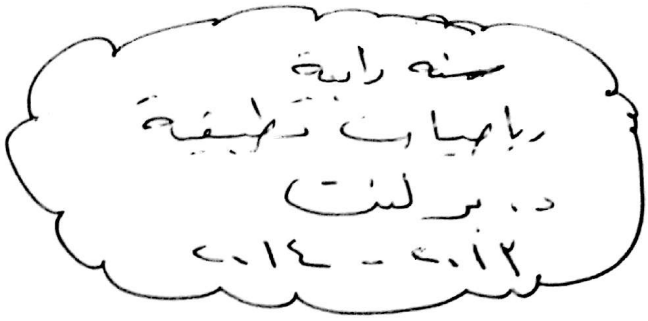

Chapter 1

Introduction

The goal of this course is to provide numerical analysis background for finite difference approximation to solve partial differential equations. The partial differential equations include

- parabolic equations,
- elliptic equations,
- hyperbolic conservation laws.

I will mainly talk about stability and convergence theory.

1.1 Finite Difference Approximation

Our goal is to appriximate differential operqtors by finite difference operators. How to perform approximation? What is the error so produced? In general, we shall assume the underlying functions are smooth. But we should notice that in some class of problems, the underlying functions may not be smooth. Nevertheless, let us limit ourselves to those smooth functions at this moment.

Assumming the underlying function $u : \mathbb{R} \to \mathbb{R}$ is smooth. Let us define the following finite difference operators:

- Forward difference: $D_+u(x) := \frac{u(x+h)-u(x)}{h}$
- Backward difference: $D_-u(x) := \frac{u(x)-u(x-h)}{h}$
- Centered difference: $D_0u(x) := \frac{u(x+h)-u(x-h)}{2h}$

By Taylor expansion, we can get

- $u'(x) = D_+u(x) + O(h)$,
- $u'(x) = D_-u(x) + O(h)$,
- $u'(x) = D_0u(x) + O(h^2)$,

We can also approximate $u'(x)$ with higher order error. For example,

$$u'(x) = D_3 u(x) + O(h^3)$$

where

$$D_3 u(x) = \frac{1}{6h}\left(2u(x+h) + 3u(x) - 6u(x-h) + u(x-2h)\right)$$

These formulae can be derived from performing Taylor expansion of u at x. For instance, we expand

$$\begin{aligned} u(x+h) &= u(x) + u'(x)h + \frac{h^2}{2}u''(x) + \frac{h^3}{3!}u'''(x) + \cdots \\ u(x-h) &= u(x) - u'(x)h + \frac{h^2}{2}u''(x) - \frac{h^3}{3!}u'''(x) + \cdots \end{aligned}$$

Substracting these two equations yield

$$u(x+h) - u(x-h) = 2u'(x)h + \frac{2h^3}{3!}u'''(x) + \cdots.$$

This gives

$$u'(x) = D_0 u(x) - \frac{h^2}{3!}u'''(x) + \cdots = D_0 u(x) + O(h^2).$$

In general, we can derive finite difference approximation for $u^{(k)}$ at x by the values of u at stencil points $x_j, j = 0, ...n$ with $n \geq k$. That is,

$$u^{(k)}(x) = \sum_{j=0}^{n} c_j u(x_j) + O(h^{p+1})$$

for some p as larger as possible. As we shall see that we can choose $p = n$. To find the coefficients $c_j, j = 0, ..., n$, we expand

$$u(x_j) = \sum_{i=0}^{p} \frac{1}{i!}(x_j - x)^i u^{(i)}(x) + O(h^{p+1}).$$

Thus,

$$u^{(k)}(x) = \sum_{j=0}^{n} c_j \sum_{i=0}^{p} \frac{1}{i!}(x_j - x)^i u^{(i)}(x) + O(h^{p+1}).$$

Comparing both sides, we obtain

$$\sum_{j=0}^{n} \frac{1}{i!}(x_j - x)^i c_j = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 0, ..., p$$

There are $p+1$ equations here. it is natural to choose $p = n$ to match the $n+1$ unknowns. This is a $n \times n$ Vandermonde system. It is nonsingular if x_i are different. The matlab code fdcoeffV(k,xbar,x) can be used to compute these coefficients.

Homeworks.

1. Consider $x_i = ih$, $i = 0, ..., n$. Let $\bar{x} = x_m$. Find the coefficients c_i for $u^{(k)}(\bar{x})$ and the coefficient of the leading truncation error for the following cases:
   - $k = 1, n = 2, 3, m = 0, 1, 2, 3.$
   - $k = 2, n = 2, m = 0, 1, 2.$

1.2 Basic Numerical Methods for Ordinary Differential Equations

The basic methods to design numerical algorithm is based on the smoothness of the solution. Techniques of numerical interpolation, numerical integration, or finite difference approximation are adopted.

Euler method

Euler method is the simplest numerical integrator for ODEs. The ODE

$$y' = f(t, y) \tag{1.1}$$

is discretized by

$$y^{n+1} = y^n + kf(t^n, y^n). \tag{1.2}$$

Here, k is time step size of the discretization. This is called the forward Euler method. It simply $dy/dt(t^n)$ replaced by forward finite difference $(y^{n+1} - y^n)/k$. To measure the error, the local truncation error is

$$\tau^n := y'(t^n) - \frac{y(t^{n+1}) - y(t^n)}{k} = O(k)$$

Let $e^n := y^n - y(t^n)$ be the true error.

Theorem 2.1 *Assuming $f \in C^1$ and the solution $y' = f(t, y)$ with $y(0) = y_0$ exists on $[0, T]$. Then the Euler method converges at any $t \in [0, T]$. In fact, the true error e^n has the following estimate:*

$$\|e^n\| \leq \frac{e^{\lambda t}}{\lambda} O(k) \to 0, \text{ as } n \to \infty. \tag{1.3}$$

Here, $\lambda = \max \partial f/\partial y$ and $nk = t$.

Proof. From the regularity of the solution, $y \in C^2[0, T]$ and

$$y(t^{n+1}) = y(t^n) + kf(t^n, y(t^n)) + k\tau^n. \tag{1.4}$$

Taking difference of (1.2) and (1.4), we obtain

$$\begin{aligned} \|e^{n+1}\| &\leq \|e^n\| + k|f(t^n, y^n) - f(t^n, y(t^n))| + k|\tau^n| \\ &\leq (1 + k\lambda)\|e^n\| + k|\tau^n|. \end{aligned}$$

where

$$\lambda = \max_{x,y} |f(t,x) - f(t,y)|/|x-y|$$

The finite difference inequality has a fundamental solution $G^n = (1+\lambda k)^n$, which is positive provided k is small. Multiplying above equation by $(1+\lambda k)^{-n-1}$, we obtain

$$e^{m+1}G^{-m-1} \le e^m G^{-m} + kG^{-m-1}|\tau^m|.$$

Summing in m from $m \equiv 0$ to $n-1$, we get

$$\begin{aligned} e^n &\le \sum_{m=0}^{n-1} G^{n-m-1} k|\tau^m| \\ &\le \sum_{m=0}^{n-1} G^m O(k^2) \\ &= \frac{G^n - 1}{G - 1} O(k^2) \\ &\le \frac{G^n}{\lambda} O(k) \\ &\le \frac{e^{\lambda t}}{\lambda} O(k) \end{aligned}$$

where $t = nk$ and we have used $(1+\lambda k)^n \le e^{\lambda t}$. ∎

Remarks.

1. The theorem says that the numerical method converges in $[0,T]$ as long as the solutions of the ODE exists.
2. One can also prove the existence of the ODE solution through Euler method. It will be a local existence theorem.

