

Lecture Notes for “Numerical Analysis: Differential
Equations”

M 387D - CSE 383L, Spring 2023

May 5, 2023

Contents

1	Jan 9: Introduction	5
2	Jan 11: Numerical Ordinary Differential Equations (Part I: General theory and LTE)	9
3	Jan 18: Numerical Ordinary Differential Equations (Part II: R-K and LMM)	15
4	Jan 23: Numerical Ordinary Differential Equations (Part III: LMM)	21
5	Jan 25: Numerical Ordinary Differential Equations (Part IV: Stability)	27
6	Jan 30: Numerical Ordinary Differential Equations (Part V: Stiff and symplectic systems)	31
7	Feb 6: Numerical Ordinary Differential Equations (Part VI: Miscellaneous remarks and DAE)	35
8	Feb 8: Numerical Ordinary Differential Equations (Part VII: SDE and BVP)	41
9	Feb 13: Numerical Partial Differential Equations (Part I: 2-Point BVP, FDM, and FEM)	47
10	Feb 15: Numerical Partial Differential Equations (Part II: FEM)	53
11	Feb 20: Numerical Partial Differential Equations (Part III: FEM and Poincare's inequality)	57
12	Feb 22: Numerical Partial Differential Equations (Part IV: Boundary Conditions for BVPs)	63

13 Feb 27: Numerical Partial Differential Equations (Part V: Inhomogeneous BC and Higher Order Problems)	67
14 Mar 1: Numerical Partial Differential Equations (Part VI: FEM for PDEs, Poisson Equation)	71
15 Mar 6: Numerical Partial Differential Equations (Part VII: Practical Concerns for FEM)	77
16 Mar 8: Numerical Partial Differential Equations (Part VIII: Remarks on FEM and Parabolic Problems)	83
17 Mar 20: Numerical Partial Differential Equations (Part IX: Time Dependent Problems and Mixed Methods)	87
18 Mar 22: Numerical Partial Differential Equations (Part X: Mixed Methods and FDM)	93
19 Mar 27: Numerical Partial Differential Equations (Part XI: FDM & Stability Analysis)	99
20 Mar 29: Numerical Partial Differential Equations (Part XII: Stability Analysis)	103
21 Apr 3: Numerical Partial Differential Equations (Part XIII: von Neumann Analysis)	107
22 Apr 5: Numerical Partial Differential Equations (Part XIV: Topics on Non-linear Problems)	113
23 Apr 10: Numerical Partial Differential Equations (Part XV: NLCL and FVM)	117
24 Apr 12: Numerical Partial Differential Equations (Part XVI: FVM and DG)	121
25 Apr 17: Numerical Partial Differential Equations (Part XVII: DG and Particle Methods)	125
26 Apr 24: Numerical Partial Differential Equations (Part XVIII: Spectral Methods)	129
A Neural Operators and PDEs	133

Jan 9: Lecture 1

Introduction

1.1 A brief overview

To put in a general sense, the differential equation of interest can be written in the form of

$$F(u, x, \partial_x) = 0 \tag{1.1.1}$$

where F is a linear or non-linear function. The prerequisite is to establish well-posedness of the solution, composed of the following three components:

- Existence;
- Uniqueness, possibly under an additional metric if intrinsically non-unique;
- Continuous dependency on data, usually coupled with the approximation property of the space of numerical solutions.

We put the methodology of numerical analysis in a four-step paradigm, namely

1. The original process in the real world;
2. A mathematical modeling that yields Eqn. 1.1.1, often an infinite-dimensional object;
3. The numerical algorithm that reduces the infinite-dimensional problem to a finite-dimensional one;
4. A computer code or solver that handles the reduced problem.

This course is focused on the numerical reduction part (step 2 to 3) with some discussion on the implementation (step 3 to 4):

1.2 Reducing the infinite-dimensional problem

There are two important ingredients that contribute to a working numerical solve: a representation/approximation scheme of the solution function u and principles for discretization of the differential equation.

