of A, meaning that the Jacobi iteration converges. □

As shown last time, if we consider the central difference for $-u'' = f$, then the matrix is irreducibly diagonally dominant. We can compute G, with:

$$G = D^{-1}(L+U) = \frac{1}{2} \begin{bmatrix} 0 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 0 \end{bmatrix}_{(N-1)\times(N-1)}.$$

In addition, we can directly compute the eigenvalues of the original differential operator, as we have:

$$-u'' = \lambda u, \quad u(0) = u(1) = 0.$$

Since the general solution to $-u'' = \lambda u$ is:

$$u = \begin{cases} A\sin(\sqrt{\lambda}x) + B\cos(\sqrt{\lambda}x) & \lambda > 0 \\ Ae^{\sqrt{-\lambda}x} + Be^{-\sqrt{-\lambda}x} & \lambda < 0 \end{cases}.$$

Note that the second case does not satisfy the boundary condition, as:

$$u(0) = A + B = 0 \quad u(1) = Ae^{\sqrt{-\lambda}} + Be^{-\sqrt{-\lambda}} = 0.$$

$$\implies Ae^{\sqrt{-\lambda}}(1 - e^{-2\sqrt{-\lambda}}) = 0.$$

which is untrue, since $e^{-2\sqrt{-\lambda}} < 1$. As such, $\lambda > 0$. Considering the boundary conditions, we have:

$$u(0) = B = 0.$$

$$u(1) = A\sin(\sqrt{\lambda}) = 0 \implies \lambda_n = (n\pi)^2.$$

since $A \neq 0$, otherwise we would get the trivial solution. The eigenvectors are thus:

$$u_n(x) = \sin(n\pi x).$$

Now consider the eigenvalues and eigenvectors of G, which are discretized. We predict that the eigenvectors are of form:

$$u_n = \sin(n\pi x_j) = \begin{bmatrix} \sin(n\pi x_1) \\ \sin(n\pi x_2) \\ \vdots \\ \sin(n\pi x_{N-1}) \end{bmatrix}.$$

where $x_j = jh = \frac{j}{N}$. We want to show that:

$$Gu_n = \lambda_n u_n.$$

Consider the j-th row of Gu_n, we have:

$$\frac{1}{2}\begin{bmatrix} 0 & 0 & \dots & 1 & 0 & 1 & 0 \dots \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_j \\ u_{j+1} \\ \vdots \\ u_{N-1} \end{bmatrix} = \frac{1}{2}(u_{j-1} + u_{j+1}) = \lambda_n u_j.$$

Thus, we have:

$$\frac{1}{2}(\sin(n\pi x_{j-1}) + \sin(n\pi x_{j+1})) = \lambda_n \sin(n\pi x_j).$$

$$\implies \frac{1}{2}(\sin(n\pi(j-1)h) + \sin(n\pi(j+1)h)) = \lambda_n \sin(n\pi jh).$$

Recalling that $\sin(x+y) = \sin x \cos y + \cos x \sin y$, giving us:

$$(\sin(n\pi jh)\cos(n\pi h) - \cos(n\pi jh)\sin(n\pi h)) + (\sin(n\pi jh)\cos(n\pi h) + \cos(n\pi jh)\sin(n\pi h)).$$

$$\implies \cos(n\pi h)\sin(n\pi jh) = \lambda_n \sin(n\pi jh).$$

This means that:

$$\lambda_n = \cos(n\pi h).$$

with eigen vectors:

$$u_n(x) = \sin(n\pi jh).$$

Thus, we have:

$$|\lambda_n| = |\cos(n\pi h)| < 1, \quad n = 1, 2, \ldots, N-1.$$

We now want to investigate the rate of iteration, which indicates the number of iterations required to obtain a required accuracy.

Definition 17.1. The **rate of convergence** is

$$Rln - \ln(\rho).$$

Example 17.2
To reduce the error by a factor of 10^p, the number of iterations required is $\frac{P}{R}\ln(10)$.