Backward Euler method

In many applications, the system is relaxed to a stable solution in a very short time. For instance, consider

$$y' = \frac{\bar{y} - y}{\tau}.$$

The corresponding solution $y(t) \to \bar{y}$ as $t \sim O(\tau)$. In the above forward Euler method, practically, we should require

$$1 + k\lambda \le 1$$

in order to have G^n remain bounded. Here, λ is the Lipschitz constant. In the present case, $\lambda = 1/\tau$. If τ is very small, the the above forward Euler method will require very small k and lead to inefficient computation. In general, forward Euler method is inefficient (require small k) if

$$\max \left| \frac{\partial f(t,y)}{\partial y} \right| >> 1.$$

In the case $\partial f/\partial y >> 1$, we have no choice to resolve details. We have to take a very small k. However, if $\partial f/\partial y < 0$, say for example, $y' = -\lambda y$ with $\lambda >> 1$. then the backward Euler method is recommended.

$$y^{n+1} = y^n + kf(t^{n+1}, y^{n+1}).$$

The error satisfies

$$e^{n+1} \leq e^n - \lambda k e^{n+1} + O(k^2)$$

The corresponding fundamental solution is $G^n := (1+\lambda k)^{-n}$. Notice that the error satisfies

$$\begin{aligned} e^n &\leq \sum_{m=0}^{n-1}(1+\lambda k)^{-m}O(k^2) \\ &\leq \frac{(1+\lambda k)^{-n+1}}{\lambda k}O(k^2) \\ &\leq \frac{e^{-\lambda T}}{\lambda}O(k) \end{aligned}$$

There is no restriction on the size of k.

Leap frog method

We integrate $y' = f(t, y)$ from t^{n-1} to t^{n+1}:

$$y(t^{n+1}) - y(t^{n-1}) = \int_{t^{n-1}}^{t^{n+1}} f(\tau, y(\tau))\, d\tau$$

We apply the midpoint rule for numerical integration, we then get

$$y(t^{n+1}) - y(t^{n-1}) = 2kf(t^n, y(t^n)) + O(k^3).$$

The midpoint method (or called leapfrog method) is

$$y^{n+1} - y^{n-1} = 2kf(t^n, y^n). \tag{1.5}$$

This is a two-step explicit method.

Homeworks.

1. Prove the convergence theorem for the backward Euler method.
   Hint: show that $e^{n+1} \leq e^n + (1+k\lambda)e^{n+1} + k\tau^n$, where λ is the Lipschitz constant of f.

2. Prove the convergence theorem for the leap-frog method.
   Hint: consider the system $y_1^n = y^{n-1}$ and $y_2^n = y^n$.

1.3 Runge-Kutta methods

The Runge-Kutta method (RK) is a strategy to integrate $\int_{t^n}^{t^{n+1}} f\,d\tau$ by some quadrature method. For instance, a second order RK, denoted by RK2, is based on the trapezoidal rule of numerical integration. The integration $\int_{t^n}^{t^{n+1}} f(\tau, y(\tau))$ is approximated by $1/2(f(t^n, y^n)+ f(t^n, y^{n+1}))k$. The latter term involves y^{n+1}. An explicit Runge-Kutta method approximate y^{n+1} by $y^n + kf(t^n, y^n)$. Thus, RK2 reads

$$\begin{aligned} \xi_1 &= f(t^n, y^n) \\ y^{n+1} &= y^n + \frac{k}{2}(f(t^n, y^n) + f(t^{n+1}, y^n + k\xi_1)) \end{aligned}$$

Another kind of RK2 is based on the midpoint rule of integration. It reads

$$\begin{aligned} \xi_1 &= f(t^n, y^n) \\ y^{n+1} &= y^n + kf(t^{n+1/2}, y^n + \frac{k}{2}\xi_1) \end{aligned}$$

1.4 Linear difference equation

Second-order linear difference equation. In the linear case $y' = \lambda y$, the above difference scheme results in a linear difference equation. Let us consider general second order linear difference equation with constant coefficients:

$$ay^{n+1} + by^n + cy^{n-1} = 0, \tag{1.6}$$

where $a \neq 0$. To find its general solutions, we try the ansatz $y^n = \rho^n$ for some number ρ. Here, the n in y^n is an index, whereas the n in ρ^n is a power. Plug this ansatz into the equation, we get

$$a\rho^{n+1} + b\rho^n + c\rho^{n-1} = 0.$$

This leads to

$$a\rho^2 + b\rho + c = 0.$$

There are two solutions ρ_1 and ρ_2. In case $\rho_1 \neq \rho_2$, these two solutions are independent. Since the equation is linear, any linear combination of these two solutions is again a solution. Moreover, the general solution can only depend on two free parameters, namely, once y^0 and y^{-1} are known, then $\{y^n\}_{n\in\mathbb{Z}}$ is uniquely determined. Thus, the general solution is

$$y^n = C_1\rho_1^n + C_2\rho_2^n,$$

where C_1, C_2 are constants. In case of $\rho_1 = \rho_2$, then we can use the two solutions ρ_2^n and ρ_1^n with $\rho_2 \neq \rho_1$, but very closed, to produce another nontrivial solution:

$$\lim_{\rho_2\to\rho_1} \frac{\rho_2^n - \rho_1^n}{\rho_2 - \rho_1}$$

This yields the second solution is $n\rho_1^{n-1}$. Thus, the general solution is

$$C_1\rho_1^n + C_2 n\rho_1^{n-1}.$$

Linear finite difference equation of order r . We consider general linear finite difference equation of order r:

$$a_r y^{n+r} + \cdots + a_0 y^n \quad 0, \tag{1.7}$$

where $a_r \neq 0$. Since y^{n+r} can be solved in terms of $y^{n+r-1}, ..., y^n$ for all n, this equation together with initial data $y_0, ..., y_{-r+1}$ has a unique solution. The solution space is r dimensions.

To find fundamental solutions, we try the ansatz

$$y^n = \rho^n$$

for some number ρ. Plug this ansatz into equation, we get

$$a_r \rho^{n+r} + \cdots + a_0 \rho^n = 0,$$

for all n. This implies

$$a(\rho) := a_r \rho^r + \cdots + a_0 = 0. \tag{1.8}$$

The polynomial $a(\rho)$ is called the characteristic polynomial of (??) and its roots $\rho_1, ..., \rho_r$ are called the characteristic roots.

- Simple roots (i.e. $\rho_i \neq \rho_j$, for all $i \neq j$): The fundamental solutions are ρ_i^n, $i = 1, ..., r$.
- Multiple roots: if ρ_i is a multiple root with multiplicity m_i, then the corresponding independent solutions

$$\rho_i^n, n\rho_i^{n-1}, C_2^n \rho_i^{n-2} ..., C_{m_i-1}^n \rho_i^{n-m_i+1}$$

Here, $C_k^n := n!/(k!(n-k)!)$. The solution $C_2^n \rho_i^{n-2}$ can be derived from differentiation $\frac{d}{d\rho} C_1^n \rho^{n-1}$ at ρ_i.

In the case of simple roots, we can express general solution as

$$y^n = C_1 \rho_1^n + \cdots + C_r \rho_r^n,$$

where the constants $C_1, ..., C_r$ are determined by

$$y^i = C_1 \rho_1^i + \cdots + C_r \rho_r^i, i = 0, ..., -r+1.$$

Syatem of linear difference equation. The above rth order linear difference equation is equivalent to a first order linear difference system:

$$\mathbf{A}_0 \mathbf{y}^{n+1} = \mathbf{A}\mathbf{y}^n \tag{1.9}$$

where

$$\mathbf{y}^n = \begin{pmatrix} y_1^n \\ \vdots \\ y_r^n \end{pmatrix} = \begin{pmatrix} y^{n-r+1} \\ \vdots \\ y^n \end{pmatrix}$$

$$\mathbf{A}_0 = \begin{pmatrix} I_{(r-1)\times(r-1)} & 0 \\ 0 & c_r \end{pmatrix}, \ \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{r-1} \end{pmatrix}.$$

We may divide (1.9) by $\mathbf{A}_0$ and get

$$\mathbf{y}^{n+1} = \mathbf{G}\mathbf{y}^n.$$

We call $\mathbf{G}$ the fundamental matrix of (1.9). For this homogeneous equation, the solution is

$$\mathbf{y}^n = \mathbf{G}^n\mathbf{y}^0$$

Next, we compute $\mathbf{G}^n$ in terms of eigenvalues of $\mathbf{G}$.

