AM213B Numerical Methods for the Solution of Differential Equations

Lecture 17

List of topics in this lecture

- General theory: a high-resolution method must be non-linear
- The split-operator method for solving 2D problems
- Numerical solution of the Poisson equation: numerical discretization, truncation error, error in numerical solution, relation between the two

---

Review:

A framework for non-oscillating high resolution methods

We use a coefficient to switch the correction on and off and in-between

$$F_{i+1/2}^{(\mathrm{HR})} = \underbrace{F_{i+1/2}^{(Up)}}_{\text{Upwind}} + \phi_{i+1/2} \underbrace{\left[F_{i+1/2}^{(\mathrm{LW})} - F_{i+1/2}^{(Up)} \right]}_{\text{Correction}}$$

where the switching coefficient ϕ is

$$\phi_{i+1/2} = \phi\left(\frac{\Delta u_{i-1/2}^{n}}{\Delta u_{i+1/2}^{n}},\ \frac{\Delta u_{i+3/2}^{n}}{\Delta u_{i+1/2}^{n}} \right), \qquad \Delta u_{i+1/2}^{n} = u_{i+1}^{n} - u_{i}^{n}$$

$$\phi(c_L, c_R) = \max\big(0,\ \min(1,\ q c_L,\ q c_R)\big), \qquad 1 \le q \le 2 \ \text{ is a parameter}$$

End of review

Observation:

For linear PDE $u_t + a u_x = 0$ $(a > 0)$, the two underlying methods are linear.

$$F_{i+1/2}^{(Up)} = a u_i^n$$

$$F_{i+1/2}^{(\mathrm{LW})} = \frac{a}{2}(u_{i+1}^n + u_i^n) - \frac{\Delta t}{2\Delta x} a^2 (u_{i+1}^n - u_i^n)$$

Both the upwind method and the Lax-Wendroff method are linear in $\{u\}$.

However, the high-resolution method is non-linear because ϕ is intrinsically non-linear in $\{u\}$ even when the PDE is linear.

<u>Question:</u> Can we have a high-resolution method that is not intrinsically non-linear?

<u>Answer:</u> A high resolution method must be intrinsically non-linear.

(See Appendix A for the discussion)

The split-operator method for solving 2D problems

Consider the 2D conservation law

$$\frac{\partial u}{\partial t}+\frac{\partial F_1(u)}{\partial x}+\frac{\partial F_2(u)}{\partial y}=0 \qquad \text{(E2D)}$$

We recognize that

$$\frac{\partial u(x,y,t)}{\partial t}+\frac{\partial F_1(u(x,y,t))}{\partial x}=0 \qquad \text{(E1D-X)}$$

is a 1D problem with parameter y.

$$\frac{\partial u(x,y,t)}{\partial t}+\frac{\partial F_2(u(x,y,t))}{\partial y}=0 \qquad \text{(E1D-Y)}$$

is a 1D problem with parameter x.

We can use any 1D solver on these two 1D problems.

<u>Motivation:</u>

If we can convert the task of solving 2D problem (E2D) to solving 1D problems (E1D-X) and (E1D-Y), then we can use any 1D solver …

<u>Basic idea:</u>

We write (E2D) in the operator form

$$u_t=(L_X+L_Y)[u]\,, \qquad L_X[u]\equiv\frac{\partial F_1(u)}{\partial x}\,, \quad L_Y\equiv\frac{\partial F_2(u)}{\partial y} \qquad \text{(E2D)}$$

Formally, the solution is given by

$$u(t)=u(0)\exp\big((L_X+L_Y)t\big)$$

In the operator form, (E1D-X) and (E1D-Y) becomes

$u_t = L_X[u]$ (E1D-X)

==> $u(t)=u(0)\exp(L_X t)$

$u_t = L_Y[u]$ (E1D-Y)

==> $u(t)=u(0)\exp(L_Y t)$

We like to connect the 2-D solution operator $\exp((L_X+L_Y)t)$ with the 1-D solution operators $\exp(L_X t)$ and $\exp(L_Y t)$.