Let's assume that $G = S\Lambda S^{-1}$, i.e. G is diagonalizable. Recall, we have:

$$x_{n+1} = Gx_n + C.$$

$$x_* = Gx_* + C.$$

We have:

$$\begin{aligned} x_{n+1} - x_* &= G(x_n - x_*) \\ \implies e_{n+1} &= Ge_n = G^2 e_{n-1} = \ldots = G^{n+1} e_0. \end{aligned}$$

This gives us:

$$\|e_n\| \le \|G^n\|\|e_0\| \le \|S\Lambda^n S^{-1}\|\|e_0\| \le \|S\|\|S^{-1}\|\|\Lambda^n\|\|e_0\|.$$

As such:

$$\frac{\|e_n\|}{\|e_0\|} \le \|S\|\|S^{-1}\|\|\Lambda\|^n \le \|S\|\|S^{-1}\|\rho^n.$$

Say we want to reduce the error by a factor of 10^p, we have:

$$\begin{aligned} &\|S\|\|S^{-1}\|\rho^n \le 10^{-p} \\ \implies &(\|S\|\|S^{-1}\|)^{\frac{1}{n}}\rho \le 10^{-\frac{p}{n}} \\ \implies &\frac{1}{n}\ln(\|S\|\|S^{-1}\|) + \ln\rho \le -\frac{p}{n}\ln 10 \\ \implies &\frac{\ln(\|S\|\|S^{-1}\|)}{\ln\rho} + n \ge -\frac{p}{\ln\rho}\ln 10 \\ \implies &n \ge \frac{P}{R}\ln 10 - C. \end{aligned}$$

where $R = -\ln\rho$ for some constant C.

Remark 17.3 — If ρ is close to 1, then $n \sim \infty$.

Remark 17.4 — If N is very large, then the first eigenvalue will be close to 1, meaning that ρ is close to 1. This is not good, since for accuracy reason we want N to be large, but convergence would be slow.

18 November 17th, 2020

18.1 Methods to solve PDEs

Last time, we tried to calculate the explicit scheme for some iterative schemes. Let us consider the model:

$$\begin{cases} -u''(x) + \sigma u(x) = f(x) \quad 0 < x < 1, \sigma > 0 \\ u(0) = u(1) = 0 \end{cases}.$$

If we use a uniform grid with:

$$h = \frac{1}{N}, x_i = ih.$$

We can use the Taylor expansion to approximate the second derivative using central different with second order accuracy. With this, we can approximate the boundary value problem with a finite difference scheme, giving us a symmetric positive definite matrix equation that we solve iteratively.

There are a few other methods to solve this:

- Direct Methods
  - Gaussian elimination
  - Factorization
- Iterative Methods
  - Jacobi
  - Gauss-Seidel
  - SOR
  - Conjugate Gradient

If we have a 2D problem, we would have a similar situation with a linear block tridiagonal system.

Coming back to the iterative method, consider $Au = f$ where A is NxN and let v be an approximation of u. There are two important measures:

- The error: $e = u - v$ with norms:

$$\|e\|_\infty = \max |e_i| \quad \|e\|_2 = \sqrt{\sum_{i=1}^{N} e_i^2}.$$

- The residual: $r = f - Av$ with norms:

$$\|r\|_\infty \quad \|r\|_2.$$

We can rewrite $Au = f$ as:

$$A(v + e) = f.$$

which means that

$$Ae = f - Av = f.$$

which is called the **residual equation**. This means that if we know the error we can correct it with $u = v + e$. This is called **residual correction**.

18.2 Relaxation Schemes

The Jacobi and Gauss-Seidel iterations are both relaxation schemes.

Let us consider the weighted Jacobi relaxation:

$$v_i^{n+1} = (1 - \omega)v_i^n + \frac{\omega}{2}(v_{i-1}^n + v_{i+1}^n + h^2 f_i), \quad 0 < \omega < 2.$$

Remark 18.1 — When $\omega = 1$, this is the Jacobi iteration.

There a few other version as well

19 November 24th, 2020

19.1 Spectral Methods

Spectral methods are based on Fourier series and have high accuracy. If we are the domain $[0, 2\pi]$, we can define the periodic functions:

$$\phi_k(x) = e^{ikx} = \cos kx + i \sin kx.$$

For any function $u(x) \in L^2[0, 2\pi$, we can express it in terms of its Fourier series:

$$u(x) = \sum_{k=-\infty}^{+\infty} u_k e^{ikx}.$$

Where:

$$u_k = \frac{1}{2\pi} \int_0^{2\pi} u(x) e^{-ikx} dx.$$

u_k are called the **fourier coefficients**.

Remark 19.1 — L^2 function means it is square integrable.

$\phi_k(x)$ are orthogonal as:

$$\int_0^{2\pi} \phi_k \overline{\phi_\ell} dx = \int_0^{2\pi} e^{ikx} e^{-i\ell x} dx = \int_0^{2\pi} e^{i(k-\ell)x} dx = \begin{cases} 2\pi & k = \ell \\ 0 & k \neq \ell \end{cases}.$$

$\phi_k(x)$ are complete, meaning for any $u \in L^2[0, 2\pi]$ it can be approximated by:

$$u(x) \sim \sum_{k=-N}^{N} u_k \phi_k(x.$$

in the sense that:

$$\lim_{N\to\infty} \|u(x) - \sum_{k=-N}^{N} u_k \phi_k(x)\| = 0.$$

As such $\{\phi_k\}$ forms a basis of $L^2[0, 2\pi]$.