In the case that all eigenvalues $\rho_i, i = 1, ..., r$ of $\mathbf{G}$ are distinct, then $\mathbf{G}$ can be expressed as

$$\mathbf{G} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}, \ \mathbf{D} = \text{ diag } (\rho_1, \cdots, \rho_r),$$

and the column vectors of $\mathbf{T}$ are the corresponding eigenvectors.

When the eigenvalues of $\mathbf{G}$ have multiple roots, we can normalize it into Jordan blocks:

$$\mathbf{G} = \mathbf{T}\mathbf{J}\mathbf{T}^{-1}, \ \mathbf{J} = \text{ diag } (\mathbf{J}_1, \cdots, \mathbf{J}_s),$$

where the Jordan block $\mathbf{J}_i$ corresponds to eigenvalue ρ_i with multiplicity m_i:

$$\mathbf{J}_i = \begin{pmatrix} \rho_i & 1 & 0 & \cdots & 0 \\ 0 & \rho_i & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \rho_i \end{pmatrix}_{m_i \times m_i}.$$

and $\sum_{i=1}^{s} m_i = r$. Indeed, this form also covers the case of distinct eigenvalues.

In the stability analysis below, we are concerned with whether $\mathbf{G}^n$ is bounnded. It is easy to see that

$$\mathbf{G}^n = \mathbf{T}\mathbf{J}^n\mathbf{T}^{-1}, \mathbf{J}^n = \text{ diag } (\mathbf{J}_1^n, \cdots, \mathbf{J}_s^n)$$

$$\mathbf{J}_i^n = \begin{pmatrix} \rho_i^n & n\rho_i^{n-1} & C_2^n\rho^{n-2} & \cdots & C_{m_i-1}^n\rho_i^{n-m_i+1} \\ 0 & \rho_i^n & n\rho_i^{n-1} & \cdots & C_{m_i-2}^n\rho_i^{n-m_i+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n\rho_i^{n-1} \\ 0 & 0 & 0 & \cdots & \rho_i^n \end{pmatrix}_{m_i \times m_i}.$$

where $C_k^n := \frac{n!}{k!(n-k)!}$.

Definition 4.1 *The fundamental matrix* $\mathbf{G}$ *is called stable if* $\mathbf{G}^n$ *remains bounded under certain norm* $\|\cdot\|$ *for all* n.

Theorem 4.2 *The fundamental matrix* $\mathbf{G}$ *is stable if and only if its eigenvalue satisfy the following condition:*

$$\begin{aligned} &\textit{either } |\rho| = 1 \textit{ and } \rho \textit{ is a simple root,} \\ &\textit{or } |\rho| < 1 \end{aligned} \tag{1.10}$$

Nonhomogeneous linear finite difference system In general, we consider the nonhomogeneous linear difference system:

$$\mathbf{y}^{n+1} = \mathbf{G}\mathbf{y}^n + \mathbf{f}^n \tag{1.11}$$

with initial data $\mathbf{y}^0$. Its solution can be expressed as

$$\begin{aligned} \mathbf{y}^n &= \mathbf{G}\mathbf{y}^{n-1} + \mathbf{f}^{n-1} \\ &= \mathbf{G}(\mathbf{G}\mathbf{y}^{n-2} + \mathbf{f}^{n-2}) + \mathbf{f}^{n-1} \\ &\vdots \\ &= \mathbf{G}^n\mathbf{y}^0 + \sum_{m=0}^{n-1} \mathbf{G}^{n-1-m}\mathbf{f}^m \end{aligned}$$

Homeworks.

1. Consider the linear ODE
$$y' = \lambda y$$
where λ considered here can be complex. Study the linear difference equation derived for this ODE by forward Euler method, backward Euler, midpoint. Find its general solutions.

2. Consider linear finite difference equation with source term
$$ay^{n+1} + by^n + cy^{n-1} = f^n$$
Given initial data $\bar{y}^0$ and $\bar{y}^1$, find its solution.

3. Find the characteristic roots for the Adams-Bashforth and Adams-Moulton schemes with steps 1-3 for the linear equation $y' = \lambda y$.

1.5 Stability analysis

1.5.1 Zero Stability

Our goal is to develop general convergence theory for numerical ODEs. First, let us see the proof of the convergence of the two stage Runge-Kutta method. The scheme can be expressed as

$$y^{n+1} = y^n + k\Psi(y^n, t^n, k) \tag{1.12}$$

where

$$\Psi(y^n, t^n, k) := f(y + \frac{1}{2}kf(y)). \tag{1.13}$$

Suppose $y(\cdot)$ is a true solution, the corresponding truncation error

$$\tau^n := \frac{y(t^{n+1}) - y(t^n)}{k} - \Psi(y(t^n), t^n, k) = O(k^2)$$

Thus, the true solution satisfies

$$y(t^{n+1}) - y(t^n) = k\Psi(y(t^n), t^n, k) + k\tau^n$$

The true error $e^n := y^n - y(t^n)$ satisfies

$$e^{n+1} = e^n + k(\Psi(y^n, t^n, k) - \Psi(y(t^n), t^n, k)) - k\tau^n.$$

This implies

$$|e^{n+1}| \leq |e^n| + k\lambda'|e^n| + k|\tau^n|.$$

Hence, we get

$$\begin{aligned} |e^n| &\leq (1 + k\lambda')^n|e^0| + k\sum_{m=0}^{n\ 1}(1 + k\lambda')^{n-1-m}|\tau^m| \\ &\leq e^{\lambda' t}|e^0| + \frac{e^{\lambda' t}}{\lambda'}\max_m |\tau^m| \end{aligned}$$

In a numerical scheme, when we fix the final time $T = nk$ and let $n \to \infty$, we want the corresponding numerical solution remains bounded, A scheme satisfies this property is called stable. We can first investigate stability for the linear equation

$$y' = \lambda y$$

Consider a finite difference scheme for this linear equation, it results in a linear finite difference equation. For instance, for the second-order Runge-Kutta in the previous example, the corresponding finite difference equation is

$$y^{n+1} = y^n + k\lambda(y^n + \frac{1}{2}k\lambda y^n) = (1 + \lambda k + \frac{1}{2}(\lambda k)^2)y^n \tag{1.14}$$

Another example, the multistep method (??), the corresponding finite difference equation is

$$\sum_{m=0}^{r} a_m y^{n+1-r+m} = k\sum_{m=0}^{r} b_m \lambda y^{n+1-r+m}$$

This gives

$$y^n = G(\lambda k, y^{n-1}, ..., y^{n-r+1}).$$

We can express this scheme in system form:

$$\mathbf{y}^n = \mathbf{G}(\lambda k)\mathbf{y}^{n-1}.$$

The stability of G means that $\|\mathbf{G}^n\|$ are uniformly bounded. The norm $\|\mathbf{G}^n\|$ has a lower bound in terms of $r(\lambda k)$, the spectral radius of $\mathbf{G}(\lambda k)$, i.e.

$$r(\lambda k) := \max_i \rho_i(\lambda k),\ \rho_i \text{ is the eigenvalues of } \mathbf{G}$$

Lemma 5.1

$$r(\mathbf{G})^n \leq \|\mathbf{G}^n\|$$

Proof. If $\mathbf{y}$ is a unit eigenvector of $\mathbf{G}$ with eigenvalue λ, then

$$|\lambda|^n = \|\mathbf{G}^n\mathbf{y}\| \leq \|\mathbf{G}^n\|$$

This yields $r(\mathbf{G})^n \leq \|\mathbf{G}^n\|$. ∎

We shall call $r(\mathbf{G})$ the amplification factor. Here are some example of the amplification factors:

$$\begin{aligned} r(z) &= |1+z| \text{ forward Euler,} \\ r(z) &= \left|1+z+\frac{z^2}{2}\right|, \text{ midpoint, RK2} \\ r(z) &= \left|\frac{1+\frac{z}{2}}{1-\frac{z}{2}}\right|, \text{ implicit trapezoidal} \end{aligned}$$

Definition 5.2 *A scheme is called zero-stable if the spectral radius of the spectral radius $r(\lambda k)$ of the corresponding amplification matrix $\mathbf{G}(k)$ (Green's function) satisfies the von Neumann condition*

$$r(\lambda k) \leq 1 + Ck \tag{1.15}$$

Theorem 5.3 *A necessary condition for*

$$\|\mathbf{G}^n\| \leq C_1$$

is that its spectral radius $r(k)$ satisfies

$$r(k) \leq 1 + C_2 k.$$

Here C_1, C_2 are independent of n.