For a small time step Δt, the 2-D solution operator has the expansion

$$\begin{aligned}\exp\left((L_X+L_Y)\Delta t\right) &= I+(L_X+L_Y)\Delta t+\frac{1}{2}(L_X+L_Y)^2(\Delta t)^2+O\left((\Delta t)^3\right)\\ &= I+(L_X+L_Y)\Delta t+\frac{1}{2}(L_X{}^2+L_XL_Y+L_YL_X+L_Y{}^2)(\Delta t)^2+O\left((\Delta t)^3\right)\end{aligned} \qquad \text{(E01)}$$

Caution:

In general, operators A and B do not commute: $AB \neq BA$. Consequently,

$$\exp(A\Delta t)\exp(B\Delta t)\neq\exp\left((A+B)\Delta t\right)$$

Example:

$$A[u]\equiv\frac{\partial(yu)}{\partial x}, \qquad B[u]\equiv\frac{\partial u}{\partial y}$$

$$B\left[A[u]\right]=B\left[yu_x\right]=\frac{\partial(yu_x)}{\partial y}=yu_{xy}+u_x$$

$$A\left[B[u]\right]=A\left[u_y\right]=\frac{\partial(yu_y)}{\partial x}=yu_{xy}$$

$$B\left[A[u]\right]\neq A\left[B[u]\right]$$

We calculate $\exp(L_Y\Delta t)\exp(L_X\Delta t)$.

$$\begin{aligned}&\exp(L_Y\Delta t)\exp(L_X\Delta t)\\ &= \left(I+L_Y\Delta t+\frac{1}{2}L_Y{}^2(\Delta t)^2\right)\left(I+L_X\Delta t+\frac{1}{2}L_X{}^2(\Delta t)^2\right)+O\left((\Delta t)^3\right)\\ &= I+\left(L_Y+L_X\right)\Delta t+\frac{1}{2}\left(L_Y{}^2+2L_YL_X+L_X{}^2\right)(\Delta t)^2+O\left((\Delta t)^3\right)\end{aligned} \qquad \text{(E02)}$$

Comparing (E02) and (E01), we obtain

$$\exp(L_Y\Delta t)\exp(L_X\Delta t)=\exp\left((L_X+L_Y)\Delta t\right)+\frac{1}{2}\left(L_YL_X-L_XL_Y\right)(\Delta t)^2+O\left((\Delta t)^3\right)$$

First order split-operator method

$$\exp\left((L_X+L_Y)\Delta t\right)=\underbrace{\exp(L_Y\Delta t)}_{\substack{\text{One }\Delta t\text{ step}\\ \text{of (E1D-Y)}}}\underbrace{\exp(L_X\Delta t)}_{\substack{\text{One }\Delta t\text{ step}\\ \text{of (E1D-X)}}}+O\left((\Delta t)^2\right)$$

Each Δt–step of (E2D) consists of

- One Δt–step of (E1D-X)
- One Δt–step of (E1D-Y)

Question: How to get the second order?

In Appendix B, we derive

$$\exp(L_X \frac{\Delta t}{2})\exp(L_Y \Delta t)\exp(L_X \frac{\Delta t}{2}) = \exp\big((L_X + L_Y)\Delta t\big) + O\big((\Delta t)^3\big)$$

Second order split-operator method

$$\exp\big((L_X + L_Y)\Delta t\big) = \underbrace{\exp(L_X \frac{\Delta t}{2})}_{\substack{\text{One } \Delta t/2 \text{ step} \\ \text{of (E1D-X)}}}\underbrace{\exp(L_Y \Delta t)}_{\substack{\text{One } \Delta t \text{ step} \\ \text{of (E1D-Y)}}}\underbrace{\exp(L_X \frac{\Delta t}{2})}_{\substack{\text{One } \Delta t/2 \text{ step} \\ \text{of (E1D-X)}}} + O\big((\Delta t)^3\big)$$

Each Δt–step of (E2D) consists of

- One $(\Delta t/2)$–step of (E1D-X)
- One Δt–step of (E1D-Y)
- One $(\Delta t/2)$–step of (E1D-X)

Remark:

The second order split-operator method is a powerful tool. Any 2D problem can be solved accurately using a second order 1D solver (i.e., the Lax-Wendroff method in the case of smooth solution or the non-oscillating high-resolution method…).