1.2.1 Representation of solution

- Point-wise representation $\{u_j\}$, namely collecting the evaluations of the given function u at some samples points $\{x_j\}$, i.e. $u_j := u(x_j)$; mainly used in FDM, spectral methods.
- Sum of known functions φ_j (or bases), in the sense that

$$\tilde{u}(x) = \sum_j \alpha_j \varphi_j(x); \quad (1.2.1)$$

used in FEM, DG, spectral methods.

- Local averages, mainly deployed in FVM, where the average value within each cell represents the neighborhood.
- Particle distribution, where the function to represent is supposed to be the p.d.f. of the ensemble of particles being tracked.
- ANN, where weights and biases are the unknowns to be determined in the context of networks with a given type of topology

1.2.2 Principles for discretization

- Finite difference method (or FDM for short): by definition,

$$\frac{d u(x)}{d x} := \lim_{h \rightarrow 0} \frac{u(x+h) - u(x)}{h}.$$

Thus, on a evenly-space grid x_j where $x_{j+1} - x_j = h$, the finite difference can be a good approximation to the differential, i.e.

$$\left. \frac{d u(x)}{d x} \right|_{x=x_j} \approx \frac{u(x_j+h) - u(x_j)}{h} = \frac{u_{j+1} - u_j}{h}.$$

Variants are developed for better precision and other concerns.

- Finite element method (FEM): this method is usually based variational forms that are equivalent to the differential equation of interest, often known as weak forms. Then, the solution

is approximated by sum of known functions which leads to a system (Eqn. 1.2.1) of algebraic equations of the unknowns $\{\alpha_j\}$. Notice that this function approximator is related to the collocation method where the system of equations $F(\tilde{u}, x_j, \partial_x) = 0$ is solved on a few sample points $\{x_j\}$, but it shall not be confused with the finite element method. A key distinct lies in that, instead of forcing the residual term to be 0, the Galerkin method aims to make the residual $F(\tilde{u}, x, \partial_x)$ orthogonal to the basis functions $\{\varphi_j\}$.

- Finite volume method (FVM): often applied to difference equations arising from conservation laws, e.g.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} f(u). \quad (1.2.2)$$

To apply the FVM, we integrate Eqn. 1.2.2 over a small rectangular area $[x_j, x_{j+1}] \times [t_k, t_{k+1}]$ that yields an equation connecting the local averages and the fluxes on the sides.

- Discontinuous Galerkin (DG): almost identical to FEM, except for allowing discontinuities in $\{\varphi_j\}$ which is handled by similar techniques borrowed from FVM.
- Spectral methods: the motivation can be illustrated via the following observation

$$\frac{d u}{d x} = \mathcal{F}^{-1} [(i\omega) \mathcal{F} u(x)] \quad (1.2.3)$$

where \mathcal{F} stands for the Fourier transform and ω denotes the dual variable. Eqn. 1.2.3 provides a possibility to solve some differential equations fast and reliably where \mathcal{F} is usually approximated by the fast Fourier transform (FFT). Be aware that the spectral method relies on periodic functions over the underlying domain, which is not always easily accessible.

- Particle methods: usually applied to differential equations representing transport, e.g.

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \quad (1.2.4)$$

with the initial condition $u(x, 0) = u_0(x)$. Eqn. 1.2.4 can be solved via the method of characteristics that yields $u(x, t) = u_0(x + t)$. An application, connecting to daily traffic, can be set up in the way that $u_0(x) = \text{Heaviside}(x - a)$, representing an accident at location a that has just been resolved. The solution, in the style of a traveling wave, reads $u(x, t) = \text{Heaviside}(x + t - a)$, indicating that the wave front has moved to $a - t$. The particle method is suitable for solving the equation as well as visualizing how the vehicles react the accident in this case.

- Machine learning based methods: apart from physics-informed neural networks (PINNs) that has been introduced above, the other prevailing idea is to learning the solution operator that maps the coefficient tensor to the solution, in the discretized form or other representations.