Definition 19.2 (Spectral Projection)**.** Given a function $u(x) \in L^2[0, 2\pi]$, we define:

$$P_N u(x) = \sum_{k=-N}^{N} u_k e^{ikx}$$

to be the **spectral projection** of u.

> **Remark 19.3 —** In other words, the spectral projection is the finite truncation of the Fourier series.

This spectral projection has a few properties:

- P_N is a projection, i.e. $P_N^2 = P_N$.

  Proof. Let $\tilde{u}(x) = P_N u(x)$ We have:

  $$P_N^2 u = P_N \tilde{u}.$$

  with .

  $$\begin{aligned} \tilde{u_\ell}(x) &= \frac{1}{2\pi} \int_0^{2\pi} \tilde{u}(x) e^{-i\ell x} dx \\ &= \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_{k=-N}^{N} u_k e^{ikx} \right) e^{-i\ell x} dx \\ &= \frac{1}{2\pi} \sum_{k=-N}^{N} u_k \int_0^{2\pi} e^{ikx} e^{-i\ell x} dx \\ &= u_\ell. \end{aligned}$$

  by using the orthogonality of ϕ_k. □

- $\|P_N u(x)\|_{L^2}^2 = 2\pi \sum |u_k|^2$.

Proof. We have:

$$\begin{aligned}\|P_N u(x)\|_{L^2}^2 &= \int_0^{2\pi} |P_N u|^2 dx \\ &= \int_0^{2\pi} \left(\sum_{k=-N}^{N} u_k e^{ikx}\right)\left(\sum_{\ell=-N}^{N} \overline{u_\ell} e^{i\ell x}\right) dx \\ &= \int_0^{2\pi} \sum_{k=-N}^{N}\sum_{\ell=-N}^{N} u_k \overline{u_\ell} e^{i(k-\ell)x} dx \\ &= \sum_k \sum_\ell u_k \overline{u_\ell} \int_0^{2\pi} e^{i(k-\ell)x} dx \\ &= 2\pi \sum_k |u_k|^2.\end{aligned}$$

again by using the orthogonality of ϕ_k. □

19.2 Decay Rate of Spectral Methods

Let us consider the decay rates of the coefficients:

$$u_k = \frac{1}{2\pi}\int_0^{2\pi} u(x) e^{-ikx} dx.$$

If u is differentiable, then do integration by parts giving us:

$$u_k = -\frac{1}{2\pi}\frac{1}{ik} u(x) e^{-ikx}\Big|_0^{2\pi} + \frac{1}{ik}\frac{1}{2\pi}\int_0^{2\pi} u'(x) e^{-ikx} dx.$$

Because u is periodic, the first term is 0, meaning that:

$$u_k = \frac{1}{ik}\frac{1}{2\pi}\int_0^{2\pi} u'(x) e^{-ikx} dx.$$

Note that this is $\frac{1}{ik}(u')_k$. This can be continued, giving us:

$$u_k = \frac{1}{(ik)^r}\frac{1}{2\pi}\int_0^{2\pi} u^{(r)} e^{-ikx} dx.$$

Taking the absolute value, we have:

$$|u_k| \le \frac{1}{|k|^r}\frac{1}{2\pi}\int_0^{2\pi} |u^{(r)}| dx \le \frac{C}{|k|^r}\|u^{(r)}\|_{L^2}.$$

Note that this goes to zero, as k goes to infinity. Note that since r is arbitrary, this goes to zero faster than any power, as we can just increase r. As such, it goes to zero exponentially fast. Of course, this requires that the function is differentiable to any order.

We will now prove this exponential convergence of Fourier series for smooth functions.