Numerical solution of the Poisson equation

Notation and background:

By convention, the Poisson equation is written in the form of

$$-\nabla^2 u(\vec{x}) = s(\vec{x}) \qquad \text{(with a negative sign on the left)}$$

where ∇^2 is the Laplace operator.

$$\nabla^2 u(\vec{x}) \equiv \frac{\partial^2 u(\vec{x})}{\partial x_1^2} + \frac{\partial^2 u(\vec{x})}{\partial x_2^2} + \cdots + \frac{\partial^2 u(\vec{x})}{\partial x_n^2}$$

Question:

Why do we have a negative sign in front of the Laplace operator?

Answer:

The solution of the Poisson equation is viewed as the steady state of the heat equation with a source term:

$$u_t(\vec{x}, t) = \nabla^2 u(\vec{x}, t) + s(\vec{x})$$

At the steady state, $u_t = 0$, we have

$$-\nabla^2 u(\vec{x}) = s(\vec{x})$$

For method development and analysis, we consider a model problem.

Dirichlet BVP of the 1-D Poisson equation

$$\begin{cases} -u''(x) = s(x) , \quad 0 < x < L \\ u(0) = c , \quad u(L) = d \end{cases} \qquad \text{(BVP-1)}$$

Keep in mind that real applications of the methods developed here are for solving 2-D and 3-D Poisson equations with variable coefficients.

Numerical grid:

$$h = \frac{L}{N} , \qquad x_i = i\,h , \qquad x_0 = 0 , \quad x_N = L$$

u_i = numerical approximation of $u(x_i)$

$s_i = s(x_i)$

Discretization:

$$-\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}\right) = s_i , \qquad 1 \le i \le (N-1) \quad \text{(internal points)}$$

$$u_0 = c , \qquad u_N = d$$

We write the numerical discretization in the operator-vector form.

We introduce vectors u and s , and linear operator T_1.

$$u = (u_1, u_2, \ldots, u_{N-1})^T = \{u_i,\ 1 \le i \le (N-1)\} \quad \text{(internal points)}$$

$$s = (s_1, s_2, \ldots, s_{N-1})^T = \{s_i,\ 1 \le i \le (N-1)\}$$

$$T_1 : u \ \rightarrow \ T_1 u$$

$$(T_1 u)_i = -\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}\right), \qquad 1 \le i \le (N-1) \quad \text{(internal points)}$$

$$u_0 = u_N = 0$$

Remarks:

- Here we are more concerned with the number of sub-intervals than with the number of internal points. In the multigrid method, the number of sub-intervals is doubled at each new grid level.

- The linear operator T_1 is defined in the difference form, instead of being written out explicitly in the matrix form. The results obtained by working with the difference form can be easily extended to 2-D and 3-D problems where the matrix form is messy and complicated.
- Vector u does not include u_0 or u_N. The zero BCs, $u_0 = 0$ and $u_N = 0$, are added to define linear operator T_1 in the difference form. In this way, linear operator T_1 is independent of the prescribed boundary conditions of BVP. The prescribed boundary conditions are taken care of in another term.

Numerical discretization in the operator-vector form

$$T_1 u = s + \beta$$

where vector β contains the effects of prescribed boundary conditions.

$$\beta = \frac{1}{h^2}\left(c, 0, \cdots, 0, d\right)$$

We study the difference between the exact solution of BVP (what we want to know) and the solution of numerical discretization (what we can calculate).

Formulation of error analysis

Let $v(x)$ be the exact solution of (BVP-1). Let $v = (v_1, v_2, \ldots, v_{N-1})^T$ where $v_i = v(x_i)$.

Vector v is the discrete version of the exact solution.