Proof. The spectral radius of $\mathbf{G}^n$ is $r(k)^n$. Thus, $\|\mathbf{G}^n\| \leq C_1$ implies $r(k) \leq C_1^{1/n} \leq 1 + C_2/n$ for some constant C_2 independent of n. We choose $nk = t$ fixed, then $r(k) \leq 1 + C_2 k/t$. ∎

For one-step methods such as the forward Euler method, backward Euler method, Runge-Kutta methods, we can show that they are always zero-stable. For multistep method, the necessary and sufficient condition is the following theorem.

Theorem 5.4 *A multistep method (??) is zero-stable if the root ρ of its characteristic polynomial $a(\rho)$ satisfies*

$$\begin{aligned} &\textit{either } |\rho| = 1 \textit{ and } \rho \textit{ is a simple root,} \\ &\textit{or } |\rho| < 1 \end{aligned} \tag{1.16}$$

Conversely, if the scheme is zero-stable, then it is necessary that all roots of $a(\rho) = 0$ satisfy

$$|\rho| \leq 1.$$

Proof. The characteristic root of (??) satisfies

$$a(\rho) - k\lambda b(\rho) = 0, \tag{1.17}$$

where

$$a(\rho) := \sum_m a_m \rho^m, \; b(\rho) := \sum_m b_m \rho^m,$$

From the perturbation theory of roots of polynomial, if ρ_i is a root of $a(\rho) = 0$, then there corresponds a root of (1.17) and satisfying

$$\rho_i(\lambda k) = \rho_i + O(k\lambda).$$

Thus, the Green's operator G satisfies

$$\|G^n\| \le (1 + C\lambda k)^n \le e^{C\lambda n k}.$$

■

On the second case where ρ_i is a repeat root, the perturbed $\rho_i(z)$ can be repeat root or split into several simple roots. In the latter case, the corresoponding $\mathbf{G}^n$ remains bounded. However, in the first cases, it becomes unbounded. Thus, the root condition for the repeat root is only a sufficient condition for stability.

Theorem 5.5 (Dahlquist) *For finite difference schemes for ODE $y' = f(t, y)$,*

$$\textit{consistency} + \textit{zero-stability} \Leftrightarrow \textit{convergence}$$

1.5.2 Absolute Stability

In the above convergence analysis, it says that the scheme convergence if the mesh size k is small enough. In practice, we can only choose finite and proper mesh size. For instance, for linear equation $y' = \lambda y$, the forward Euler scheme satisfies

$$y^n = (1 + k\lambda)^n y^0$$

In order to have the solution remain bounded as $n \to \infty$ with $kn = T$ fixed, we need to require

$$|1 + k\lambda| < 1.$$

This gives a restriction on k. The region $\{z \| 1 + z| < 1\}$ is the region of absolute stability for the forward Euler scheme. It is then important to study the region of absolute stability in order to to choose proper mesh size in practical computation.

For general linear system of equations $y' = Ay$, the same condition should be satisfied for all eigenvalues λ_i of A. More precisely,

$$|1 + k\lambda_i| \le 1$$

if λ_i is an eigenvalue of A with multiplicity 1, and

$$|1 + k\lambda_i| < 1$$

if λ_i is an eigenvalue of A with multiplicity greater than 1.

Back to the scalar linear equation $y' = \lambda y$. Let abbreviate λk by z. Denote the amplification factor by $r(z)$. We have

$$\begin{aligned} r(z) &= 1 + z \text{ forward Euler,} \\ r(z) &= 1 + z + \frac{z^2}{2}, \text{ midpoint, RK2} \\ r(z) &= \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}}, \text{ implicit trapezoidal} \end{aligned}$$

The stability region $D := \{z | |r(z)| < 1\}$.

Chapter 2

Finite Difference Methods for Linear Parabolic Equations

2.1 A review of Heat equation

2.2 Finite Difference Methods for the Heat Equation

2.2.1 Some discretization methods

Let us start from the simplest parabolic equation, the heat equation:

$$u_t = u_{xx}$$

Let $h = \Delta x$, $k = \Delta t$ be the spatial and temporal mesh sizes. Define $x_j = jh$, $j \in \mathbb{Z}$ and $t^n = nk$, $n \geq 0$. Let us abbreviate $u(x_j, t^n)$ by u_j^n. We shall approximate u_j^n by U_j^n, where U_j^n satisfies some finite difference equations.

Spatial discretization : The simplest one is that we use centered finite difference approximation for u_{xx}:

$$u_{xx} = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + O(h^2)$$

This results in the following systems of ODEs

$$\dot{u_j}(t) = \frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{h^2}$$

or in vector form

$$\dot{U} = \frac{1}{h^2} AU$$

where $U = (u_0, u_1, ...)^t$, $A = \text{diag}\,(1, -2, 1)$.

Homeworks.

1. Derive the 4th order centered finite difference approximation for u_{xx}:

$$u_{xx} = \frac{1}{h^2}(-u_{j-2} + 16u_{j-1} - 30u_j + 16u_{j+1} - u_{j+2}) + O(h^4).$$

2. Derive a 2nd order centered finite difference approximation for $(\kappa(x)u_x)_x$.

Temporal discretization We can apply numerical ODE solvers

- Forward Euler method:
$$U^{n+1} = U^n + \frac{k}{h^2}AU^n \tag{2.1}$$

- Backward Euler method:
$$U^{n+1} = U^n + \frac{k}{h^2}AU^{n+1} \tag{2.2}$$

- 2nd order Runge-Kutta (RK2):
$$U^{n+1} - U^n = \frac{k}{h^2}AU^{n+1/2},\ U^{n+1/2} = U^n + \frac{k}{2h^2}AU^n \tag{2.3}$$

- Crank-Nicolson:
$$U^{n+1} - U^n = \frac{k}{2h^2}(AU^{n+1} + AU^n). \tag{2.4}$$

These linear finite difference equations can be solved formally as

$$U^{n+1} = GU^n$$

where

- Forward Euler: $G = 1 + \frac{k}{h^2}A$,
- Backward Euler: $G = (1 - \frac{k}{h^2}A)^{-1}$,
- RK2: $G = 1 + \frac{k}{h^2}A + \frac{1}{2}\left(\frac{k}{h^2}\right)^2 A^2$
- Crank-Nicolson: $G = \frac{1+\frac{k}{2h^2}A}{1-\frac{k}{2h^2}A}$

For the Forward Euler, We may abbreviate it as

$$U_j^{n+1} = G(U_{j-1}^n, U_j^n, U_{j+1}^n), \tag{2.5}$$

where

$$G(U_{j-1}, U_j, U_{j+1}) = U_j + \frac{k}{h^2}(U_{j-1} - 2U_j + U_{j+1})$$

2.2.2 Stability and Convergence for the Forward Euler method

Our goal is to show under what condition can U_j^n converges to $u(x_j, t^n)$ as the mesh sizes $h, k \to 0$.