Proof. Let $u(x)$ be smooth, i.e. differentiable to any order, then:

$$u(x) = \sum_{k=-\infty}^{+\infty} u_k e^{ikx}, \quad u_k = \frac{1}{2\pi}\int_0^{2\pi} u(x)e^{-ikx}dx.$$

Note that $u(x)$ is absolutely convergent, as u_k decays faster than any power. As such, we have:

$$|u(x)| \leq \sum_k |u_k|.$$

If $u_k \leq \frac{1}{|k|^2}$, then the right hand side is convergent. Now let's consider:

$$\|u(x) - \sum_{k=-N}^{N} u_k e^{ikx}\|_\infty.$$

which converges to 0 as $N \to +\infty$. We have:

$$\begin{aligned}
\|u(x) - \sum_{k=-N}^{N} u_k e^{ikx}\|_\infty &= \|\sum_{|k|>N} u_k e^{ikx}\|_\infty \\
&\leq \sum_{|K|>N} |u_k||e^{ikx}| \\
&= \sum_{|k|>N} |u_k| \\
&\leq \sum_{|k|>N} \frac{1}{|k|^r}\|u^{(r)}\| \\
&\leq C\sum_{|k|>N} \frac{1}{|k|^r} \\
&= \frac{C}{(N+1)^r}\left(\sum_{j=1}\left(\frac{N+1}{N+j}\right)^r\right) \\
&\leq \frac{C}{(N+1)^r}\cdot C_1.
\end{aligned}$$

This means that:

$$\|u(x) - P_N u(x)\| \leq \frac{M}{(N+1)^r} \to 0.$$

for some constant M, which converges faster than any power. As such, it has exponential convergence. □

Remark 19.4 — How fast the function decays in the frequency space depends on how smooth it is in the physical space.

Definition 19.5 (Total Variation)**.** Let us define $u(x)$ on $[0, 2\pi]$ with a mesh $0 = x_0 \leq x_1 \leq x_2 \leq \ldots \leq x_n \leq x_{n+1} = 2\pi$. We define the **total variation** as:

$$\gamma(u) = \sup_n \sup_{\{x_i\}} \sum_i |u(x_i) - u(x_{i-1})|.$$

Example 19.6
If $u(x)$ is a monotone increasing function on $[0, 2\pi]$, then we have $\gamma(u) = u(2\pi) - u(0)$.

Remark 19.7 — If $u(x)$ is not monotone $[0, 2\pi]$, then if we can split the region into monotone sections, we can calculate the total variation as the absolute value of all the difference in the monotone sections.

Example 19.8
Consider $u(x) = \sin x$. We can split it into 3 sections, with the total variation being $\gamma(u) = 4$.

Example 19.9
If $u(x) = \sin \frac{1}{x}$, then $u(x)$ has no bounded variation.

Bounded variation is a very important property to determine the convergence of its Fourier series. It has the following properites:

- If u is continuous, periodic, and has bounded variation, then we have: $P_N u(x) \to u(x)$ uniformly.
- If $u(x)$ has bounded variation, then $P_N u(x) \to \frac{1}{2}(u^+(x) + u^-(x))$, which is the average of its left and right limits (note that it does not need to be continuous), this is pointwise convergence.
- If $u \in L^2[0, 2\pi]$, then:

$$\|P_N u(x) - u(x)\|_{L^2} = \int_0^{2\pi} |P_N u(x) - u(x)|^2 dx \to 0.$$

  This means it converges in the average sense.
- If $u(x)$ is continuous and periodic, but not of bounded variation, then $P_N u(x)$ does not necessarily converge for all x, e.g. $u(x) = x \sin \frac{1}{x}, x \in [-\frac{1}{\pi}, \frac{1}{\pi}]$. At $x = 0$, $P_N u(x)$ is not convergent.

Remark 19.10 — Uniform convergence means that:

$$\lim_{N\to\infty} \max_{x\in[0,2\pi]} |u(x) - P_N u(x)| = 0.$$

Pointwise convergences means that for any $x \in [0, 2\pi]$:

$$\lim_{N\to\infty} |u(x) - P_N u(x)| = 0.$$

20 November 26th, 2020

20.1 Spectral Method Continued

Example 20.1

Consider $u(x) = 1$ $x \in [0, 2\pi]$. The corresponding Fourier coefficients are given by:

$$u_k = \frac{1}{2\pi}\int_0^{2\pi} u(x)e^{ikx}dx = \frac{1}{2\pi}\int_0^{2\pi} e^{ikx}dx = \begin{cases} 1 & k = 0 \\ \frac{1}{2\pi}\frac{1}{ik}e^{ikx}\Big|_0^{2\pi} = 0 & k \neq 0 \end{cases}.$$

Example 20.2

Consider $u(x) = \begin{cases} x & 0 \leq x < 2\pi \\ 0 & x = 2\pi \end{cases}$. Note that this is periodic. Computing the Fourier series, we have:

$$u_k = \frac{1}{2\pi}\int_0^{2\pi} xe^{ikx}dx = \frac{1}{2\pi}\frac{1}{ik}e^{ikx}\Big|_0^{2\pi} - \frac{1}{2\pi}\frac{1}{ik}\int_0^{2\pi} e^{ikx}dx.$$

$$= \begin{cases} \frac{1}{ik}(-1)^{k+1} & k \neq 0 \\ \frac{1}{2\pi}\int_0^{2\pi} xdx = \frac{1}{2\pi}2\pi = \pi & k = 0 \end{cases}.$$