Jan 11: Lecture 2

Numerical Ordinary Differential Equations (Part I: General theory and LTE)

2.1 General theory

2.1.1 Autonomous first-order form

A linear system of ODE possess the form $\frac{du(t)}{dt} = f(u)$ where $u = (u^{(1)}, \dots, u^{(k)})^T : \mathbb{R}^1 \rightarrow \mathbb{R}^k$ is a vector collecting different components. The solution is, however, not unique in general if without additional conditions which can be classified by the number of temporal occurrences:

- Initial value problem (IVP), i.e. extra condition at one t -value;
- Boundary value problem (BVP), i.e. extra condition at two (possibly more) t -values.

We are interested in the autonomous first-order form

$$\begin{cases} u' = f(u) & t > t_0, \\ u(t_0) = u_0 \end{cases} \quad (2.1.1)$$

since it is usually the most general form of an linear IVP that is possibly of higher orders or is time-dependent. For example, a system

$$\begin{cases} u'' = f(u', u, t) \\ u(t_0) = u_0 \\ u'(t_0) = u_{0,1} \end{cases}$$

can be reduced into the form (Eqn. 2.1.1):

$$\begin{cases} (u^{(1)})' = u^{(2)} \\ (u^{(2)})' = f(u^{(2)}, u^{(1)}, u^{(3)}) \\ (u^{(3)})' = 1 \end{cases}, \begin{cases} (u^{(1)})(0) = u_0 \\ (u^{(2)})(0) = u_{0,1} \\ (u^{(3)})(0) = t_0 \end{cases}$$

via identifying $u^{(1)} := u, u^{(2)} := u', u^{(3)} := t$.

2.1.2 Well-posedness

The well-posedness is guaranteed by the Picard-Lindberg Theorem:

Theorem 2.1.1. *There exists a unique solution $u \in C^1(t_0, \infty)$ to Eqn. 2.1.1 if f is Lipschitz continuous, i.e. $\exists L > 0$ s.t. $|f(u) - f(v)| \leq L|u - v|, \forall u, v \in \mathbb{R}^k$.*

Remark. The proof is based on the Picard iteration

$$u^{[k]}(t) = u(t_0) + \int_{t_0}^t f(u^{[k-1]}(t)) dt.$$

Remark. Local existence only relies on $f \in C(\mathbb{R}^k)$; nevertheless, it does not guarantee uniqueness since bifurcation might occur. For example,

$$\begin{cases} u' = 2\sqrt{u} & t > 0, \\ u(0) = 0 \end{cases}$$

admits solutions $u = 0$ and $\tilde{u} = t^2$, due to \sqrt{u} being not Lipschitz continuous at $u = 0$.

Remark. The solution might not exist globally if f is Lipschitz but the Lipschitz constant is not uniformly bounded. For example,

$$\begin{cases} u' = u^2 & t > 0, \\ u(0) = 1 \end{cases}$$

admits solution $u = \frac{1}{1-t}$ which can not be extended beyond $t = 1$.

We also wish to establish the continuous dependence on data which is a proof model for stability of other numerical methods.

Proposition 2.1.1. *Given a Lipschitz function f with Lipschitz constant L and initial perturbation $\delta \in \mathbb{R}^k$. For the following two IVPs*

$$\begin{cases} u' = f(u) \\ u(t_0) = u_0 \end{cases}, \begin{cases} v' = f(v) \\ v(t_0) = u_0 + \delta \end{cases},$$

the difference is estimated by

$$\|u(t) - v(t)\| \leq e^{L(t-t_0)} \|\delta\|.$$

Proof. Consider the error function $e(t) := v(t) - u(t)$ which satisfies the integral equation

$$e(t) = v(t) - u(t) = \delta + \int_{t_0}^t [f(v) - f(u)] \, ds.$$

Since f is Lipschitz continuous, we have

$$\|e(t)\| \leq \|\delta\| + \int_{t_0}^t L \|v(s) - u(s)\| \, ds = \|\delta\| + \int_{t_0}^t L \|e(s)\| \, ds$$

Thus, by the Gronwall's inequality in the integral form,

$$\|e(t)\| \leq e^{L(t-t_0)} \|e(t_0)\| = e^{L(t-t_0)} \|\delta\|$$

□

2.1.3 Applications

- Molecule dynamics
 - u being the physical (and possibly velocity) coordinates;
 - f can be interaction between molecules or external forces;
 - similar formulation for particle interactions or mechanical systems.
- Used in conjunction with methods of characteristics; ODEs from a space discretization of PDEs.
- Rate estimation in the context of chemistry reactions or virus infections.