To see this, we first see the local error a true solution can produce. Plug a true solution $u(x,t)$ into (2.1). We get

$$u_j^{n+1} - u_j^n = \frac{k}{h^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) + k\tau_j^n \tag{2.6}$$

where

$$\tau_j^n = D_+u_j^n - (u_t)_j^n - (D_+D_-u_j^n - (u_{xx})_j^n) = O(k) + O(h^2).$$

Let e_j^n denote for $u_j^n - U_j^n$. Then substract (2.1) from (2.6), we get

$$e_j^{n+1} - e_j^n = \frac{k}{h^2}\left(e_{j+1}^n - 2e_j^n + e_{j-1}^n\right) + k\tau_j^n \tag{2.7}$$

This can be expressed as

$$e_j^{n+1} = G(e_{j-1}^n, e_j^n, e_{j+1}^n) + k\tau_j^n \tag{2.8}$$

or in operator form:

$$e^{n+1} = \mathbf{G}(e^n) + \tau^n \tag{2.9}$$

where $e^n = (e_j^n)_{j\in\mathbb{Z}}$, $\mathbf{G}(e)_j = G(e_{j-1}, e_j, e_{j+1})$.

Suppose G satisfies

$$\|\mathbf{G}(U)\| \le \|U\|$$

under certain norm $\|\cdot\|$, we can accumulate the local truncation errors in time to get the global error as the follows.

$$\begin{aligned}
\|e^n\| &\le \|\mathbf{G}e^{n-1}\| + k\|\tau^{n-1}\| \\
&\le \|e^{n-1}\| + k\|\tau^{n-1}\| \\
&\le \|\mathbf{G}e^{n-2}\| + k(\|\tau^{n-2}\| + \|\tau^{n-1}\|) \\
&\le \|e^0\| + k(\|\tau^0\| + \cdots + \|\tau^{n-2}\| + \|\tau^{n-1}\|)
\end{aligned}$$

If the local truncation error has the estimate

$$\max_n \|\tau^n\| = O(h^2) + O(k)$$

and the initial error e^0 satisfies

$$\|e^0\| = O(h^2),$$

then so does the global true error $\|e^n\|$ for all n.

The above analysis leads to the following definitions.

Definition 2.3 *A finite difference method is called consistent if its local truncation error τ satisfies*

$$\|\tau_{h,k}\| \to 0 \text{ as } h, k \to 0.$$

Definition 2.4 *A finite difference scheme $U^{n+1} = G_{h,k}(U^n)$ is called stable under the norm $\|\cdot\|$ in a region $(h,k) \in R$ if*

$$\|G_{h,k}^n U\| \leq \|U\|$$

for all n with nk fixed.

Definition 2.5 *A finite difference method is called convergence if the true error*

$$\|e_{h,k}\| \to 0 \text{ as } h, k \to 0.$$

In the above analysis, we have seen that

$$\text{stability} + \text{consistency} \Rightarrow \text{convergence}.$$

2.3 L^2 Stability – Von Neumann Analysis

Since we only deal with smooth solutions in this section, the L^2-norm or the Sobolev norm is a proper norm to our stability analysis. For constant coefficient and scalar case, the von Neumann analysis (via Fourier method) provides a necessary and sufficient condition for stability. For system with constant coefficients, the von Neumann analysis gives a necessary condition for statbility. For systems with variable coefficients, the Kreiss' matrix theorem provides characterizations of stability condition.

Below, we give L^2 stability analysis. We use two methods, one is the energy method, the other is the Fourier method, that is the von Neumann analysis. We describe the von Neumann analysis below.

Given $\{U_j\}_{j\in\mathbb{Z}}$, we define

$$\|U\|^2 = \sum_j |U_j|^2$$

and its Fourier transform

$$\hat{U}(\xi) = \frac{1}{2\pi} \sum U_j e^{-ij\xi}.$$

The advantages of Fourier method for analyzing finite difference scheme are

- the shift operator is transformed to a multiplier:

$$\widehat{TU}(\xi) = e^{i\xi}\hat{U}(\xi),$$

  where $(TU)_j := U_{j+1}$;

- the Parseval equility

$$\begin{aligned} \|U\|^2 &= \|\hat{U}\|^2 \\ &\equiv \int_\pi^\pi |\hat{U}(\xi)|^2 \, d\xi. \end{aligned}$$

If a finite difference scheme is expressed as

$$U_j^{n+1} = (GU^n)_j = \sum_{i=-l}^{m} a_i (T^i U^n)_j,$$

then

$$\widehat{U^{n+1}} = \widehat{G}(\xi)\widehat{U^n}(\xi).$$

From the Parseval equality,

$$\begin{aligned} \|U^{n+1}\|^2 &= \|\widehat{U^{n+1}}\|^2 \\ &= \int_{-\pi}^{\pi} |\widehat{G}(\xi)|^2 \, |\widehat{U^n}(\xi)|^2 \, d\xi \\ &\leq \max_{\xi} |\widehat{G}(\xi)|^2 \int_{-\pi}^{\pi} |\widehat{U^n}(\xi)|^2 \, d\xi \\ &= |\widehat{G}|_\infty^2 \|U\|^2 \end{aligned}$$

Thus a sufficient condition for stability is

$$|\widehat{G}|_\infty \leq 1. \tag{2.10}$$

Conversely, suppose $|\widehat{G}(\xi_0)| > 1$, from $\widehat{G}$ being a smooth function in ξ, we can find ϵ and δ such that

$$|\widehat{G}(\xi)| \geq 1 + \epsilon \text{ for all } |\xi - \xi_0| < \delta.$$

Let us choose an initial data U_0 in ℓ^2 such that $\widehat{U^0}(\xi) = 1$ for $|\xi - \xi_0| \leq \delta$. Then

$$\begin{aligned} \|\widehat{U^n}\|^2 &= \int |\widehat{G}|^{2n}(\xi) |\widehat{U^0}|^2 \\ &\geq \int_{|\xi - \xi_0| \leq \delta} |\widehat{G}|^{2n}(\xi) |\widehat{U^0}|^2 \\ &\geq (1+\epsilon)^{2n} \delta \to \infty \text{ as } n \to \infty \end{aligned}$$

Thus, the scheme can not be stable. We conclude the above discussion by the following theorem.

Theorem 3.6 *A finite difference scheme*

$$U_j^{n+1} = \sum_{k=-l}^{m} a_k U_{j+k}^n$$

with constant coefficients is stable if and only if

$$\widehat{G}(\xi) := \sum_{k=-l}^{m} a_k e^{-ik\xi}$$

satisfies

$$\max_{\pi \leq \xi \leq \pi} |\widehat{G}(\xi)| \leq 1. \tag{2.11}$$

Homeworks.

1. *Compute the $\widehat{G}$ for the schemes: Forward Euler, Backward Euler, RK2 and Crank-Nicolson.*

2.4 Energy method

We write the finite difference scheme as

$$U_j^{n+1} = \alpha U_{j-1}^n + \beta U_j^n + \gamma U_{j+1}^n, \tag{2.12}$$

where

$$\alpha, \beta, \gamma \geq 0 \text{ and } \alpha + \beta + \gamma = 1.$$

We multiply (2.12) by U_j^{n+1} on both sides, apply Cauchy-Schwarz inequality, we get

$$\begin{aligned}(U_j^{n+1})^2 &= \alpha U_{j-1}^n U_j^{n+1} + \beta U_j^n U_j^{n+1} + \gamma U_{j+1}^n U_j^{n+1} \\ &\leq \frac{\alpha}{2}((U_{j-1}^n)^2 + (U_j^{n+1})^2) + \frac{\beta}{2}((U_j^n)^2 + (U_j^{n+1})^2) + \frac{\gamma}{2}((U_{j+1}^n)^2 + (U_j^{n+1})^2)\end{aligned}$$

Here, we have used $\alpha, \beta, \gamma \geq 0$. We multiply this inequality by h and sum it over $j \in \mathbb{Z}$. Denote

$$\|U\|_2 := \left(\sum_j |U_j|^2 h\right)^{1/2}.$$

We get

$$\begin{aligned}\|U^{n+1}\|^2 &\leq \frac{\alpha}{2}(\|U^n\|^2 + \|U^{n+1}\|^2) + \frac{\beta}{2}(\|U^n\|^2 + \|U^{n+1}\|^2) + \frac{\gamma}{2}(\|U^n\|^2 + \|U^{n+1}\|^2) \\ &- \frac{1}{2}(\|U^n\|^2 + \|U^{n+1}\|^2).\end{aligned}$$

Here, $\alpha + \beta + \gamma = 1$ is applied. Thus, we get the energy estimate

$$\|U^{n+1}\|^2 \leq \|U^n\|^2. \tag{2.13}$$

Homeworks.