Definition (truncation error)

The truncation error of the numerical discretization $T_1 u = s + \beta$ is

$$e(h) \equiv T_1 v - (s + \beta).$$

Definition (error in numerical solution)

The error in the numerical solution u is

$$E(h) \equiv v - u .$$

Remarks:

- Here the truncation error is defined in the same way as the local truncation error for a method solving a time evolution equation:

  Truncation error = the residual term when substituting the exact solution into the numerical discretization.

- The truncation error here has already been divided by h^2.
- Since the Poisson equation describes an equilibrium instead of a time evolution, the two errors are named as

Truncation error vs Error in the numerical solution

- For a time evolution PDE, the two errors are names as

  Local truncation error vs Global error

Big picture of error analysis

We can calculate the truncation error, $e(h)$, using Taylor expansion.

We want to know the error in numerical solution, $E(h)$.

We need to connect these two.

$$0 = T_1 u - (s + \beta) \qquad \text{numerical discretization}$$

$$e(h) = T_1 v - (s + \beta) \qquad \text{definition of truncation error}$$

==> $e(h) = T_1 \underbrace{(v - u)}_{E(h)}$

==> $E(h) = T_1^{-1}\left(e(h)\right)$

==> $\|E(h)\| \le \|T_1^{-1}\| \cdot \|e(h)\|$

We need to study operators T_1, T_1^{-1}, and the norm of T_1^{-1}.

Mathematical preparations for error analysis

Theorem (truncation error)

For the numerical discretization $T_1 u = s + \beta$ described above, we have

$$e(h) = O(h^2).$$

Proof: Use Taylor expansion.

Theorem (T_1 is self-adjoint)

The linear operator T_1 is self-adjoint (as a matrix, it is real and symmetric).

Specifically, it satisfies

$$\langle v, T_1 u \rangle = \langle T_1 v, u \rangle \qquad \text{for all vectors } u \text{ and } v.$$

Proof:

From the definition of T_1, We extend u and v with $v_0 = v_N = u_0 = u_N$. We have

$$\langle v, T_1 u \rangle = -\sum_{i=1}^{N-1} v_i \left(\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} \right), \qquad u_0 = u_N = 0$$

$$= \frac{-1}{h^2}\sum_{i=1}^{N-1}\left(v_i u_{i-1} - 2 v_i u_i + v_i u_{i+1}\right)$$

Using $\sum_{i=1}^{N-1} v_i u_{i-1} = \sum_{k=0}^{N-2} v_{k+1} u_k = \sum_{k=1}^{N-1} v_{k+1} u_k$

$$= \frac{-1}{h^2}\sum_{i=1}^{N-1}\left(v_{i+1} u_i - 2 v_i u_i + v_{i-1} u_i\right), \qquad v_0 = v_N = 0$$

$$= -\sum_{i=1}^{N-1}\left(\frac{v_{i-1} - 2v_i + v_{i+1}}{h^2}\right) u_i = \left\langle T_1 v, u \right\rangle$$

Theorem (Eigenvalues and eigenvectors of T_1)

The eigenvalues of T_1 are

$$\lambda^{(n)} = \frac{2}{h^2}\left[1 - \cos\left(nh\frac{\pi}{L}\right)\right] = \frac{4}{h^2}\sin^2\left(\frac{n\pi}{2N}\right), \qquad n = 1, 2, \ldots, N-1$$

The corresponding eigenvectors are

$$w^{(n)} = \left(w_1^{(n)}, w_2^{(n)}, \ldots, w_{N-1}^{(n)}\right)^T, \quad w_i^{(n)} = \sin\left(\frac{in\pi}{N}\right), \quad n = 1, 2, \ldots, N-1$$

Proof: You verified this in one of your homework assignments.