2.2 Numerical approximation via FDM

2.2.1 A list of common schemes

A first attempt of building a numerical solver starts with an evenly-spaced discretization in time $t_n := t_0 + nh$, $u_n := u(t_n)$ for a fixed time step $h > 0$ (which can be easily extended to variable h). We use finite difference $\frac{u(t+h)-u(t)}{h}$ to approximate the derivative $u'(t)$. This leads to the following numerical scheme

$$\begin{cases} \frac{u_{n+1}-u_n}{h} = f(u_n) \\ u_0 \text{ as given} \end{cases} \quad (2.2.1)$$

which is often known as forward/explicit Euler (or FE for short). A variant to Eqn. 2.2.1 is to adopt u_{n+1} for the input to the source term, leading to the backward/implicit Euler

$$\begin{cases} \frac{u_{n+1}-u_n}{h} = f(u_{n+1}) \\ u_0 \text{ as given} \end{cases} \quad (2.2.2)$$

Another variant uses the central difference

$$\begin{cases} \frac{u_{n+2}-u_n}{2h} = f(u_{n+1}) \\ u_0 \text{ as given} \\ u_1 = u_0 + hf(u_0) \quad (\text{common}) \end{cases} \quad (2.2.3)$$

which is often known as leap frog in the context of methods of lines. The trapezoidal rule can also be applied in the sense that

$$\begin{cases} \frac{u_{n+1}-u_n}{h} = \frac{1}{2} [f(u_{n+1}) + f(u_n)] \\ u_0 \text{ as given} \end{cases} \quad (2.2.4)$$

which is known as Crank-Nicolson in the context of PDEs.

We can classify the aforementioned methods by the explicit/implicit properties:

- Explicit: Euler (Eqn. 2.2.1), Central difference (Eqn. 2.2.3); easier to solve.
- Implicit: Backward Euler (Eqn. 2.2.2), Trapezoidal (Eqn. 2.2.4); harder to solve at each step, but usually lead to higher precision and suitable for stiff systems.

Another classification is by the number of steps involved:

- One-step method: E, IE, T; easier to change step sizes.

- Multi-step method: C; need to handle careful or otherwise might not work, prototype for R-K methods.

Another popular ranking is based on the order of accuracy:

- First-order $\mathcal{O}(h)$: E, IE.
- Second-order $\mathcal{O}(h^2)$: C, T.

2.2.2 Local truncation error

The order of accuracy be shown via the local truncation error (LTE) if the method is stable. To see how LTE can be used to determine the order, we study FE for example. If we plug the true ODE solution u to Eqn. 2.2.1, we shall not expect the discrete equation to hold since the finite difference is an approximation. Thus, we define the LTE as

$$\text{LTE}_n := \frac{u(t_{n+1}) - u(t_n)}{h} - f(u(t_n))$$

By applying the Taylor expansion, we have

$$\text{LTE}_n = \frac{h}{2} u''(\xi) = \mathcal{O}(h), t_n \leq \xi \leq t_{n+1}.$$

Remark. An alternative formulation for FE is

$$u_{n+1} = u_n + hf(u_n) \tag{2.2.5}$$

which is more suitable for coding. An equivalent definition LTE can be based on Eqn. 2.2.5, usually resulting in one higher order of h .

2.2.3 From local error to global error

Proposition 2.2.1. *For Lipschitz f , the error between the numerical solution u_{n+1} solved by Eqn. 2.2.1 and the true solution $u(t_n)$ can be estimated by*

$$\|u(t_n) - u_n\| \leq Ch \max_{t_0 \leq t \leq t_0+T} \|u''(t)\|$$

where $C := \frac{e^{LT} - 1}{2L}$.