1. *Can the RK-2 method possess an energy estimate?*

2.5 Stability Analysis for Montone Operators– Entropy Estimates

Stbility in the maximum norm

We notice that the action of G is a convex combinition of U_{j-1}, U_j, U_{j+1}, provided

$$0 < \frac{k}{h^2} \leq \frac{1}{2}. \tag{2.14}$$

Thus, we get

$$\min\{U_{j-1}^n, U_j^n, U_{j+1}^n\} \le U_j^{n+1} \le \max\{U_{j-1}^n, U_j^n, U_{j+1}^n\}.$$

This leads to

$$\min_j U_j^{n+1} \ge \min_j U_j^n,$$
$$\max_j U_j^{n+1} \le \max_j U_j^n$$

and

$$\max_j |U_j^{n+1}| \le \max_j |U_j^n|$$

Such an operator G is called a monotone operator.

Entropy estimates

The property that U^{n+1} is a convex combination (average) of U^n is very important. Given any convex function $\eta(u)$, called entropy function, by Jenson's inequality,

$$\eta(U_j^{n+1}) \le \alpha\eta(U_{j-1}^n) + \beta\eta(U_j^n) + \gamma\eta(U_{j+1}^n) \tag{2.15}$$

Summing over all j and using $\alpha + \beta + \gamma = 1$, we get

$$\sum_j \eta(U_j^{n+1}) \le \sum_j \eta(U_j^n). \tag{2.16}$$

This means that the "entropy" decreases in time. In particular, we choose

- $\eta(u) = |u|^2$, we recover the L^2 stability,
- $\eta(u) = |u|^p$, $1 \le p < \infty$, we get

$$\sum_j |U_j^{n+1}|^p \le \sum_j |U_j^n|^p$$

This leads to

$$\left(\sum_j |U_j^{n+1}|^p h\right)^{1/p} \le \left(\sum_j |U_j^n|^p h\right)^{1/p},$$

the general L^p stability. Taking $p \to \infty$, we recover L^∞ stability.

- $\eta(u) = |u - c|$ for any constant c, we obtain Kruzkov's entropy estimate.

Homeworks.

1. Show that the solution of the difference equation derived from the RK2 satisfies the entropy estimate. What is the condition required on h and k for such entropy estimate?

2.6 Entropy estimate for backward Euler method

In the backward Euler method, the amplification matrix is given by

$$G = (I - \lambda A)^{-1} \tag{2.17}$$

where

$$\lambda = \frac{k}{h^2}, \ A = \ \text{diag}(1, -2, 1).$$

The matrix $M := I - \lambda A$ has the following property:

$$m_{ii} > 0, \ m_{ij} \leq 0, \ \sum_{j \neq i} |m_{ij}| \leq m_{ii} \tag{2.18}$$

Such a matrix is called an M-matrix.

Theorem 6.7 *The inverse of an M-matrix is a nonnegative matrix, i.e. all its entries are non-negative.*

I shall not prove this general theorem. Instead, I will find the inverse of M. Let us express

$$M = \frac{1 + 2\lambda}{2} \ \text{diag} \ (-a, 2, -a)$$

In our case,

$$a = \frac{2\lambda}{1 + 2\lambda}$$

The general solution of the difference equation

$$-au_{j-1} + 2u_j - au_{j+1} = 0 \tag{2.19}$$

has the form:

$$u_j = C_1 \rho_1^j + C_2 \rho_2^j$$

where ρ_1, ρ_2 are the characteristic roots, i.e. the roots of the polynomial

$$-a\rho^2 + 2\rho - a = 0.$$

Thus,

$$\rho_i = \frac{1 \pm \sqrt{1 - a^2}}{a}.$$

From the assumption:

$$0 < a < 1,$$

we have $\rho_1 < 1$ and $\rho_2 > 1$.

Now, we define a fundamental solution:

$$g_j = \begin{cases} \rho_1^j & \text{for } j \geq 0 \\ \rho_2^j & \text{for } j < 0 \end{cases}$$

We can check that $g_j \to 0$ as $|j| \to \infty$. g_j satisfies the difference equation (2.19) for $|j| \geq 1$. For j 0, we have

$$-ag_{-1} + 2g_0 - ag_1 = -a\rho_2^{-1} + 2 - a\rho_1 = 2 - a(\rho_1 + \rho_2^{-1}) = d$$

We reset $g_j \leftarrow g_j/d$. Then we have

$$\sum_j g_{i-j} m_j = \delta_{i,0}$$

Thus, M^{-1} is a positive matrix (i.e. all its entries are positive). In fact,

$$\sum_j g_{i-j} = 1 \text{ for all } i$$

Such a matrix appears in probability called transition matrix of a Markov chain.

Let us go back to our backward Euler method for the heat equation, we get that

$$U^{n+1} = (1 - \lambda A)^{-1} = GU^n$$

where

$$(GU)_i = \sum_j g_{i-j} U_j$$

We can think U_j^{n+1} is a *convex combination* of U_j^n with weights g_j. This weight has the properties:

- $g_j > 0$
- $\sum_j g_j = 1$

Thus, G is a *monotone operator.* With this property, we can apply Jansen's inequality to get the entropy estimates:

Theorem 6.8 *Let $\eta(u)$ be a convex function. Let U_j^n be a solution of the difference equation derived from the backward Euler method for the heat equation. Then we have*

$$\sum_j \eta(U_j^n) \leq \sum_j \eta(U_j^0). \tag{2.20}$$

Homeworks.

1. Can the Crank-Nicolson method for the heat equation satisfy the entropy estimate? What is the condition on h and k?

2.7 Existence Theory

2.7.1 Existence via forward Euler method

From energy estimate, we get $\|U^n\| \leq \|U^0\|$. We can take dinite difference quotient to the equation (forward Euler equation, for instance), then $(D_{x,+}U)_j^n := U_{j+1}^n - U_j^n$ also satisfies the same equation. Thus, it also has the same estimate for $D_{x,+}U$. Similar estimate for $D_{x,+}^2 U$. We have

$$\|D_{x,+}^m U^n\| \leq \|D_{x,+}^m U^0\|. \tag{2.21}$$

If we assume the initial data $f \in H^2$, then we get $U^n \in H_h^2$. Here, $h = 1/n$.

For any discrete function $U_j \in H_h^m$ we can construct a function u in H^m defined by

$$u(x) := \sum_j U_j \phi_h(x - x_j) \tag{2.22}$$

where $\phi_h(x) = \text{sinc}(x/h)$. We have

$$u_h(x_j) = U_j$$

It can be shown that

$$\|D_x^m u_h\| \equiv \|D_{x,+}^m U\|. \tag{2.23}$$

Similarly, the space $L_k^\infty(H_h^m)$ can be embeded into $L^\infty(H^m)$ by defining

$$u_{h,k}(x,t) = \sum_{n\geq 0}\sum_j U_j^n \phi_k(t)\phi_h(x)$$

The discrete norm and the continuous norm are equivalent.