Vector norm and matrix norm (a very brief review)

The 2-norm of vector $u = \left(u_1, u_2, \ldots, u_{N-1}\right)^T$ is defined as

$$\|u\|_2 = \langle u, u\rangle^{\frac{1}{2}} = \left(\sum_{i=1}^{N-1} u_i^{\,2}\right)^{\frac{1}{2}}$$

The 2-norm of matrix A is a derived norm, defined as

$$\|A\|_2 = \max_{u \neq 0} \frac{\|Au\|_2}{\|u\|_2}$$

From the definition of derived norm, it follows that

$$\|Au\|_2 \le \|A\|_2 \|u\|_2 \qquad \text{for all } u$$

$$\|ABu\|_2 \le \|A\|_2 \|Bu\|_2 \le \left(\|A\|_2 \|B\|_2\right)\|u\|_2 \qquad \text{for all } u$$

$$\|AB\|_2 \le \|A\|_2 \|B\|_2$$

The 2-norm of matrix A has the general expression

$$\|A\|_2 = \sqrt{\text{The largest eigenvalue of } (A^T A)}$$

When A is symmetric, the 2-norm of matrix A is

$$\|A\|_2 = \text{The largest eigenvalue (in absolute value) of } A$$

Theorem (norm of T_1^{-1})

For the linear operator T_1, we have

$$\|T_1^{-1}\|_2 = \frac{1}{\lambda^{(1)}} \le \frac{L^2}{4}$$

Proof:

T_1 is symmetric and has eigenvalues $\lambda^{(n)}, \quad n = 1, 2, \ldots, N-1$.

==> T_1^{-1} is symmetric and has eigenvalues $\dfrac{1}{\lambda^{(n)}}, \quad n = 1, 2, \ldots, N-1$

==> $\|T_1^{-1}\|_2 = \max\limits_n \left| \dfrac{1}{\lambda^{(n)}} \right| = \dfrac{1}{\min\limits_n |\lambda^{(n)}|} = \dfrac{1}{\lambda^{(1)}}, \qquad \lambda^{(1)} = \dfrac{4}{h^2} \sin^2\left(\dfrac{\pi}{2N} \right)$

For sine function, we have the inequality (see Appendix C for a proof)

$$\sin(x) \ge \frac{2}{\pi} x \quad \text{for } 0 \le x \le \frac{\pi}{2},$$

Using this inequality on $\lambda^{(1)}$ we have

$$\lambda^{(1)} = \frac{4}{h^2} \sin^2\left(\frac{\pi}{2N} \right) \ge \frac{4}{h^2} \left(\frac{2}{\pi} \cdot \frac{\pi}{2N} \right)^2 = \frac{4}{L^2}$$

==> $\|T_1^{-1}\|_2 = \dfrac{1}{\lambda^{(1)}} \le \dfrac{L^2}{4}$

Remark:

The upper bound above is not tight. For large N, asymptotically we have

$$\|T_1^{-1}\|_2 = \frac{1}{\lambda^{(1)}} = \frac{h^2}{4 \sin^2\left(\frac{\pi}{2N} \right)} \approx \frac{h^2}{4 \left(\frac{\pi}{2N} \right)^2} = \frac{L^2}{\pi^2}.$$

Main conclusion of error analysis

Theorem (bound on the error)

For the numerical discretization $T_1 u = s + \beta$ described above, we have

$$\|E(h)\|_2 \le \frac{L^2}{4}\|e(h)\|_2$$

Proof:

The two errors $e(h)$ and $E(h)$ are related as follows.

$$0 = T_1 u - (s+\beta) \qquad \text{numerical discretization}$$

$$e(h) = T_1 v - (s+\beta) \qquad \text{definition of truncation error}$$

==> $\quad e(h) = T_1 \underbrace{(v-u)}_{E(h)}$

==> $\quad E(h) = T_1^{-1}\left(e(h)\right)$

==> $\quad \|E(h)\|_2 = \|T_1^{-1} e(h)\|_2 \le \|T_1^{-1}\|_2 \|e(h)\|_2 \le \dfrac{L^2}{4}\|e(h)\|_2$

Note:

This theorem tells us that the difference between the exact solution and the numerical solution decreases as $O(h^2)$.