Proof. Let $e_n = u(t_n) - u_n$ denote the error. Recall the ODE solution continuously depends on

initial data, leading to

$$\begin{aligned} e_{n+1} &= u(t_{n+1}) - u_{n+1} \\ &= [u(t_n) + hf(u(t_n)) + h\text{LTE}_n] - [u_n + hf(u_n)] \\ &= e_n + h[f(u(t_n)) - f(u_n) + \text{LTE}_n]. \end{aligned}$$

Since f is Lipschitz continuous, we have

$$\|e_{n+1}\| \leq (1 + hL)\|e_n\| + h\|\text{LTE}_n\|.$$

Solving the recursive inequality leads to

$$\|e_n\| \leq (1 + hL)^n \|e_0\| + h \sum_{j=0}^{n-1} (1 + hL)^{n-1-j} \|\text{LTE}_j\|.$$

We estimate the power term by $(1 + hL)^n \leq e^{hLn} = e^{LT}$ if we identify $T := t_n - t_0$; the LTE is bounded by

$$h \sum_{j=0}^{n-1} (1 + hL)^{n-1-j} \|\text{LTE}_j\| \leq \frac{(1 + hL)^n - 1}{L} \max_{0 \leq j \leq n-1} \|\text{LTE}_j\| \leq Ch \max_{t_0 \leq t \leq t_0+T} \|u''(t)\|$$

where $C = \frac{e^{LT} - 1}{2L}$. Recall that $e_0 = u(t_0) - u_0 = 0$, thus

$$\|u(t_n) - u_n\| = \|e_n\| \leq Ch \max_{t_0 \leq t \leq t_0+T} \|u''(t)\|.$$

□

Jan 18: Lecture 3

Numerical Ordinary Differential Equations (Part II: R-K and LMM)

In the last lecture, we derived a few numerical theories for ODEs in the autonomous form $u' = f(u)$. The general idea is to discretize the function u by its value on a mesh $u_n \approx u(t_n)$, $t_n = t_0 + nh$ and use Taylor expansion to derive iterative schemes. We also derive an error estimate for the Euler scheme: the error $e_n = u(t_n) - u_n$ is bounded by $h \max |u''|$ up to a constant, thus we call it a first order method.

In seek for higher orders, one way is to carry out Taylor methods which involves more derivative terms and the practical alternative is to the Runge-Kutta family. Another approach is to study linear multistep methods (LMM).

3.1 Taylor methods

Let us consider the Taylor expansion at time t :

$$u(t+h) = u(t) + hu'(t) + \frac{1}{2}h^2u''(t) + o(h^2). \quad (3.1.1)$$

Since $u(t)$ solves $u' = f(u)$, the first two terms reduces to $u(t) + hf(u(t))$ which recovers the Euler scheme. To handle the third term, we differentiate the ODE to obtain

$$u'' = [f(u)]' = [\nabla f(u)]u' = [(\nabla f)f](u)$$

where the Jacobian is defined as $(\nabla f)_{i,j} := \partial_j f_i$. With this established, Eqn. 3.1.1 yields the following scheme

$$u_{n+1} = u_n + hf(u_n) + \frac{1}{2}h^2 [(\nabla f) f](u_n). \quad (3.1.2)$$

We shall point out that Eqn. 3.1.2 is not widely adopted in practice for the following drawbacks:

- It is often the case that f not given in a closed form but rather obtained from data or loop-up table, let alone ∇f . Even if f has an analytical form, ∇f might be hard to evaluate (and might be costly even with the help of auto-differentiation mechanism).
- Variants with higher orders are possible but potentially even more inefficient than simply decreasing the time step.

3.2 Runge-Kutta methods (R-K)

To overcome the drawbacks of the Taylor method, we aim to replace higher derivatives of f by several evaluations of $f(u)$ in each step.