With this background, we get

Theorem 7.9 *If the initial data $\phi \in H^m$,$m \geq 2$ and $k/h^2 \leq 1/2$, then the solution of forward Euler equation has the estimate*

$$\|D_{x,+}^m U^n\| \leq \|D_{x,+}^m U^0\|, \|D_{t,+}U^n\| \leq \|D_{x,+}^2 U^0\| \tag{2.24}$$

Further, the corresponding smoothing function $u_{h,k}$ has the same estimate and has a subsequence converges to a solution $u(x,t)$ of the original equation.

Proof. The functions $u_{h,k}$ are unformly bounded in $W^{1,\infty}(H^2)$. Hence they have a subsequence converges strongly in $L^\infty(H^1)$ and $u \in W^{1,\infty}(H^2)$. The functions $u_{h,k}$ satisfy

$$u_{h,k}(x_j, t^{n+1}) - u_{h,k}(x_j, t^n) = \frac{k}{h^2}(u_{h,k}(x_{j-1}, t^n) - 2u_{h,k}(x_j, t^n) + u_{h,k}(x_{j+1}, t^n))$$

Multiply a test smooth function ϕ, sum over j and n, take summation by part, we can get the subsubsequence converges to a solution of $u_t = u_{xx}$ *weakly*. ∎

2.7.2 A Sharper Energy Estimate for backward Euler method

Let us see that we can have a sharper energy estimate for the finite difference derived by backward Euler method. Recall the backward Euler method for solving the heat equation is

$$U_j^{n+1} - U_j^n = \lambda(U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1}) \tag{2.25}$$

An important technique is the summation by part:

$$\sum_j (U_j - U_{j-1})V_j = -\sum_j U_j(V_{j+1} - V_j) \tag{2.26}$$

There is no boundary term because we consider periodic condition in the present case.

Now We multiply both sides by U_j^{n+1}, then sum over j. We get

$$\sum_j (U_j^{n+1})^2 - U_j^{n+1}U_j^n = -\lambda \sum_j |U_{j+1}^{n+1} - U_j^{n+1}|^2$$

The term

$$U_j^{n+1}U_j^n \le \frac{1}{2}((U_j^{n+1})^2 - (U_j^n)^2))$$

Hence, we get

$$\frac{1}{2}\sum_j \left((U_j^{n+1})^2 - (U_j^n)^2)\right) \le -\lambda \sum_j |U_{j+1}^{n+1} - U_j^{n+1}|^2$$

Or

$$\frac{1}{2}\|D_{t,-}U^{n+1}\| \le -k\frac{h}{k^2}\frac{k}{h^2}\|D_{x,+}U^{n+1}\| \tag{2.27}$$

Here,

$$D_{t,-}U_j^{n+1} := \frac{U_j^{n+1} - U_j^n}{k},\ D_{x,+}U_j^{n+1} := \frac{U_{j+1}^{n+1} - U_j^{n+1}}{h},$$

Theorem 7.10 *For the backward Euler method, we have the estimate*

$$\|U^N\|^2 + C\sum_{n=1}^{N} \|D_{x,+}U^n\|^2 \le \|U^0\|^2 \tag{2.28}$$

This gives controls not only on $\|U^n\|^2$ but also on $\|D_{x,+}U^n\|$.

Homeworks.

1. Show that the Crank-Nicolson method also has similar energy estimate.
2. Can forward Euler method have similar energy estimate?

2.8 Relaxation of errors

In this section, we want to study the evolution of an error. We consider

$$u_t = u_{xx} + f(x) \tag{2.29}$$

with initial data ϕ. The error $e_j^n := u(x_j, t^n) - U_j^n$ satisfies

$$e_j^{n+1} = e_j^n + \lambda(e_{j-1}^n - 2e_j^n + e_{j+1}^n) + k\tau_j^n \tag{2.30}$$

We want to know how error is relaxed to zero from an initial error e^0. We study the homogeneous finite difference quation first. That is

$$e_j^{n+1} = e_j^n + \lambda(e_{j-1}^n - 2e_j^n + e_{j+1}^n). \tag{2.31}$$

or $e^{n+1} = G(u^n)$. The matrix is a tridiagonal matrix. It can be diagonalized by Fourier method. The eigenfunctions and eigenvalues are

$$v_{k,j} = e^{2\pi ijk/N}, \rho_k = 1 - 2\lambda + 2\lambda\cos(2\pi k/N) = 1 - 4\lambda\sin^2(\pi k/N), k = 0, ..., N-1.$$

When $\lambda \leq 1/2$, all eigenvalues are negative except ρ_0:

$$1 = \rho_0 > |\rho_1| > |\rho_2| > \cdots.$$

The eigenfunction

$$v_0 \equiv 1.$$

Hence, the projection of any discrete function U onto this eigenfunction is the average: $\sum_j U_j$.

Now, we decompose the error into

$$e^n = \sum e_k^n v_k$$

Then

$$e_k^{n+1} = \rho_k e_k^n.$$

Thus,

$$e_k^n = \rho_k^n e_k^0.$$

We see that e_k^n decays exponentially fast except e_0^n, which is the average of e^0. Thus, the average of initial error never decay unless we choose it zero. To guarantee the average of e^0 is zero, we may choose U_j^n to be the cell average of $u(x, t^n)$ in the jth cell:

$$U_j^n = \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t^n)\, dx.$$

instead of the grid data. This implies the initial error has zero local averages, and thus so does the global average.

The contribution of the truncation to the true solution is:

$$e^{n+1} = \rho_k e_k^n + \Delta t \tau_k^n$$

Its solution is

$$e_k^n = \rho_k^n e_k^0 + \Delta t \sum_{m=0}^{n-1} \rho_k^{n-1-m} \tau_k^m$$

We see that the term e_0^n does not tend to zero unless $\tau_0^m = 0$. This can be achieved if we choose U_j as well as f_j to be the cell averages instead the grid data.

Homeworks.

1. Define $U_j := \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x)\,dx$. Show that if $u(x)$ is a smooth periodic function on $[0,1]$, then
$$u''(x_j) = \frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) + \tau$$
with $\tau = O(h^2)$.

2.9 Boundary Conditions

2.9.1 Dirichlet boundary condition

Dirichlet boundary condition is

$$u(0) = a,\ u(1) = b \tag{2.32}$$

The finite difference approximation of u_{xx} at x_1 involves u at $x_0 = 0$. We plug the boundary contion:

$$u_{xx}(x_1) = \frac{U_0 - 2U_1 + U_2}{h^2} + O(h^2) = \frac{a - 2U_1 + U_2}{h^2} + O(h^2)$$

Silimarly,

$$u_{xx}(x_{N-1}) = \frac{U_{N-2} - 2U_{N-1} + U_N}{h^2} + O(h^2) = \frac{U_{N-2} - 2U_{N-1} + b}{h^2} + O(h^2)$$

The unknowns are $U_1^n, ..., U_{N-1}^n$ with $N-1$ finite difference at $x_1, ..., x_{N-1}$. The discrete Laplacian becomes

$$A = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix}. \tag{2.33}$$

This discrete Lalacian is the same as a discrete Laplacian with zero Dirichlet boundary condition.

We can have energy estimates, entropy estimates as the case of periodic boundary condition.

Next, we exame how error is relaxed for the Euler method with zero Dirichlet boundary condition. From Fourier method, we observe that the eigenfunctions and eigenvalues for the forward Euler method are

$$v_{k,j} = \sin(2\pi jk/N), \rho_k = 1 - 2\lambda + 2\lambda\cos(2\pi k/N) = 1 - 4\lambda\sin^2(\pi k/N), k = 1, ..., N-1.$$

In the present case, all eigenvalues

$$\rho_i < 1, i = 1, ..., N-1.$$

provided the stability condition

$$\lambda \le 1/2.$$

Thus, the errors e_i^n decays to zero exponentially for all $i = 1, ..., N-1$. The slowest mode is ρ_1 which is

$$\rho_1 = 1 - 4\lambda \sin^2(\pi/N) \approx 1 - 4\left(\frac{\pi}{N}\right)^2$$

and

$$\rho_1^n \approx \left(1 - 4\left(\frac{\pi}{N}\right)^2\right)^n \approx e^{-4\pi^2 t}$$

where we have used k/h^2 is fixed and $nk = t$.