Appendix A

General theory of numerical methods for solving $u_t + F(u)_x = 0$

Definition: (Total variation)

The total variation of a discrete function $u^n = \{u_i^n\}$ is defined as

$$\text{TV}(u^n) \equiv \sum_{i=-\infty}^{+\infty} \left|u_{i+1}^n - u_i^n\right|$$

Definition: (Total variation diminishing method, TVD)

A numerical method is called total variation diminishing (TVD) if it satisfies

$$\text{TV}(u^{n+1}) \le \text{TV}(u^n)$$

Definition: (Monotone method)

A numerical method is called monotone if

$$u_i^n \ge v_i^n \text{ for all } i \qquad \text{implies} \qquad u_i^{n+1} \ge v_i^{n+1} \text{ for all } i$$

Definition: (Monotonicity preserving method)

A numerical method is called monotonicity preserving if

$$u_{i+1}^{n} \geq u_{i}^{n} \text{ for all } i \quad \text{implies} \quad u_{i+1}^{n+1} \geq u_{i}^{n+1} \text{ for all } i$$

$$\text{and} \quad u_{i+1}^{n} \leq u_{i}^{n} \text{ for all } i \quad \text{implies} \quad u_{i+1}^{n+1} \leq u_{i}^{n+1} \text{ for all } i\,.$$

Theorem:

- A monotone method must be total variation diminishing.

  That is, monotone ==> TVD
- A total variation diminishing method must be monotonicity preserving.

  That is, TVD ==> monotonicity preserving

Proof: skipped

Definition: (Linear method for conservation laws)

If a numerical method, when applied to linear PDE $u_t + a\, u_x = 0$, is linear, then it is called a linear method for solving conservation laws.

Theorem:

For a linear method, monotone, TVD and monotonicity preserving are equivalent.

Monotone <==> TVD <==> monotonicity preserving

Proof: skipped

Here is the main result about linear methods.

Theorem:

Consider a linear method for solving $u_t + F(u)_x = 0$.

If it is monotone, then its accuracy is limited to the first order.

Proof: skipped

Remark:

A linear method that is non-oscillating is limited to the first order.

A high resolution method must be intrinsically non-linear.

Appendix B

We derive $\exp(A\frac{\Delta t}{2})\exp(B\Delta t)\exp(A\frac{\Delta t}{2}) = \exp\left((A+B)\Delta t\right) + O\left((\Delta t)^3\right)$.

$$
\begin{aligned}
\exp(A\frac{\Delta t}{2})\exp(B\Delta t)\exp(A\frac{\Delta t}{2}) &= \left(I + \frac{1}{2}A\Delta t + \frac{1}{8}A^2(\Delta t)^2 \right) \\
&\quad \times \left(I + B\Delta t + \frac{1}{2}B^2(\Delta t)^2 \right)\left(I + \frac{1}{2}A\Delta t + \frac{1}{8}A^2(\Delta t)^2 \right) + O\left((\Delta t)^3\right) \\
&= \left(I + \frac{1}{2}A\Delta t + \frac{1}{8}A^2(\Delta t)^2 \right)\left(I + \left(\frac{1}{2}A + B\right)\Delta t + \frac{1}{2}\left(\frac{A^2}{4} + BA + B^2\right)(\Delta t)^2 \right) + O\left((\Delta t)^3\right) \\
&= \left(I + (A+B)\Delta t + \frac{1}{2}(A^2 + AB + BA + B^2)(\Delta t)^2 \right) + O\left((\Delta t)^3\right) \\
&= \exp\left((A+B)\Delta t\right) + O\left((\Delta t)^3\right)
\end{aligned}
$$

Appendix C

We prove the inequality

$$\sin(x) \ge \frac{2}{\pi}x \quad \text{for } 0 \le x \le \frac{\pi}{2},$$

Proof:

Let $f(x) \equiv \sin(x) - \frac{2}{\pi}x$.

We have $f''(x) = -\sin(x) < 0$ in $(0, \pi/2)$, that is, $f(x)$ is concave down in $(0, \pi/2)$.

Since $f(0) = 0$ and $f(\pi/2) = 0$, we conclude $f(x) \ge 0$ in $[0, \pi/2]$.