Thus:

$$u(x) = \pi + \sum_{k\neq 0} \frac{1}{ik}(-1)^k e^{ikx}.$$

Let us consider $\begin{cases} -u'' = f \\ u(0) = u(2\pi) \end{cases}$ meaning u is periodic. We have:

$$u(x) = \sum_{k=-\infty}^{+\infty} u_k e^{ikx}.$$

Taking derivative, we have:

$$u' = \sum_{k=-\infty}^{+\infty} u_k(ik)e^{ikx} \quad u'' = \sum_{k=-\infty}^{+\infty} u_k(ik)^2 e^{ikx} = -\sum_{k=-\infty}^{+\infty} u_k k^2 e^{ikx}.$$

Let us also consider the Fourier expansion of f:

$$f = \sum_{k=-\infty}^{+\infty} f_k e^{ikx} \quad f_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}dx.$$

Plugging this in, we have:

$$-u'' = f \implies \sum_{k=-\infty}^{+\infty} k^2 u_k e^{ikx} = \sum_{k=-\infty}^{+\infty} f_k e^{ikx}.$$

Since e^{ikx} forms a base, we have:

$$k^2 u_k = f_k \quad \forall k.$$

and if $k \neq 0$:

$$u_k = \frac{1}{k^2} f_k.$$

If $k = 0$, then $f_0 = 0$, meaning u_0 is arbitrary. This is reasonable, since we can add any constant to a solution to get another solution. For simplicity, we can set $u_0 = 0$. As such, we have:

$$u(x) = \sum_{k \neq 0} \left(\frac{1}{k^2} f_k \right) e^{ikx} = \sum_{k=-\infty}^{+\infty} u_k e^{ikx}.$$

Truncating this infinite series, we have:

$$u_N = \sum_{k=-N}^{N} \hat{u_k} e^{ikx}.$$

where:

$$\hat{u_k} = \frac{1}{2\pi} \frac{1}{k^2} \int_0^{2\pi} f(x) e^{ikx} dx.$$

This can be computed numerically.

One way to do this is to use Trapezoidal rule, with:

$$\int_0^{2\pi} g(x) dx = \sum_{i=0}^{N-1} \frac{1}{2} \left(g(x_i) + g(x_{i+1}) \right) h.$$

where $h = \frac{2\pi}{N}$ being the width between grid points. Since $g(x_0) = g(x_N)$, this would give us:

$$\sum_{n=0}^{N-1} g(x_i) h.$$

since each point is counted twice and $g(x_0) = g(x_N)$.

Applying this, we have:

$$\hat{f}_k = \frac{1}{2\pi} \sum_{j=0}^{N-1} f(jh) e^{-ik(jh)} h = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{\frac{-ikj\pi}{N}}.$$

This is the definition of the **discrete Fourier transform (DFT)**. The computational cost for $\hat{f}_k$ is $O(N^2)$. Which is much higher than the $O(N)$ for finite difference (for this case) since it is tridiagonal. As such, even though it gives good accuracy, it was not used early on since the direct method is computationally expensive. This is until FFT was discovered, which reduces it to $O(N \log N)$. This makes the spectral method very popular and affordable.

20.2 Fast Fourier Transform

FFT assums that $N = 2^p$. Consider f_j for $j \in \{1, 2, \ldots, 2N-1\}$ since $f_{2N} = f_1$. Then the Fourier coefficients:

$$\hat{f}_k \frac{1}{2N} \sum_{j=0}^{2N-1} f_j e^{-ikx_j}.$$

with $\hat{f_{2N}} = \hat{f}_0$ Separating the indexes into even and odd indices:

$$f_1(k) = f_{2k}, \quad k = 0, 1, \ldots, N-1.$$

$$f_2(k) = f_{2k+1}, \quad k = 0, 1, \ldots, N-1.$$

With this, we can calculate:

$$\hat{f}_1(k) = \frac{1}{N} \sum_{n=0}^{N-1} f_1(j) e^{-ik(j\frac{2\pi}{N})}.$$

$$\hat{f}_2(k) = \frac{1}{N} \sum_{n=0}^{N-1} f_2(j) e^{-ik(j\frac{2\pi}{N})}.$$