2.9.2 Neumann boundary condition

The Neumann boundary condition is

$$u'(0) = \sigma_0, \; u'(1) = \sigma_1. \tag{2.34}$$

We may use the following disrete discretization methods:

- First order:
$$\frac{U_1 - U_0}{h} = \sigma_0.$$
- Second order-I:
$$\frac{U_1 - U_0}{h} = u_x(x_{1/2}) = u_x(0) + \frac{h}{2}u_{xx}(x_0) = \sigma_0 + \frac{h}{2}f(x_0)$$
- Second order-II: we use extrapolation
$$\frac{3U_0 - 2U_1 + U_2}{2h^2} = \sigma_0.$$

The knowns are U_j^n with $j = 0, ..., N$. In the mean time, we add two more equations at the boundaries.

Homeworks.

1. Find the eigenfunctions and eigenvalues for the discrete Laplacian with the Neumann boundary condition (consider both first order and second order approximation at boundary). Notice that there is a zero eigenvalue.

   Hint: You may use Matlab to find the eigenvalues and eigenvectors.

Here, I will provide another method. Suppose A is the discrete Laplacian with Neumann boundary condition. A is an $(N+1) \times (N+1)$ matrix. Suppose $Av = \lambda v$. Then for $j = 1, ..., N-1$, v satisfies

$$v_{j-1} - 2v_j + v_{j+1} = \lambda v_j, j = 1, ..., N-1.$$

For v_0, we have

$$v_0 - v_1 = \lambda v_0.$$

For v_N, we have

$$v_N - v_{N-1} = \lambda v_N.$$

Suppose the general solution has the ansatz:

$$C_1\rho_1^j + \rho_2^j,\ j = 0, ..., N,$$

where ρ_1, ρ_2 satisfy

$$\rho^2 - (2+\lambda)\rho + 1 = 0,$$

At x_0, we have $v_0 - v_1 = \lambda v_0$, i.e. we get

$$(1-\lambda)(C_1+1) = C_1\rho_1 + \rho_2$$

At x_N, we have $v_N - v_1 = \lambda v_N$, i.e.

$$(1-\lambda)(C_1\rho_1^N + \rho_2^N) = C_1\rho_1^{N-1} + \rho_2^{N-1}.$$

There are two equations for two unknowns λ and C_1. We can get nontrivial solutions for suitable λ and C_1. Finally, we normalize v to be a unite vector.

Homeworks.

1. Complete the calculation.
2. Consider

   $$u_t = u_{xx} + f(x)$$

   on $[0, 1]$ with Neumann boundary condition $u'(0) = u'(1) = 0$. If $\int f(x)\,dx \neq 0$. What wil happen to u as $t \to \infty$?

2.10 The discrete Laplacian and its inversion

We consider the elliptic equation

$$u_{xx} - \alpha u = f(x), x \in (0, 1)$$

2.10.1 Dirichlet boundary condition

Dirichlet boundary condition is

$$u(0) = a,\ u(1) = b \tag{2.35}$$

The finite difference approximation of u_{xx} at x_1 involves u at $x_0 = 0$. We plug the boundary contion:

$$u_{xx}(x_1) = \frac{U_0 - 2U_1 + U_2}{h^2} + O(h^2) = \frac{a - 2U_1 + U_2}{h^2} + O(h^2)$$

Similarly,

$$u_{xx}(x_{N-1}) = \frac{U_{N-2} - 2U_{N-1} + U_N}{h^2} + O(h^2) = \frac{U_{N-2} - 2U_{N-1} + b}{h^2} + O(h^2)$$

The unknowns are $U_1^n, ..., U_{N-1}^n$ with $N-1$ finite difference at $x_1, ..., x_{N-1}$. The discrete Laplacian becomes

$$A = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix}. \tag{2.36}$$

This is the discrete Lalacian with Dirichlet boundary condition. In one dimension, we can solve A^{-1} explicitly. Let us solve $(A - 2\beta)^{-1}$ where $\beta = \alpha h^2/2$. The difference equation

$$U_{j-1} - (2 + 2\beta)U_j + U_{j+1} = 0$$

has two independent solutions ρ_1 and ρ_2, where ρ_i are roots of

$$\rho^2 - (2 + 2\beta)\rho + 1 = 0.$$

That is

$$\rho = 1 + \beta \pm \sqrt{(1+\beta)^2 - 1}$$

When $\beta = 0$, the two solutions are $U_j = 1$ and $U_j = j$. This gives the fundamental solution

$$G_{i,j} = \begin{cases} jC_i & j \le i \\ (N-j)C_i' & j \ge i \end{cases}$$

From $G_{i,i-1} - 2G_{i,i} + G_{i,i+1} = 1$ and $iC_i = (N-i)C_i'$ we get $C_i = -(N-i)/N$ and $C_i' = -i/N$.

When $\beta > 0$, the two roots are $\rho_1 < 1$ and $\rho_2 > 1$.

Homeworks.

1. Use matlab or maple to find the fundamental solution $G_{i,j} := (A - 2\beta)^{-1}$ with $\beta > 0$.
2. Is it correct that $v_{i,j}$ has the following form?

$$G_{i,j} = \begin{cases} \rho_1^{j-i} & N-1 > j \ge i \\ \rho_2^{j-i} & 1 < j < i \end{cases}$$

Let us go back to the original equation:

$$u_{xx} - \alpha u = f(x)$$

The above study of the Green's function of the discrete Laplacian helps us to quantify the the error produced from the source term. If $Au = f$ and $A^{-1} = G$, then an error in f, say τ, will produce an error

$$e = G\tau.$$

If the off-diagonal part of G decays exponentially (i.e. $\beta > 0$), then the error is "localized," otherwise, it polutes everywhere. The error from the boundary also has the same behavior. Indeed, if $\beta = 0$, then The discrete solution is

$$u(x_j) = aG_0(j) + bG_1(j) + \sum_j G_{i,j} f_j$$

where $G(j) = jh$, $G_1(j) = 1 - jh$ and $G = A^{-1}$, the Green's function with zero Dirichlet boundary condition. Here, G_0 solves the equation

$$G_0(i-1) - 2G_0(i) + G_0(i+1) = 0, i = 1, ..., N-1,$$

for $j = 1, ..., N-1$ with $G_0(0) = 1$ and $G_0(N) = 0$. And G_1 solves the same equation with $G_1(0) = 0$ and $G_1(N) = 1$.

If $\beta > 0$, we can see that both G_0 and G_1 are also localized.

Project 2. Solve the following equation

$$\alpha u_{xx} - \beta u + f(x) = 0, \ x \in [0, 1]$$

numerically with periodic, Dirichlet and Neumann boundary condition. The equilibrium

1. A layer structure
$$f(x) = \begin{cases} -1 & 1/4 < x < 3/4 \\ 1 & \text{otherwise} \end{cases}$$
2. An impluse
$$f(x) = \begin{cases} \gamma & 1/2 - \delta < x < 1/2 + \delta \\ 0 & \text{otherwise} \end{cases}$$
3. A dipole
$$f(x) = \begin{cases} \gamma & 1/2 - \delta < x < 1/2 \\ -\gamma & 1/2 < x < 1/2 + \delta \\ 0 & \text{otherwise} \end{cases}$$

You may choose $\alpha = 0.1, 1$, $\beta = 0.1, 1, 2$. Observe how solutions change as you vary α and β.

Project 3. Solve the following equation

$$-u_{xx} + f(u) = g(x), \ x \in [0, 1]$$

numerically with Neumann boundary condition. Here, $f(u) = F'(u)$ and the potential is

$$F(u) = u^4 - \gamma u^2.$$

Study the solution as a function of γ. Choose simple g, say piecewise constant, a delta function, or a dipole.