Example 3.2.1. Recall that the trapezoidal scheme

$$u_{n+1} = u_n + \frac{h}{2} [f(u_{n+1}) + f(u_n)] \quad (3.2.1)$$

leads to a second order method. Although one can adopt Newton's method to solve Eqn. 3.2.1 at each step, it adds additional complexity due to being an implicit method. One way to eliminate the implicitness is to first approximate u_{n+1} by the Euler method which is provided to f later, leading to

$$\begin{aligned} \tilde{u}_{n+1} &= u_n + hf(u_n), \\ u_{n+1} &= u_n + \frac{h}{2} [f(\tilde{u}_{n+1}) + f(u_n)]. \end{aligned} \quad (3.2.2)$$

This is also known as the improved Euler method.

Proposition 3.2.1. *The improved Euler method is of second order.*

Proof. It suffices to verify that the LTE for the true solution $u(t)$ is $\mathcal{O}(h^2)$. Let us introduce shorthand $f_n := f(u(t_n))$, $(\nabla f)_n := \nabla f(u(t_n))$, and $(\nabla^2 f)_n := \nabla^2 f(u(t_n))$ (not to confuse f_n with $f(u_n)$). By Taylor expansion, for $\tilde{u}(t) := u(t) + hf(u(t))$, we have

$$f(\tilde{u}(t_n)) = f_n + h(\nabla f)_n f_n + \frac{h^2}{2} f_n^T (\nabla^2 f)_n f_n + o(h^2),$$

as well as

$$\begin{aligned} u(t_n + h) &= u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + \frac{h^3}{6}u'''(t_n) + o(h^3) \\ &= u(t_n) + hf_n + \frac{h^2}{2}(\nabla f)_n f_n + \frac{h^3}{6} \left[f_n^T (\nabla^2 f)_n f_n + (\nabla f)_n^2 f_n \right] + o(h^3) \end{aligned}$$

Thus,

$$\begin{aligned} \text{LTE} &= \frac{u(t_n + h) - u(t_n)}{h} - \frac{1}{2} [f(\tilde{u}(t_n)) + f(u(t_n))] \\ &= \frac{h^2}{6} \left[f_n^T (\nabla^2 f)_n f_n + (\nabla f)_n^2 f_n + o(1) \right] - \left[\frac{h^2}{4} f_n^T (\nabla^2 f)_n f_n + o(h^2) \right] \\ &= \frac{h^2}{12} \left[2(\nabla f)_n^2 - f_n^T (\nabla^2 f)_n \right] f_n + o(h^2). \end{aligned}$$

□

The improved Euler method is a particular example of the so-called (explicit) Runge-Kutta family, of which the general form reads

$$u_{n+1} = u_n + h \sum_{j=1}^J b_j k_j, \quad (3.2.3)$$

$$k_j = f \left(u_n + h \sum_{i=1}^{j-1} a_{ji} k_i \right), 1 \leq j \leq J. \quad (3.2.4)$$

Eqn. 3.2.3 indicates that the update term is an interpolation of the intermediate terms k_j which shall be close to $f(u_n)$. Eqn. 3.2.4 defines that each intermediate term is an evaluation of f at a slightly updated location of u_n , using the information k_1, \dots, k_{j-1} that is computed before. The coefficients b_j, a_{ji} shall satisfy a certain constraint system to guarantee good properties of the scheme (e.g. accuracy, stability, ...). The R-K family can also be implicit where the most general form reads

$$k_j = f \left(u_n + h \sum_{i=1}^J a_{ji} k_i \right), 1 \leq j \leq J.$$

We can classify the R-K family based on $\{a_{ji}\}$:

- Explicit if $a_{ji} = 0, \forall j < i$;
- Diagonally implicit (DIRK) $a_{ji} = 0, \forall j \leq i$;
- Fully implicit (IRK) if $a_{ji} \neq 0, \forall j, i$.

We show the table on order of accuracy in Tab. 3.2.1. In general, a higher order scheme requires more stages at each step which grows slightly faster than the order of accuracy. We can find the Euler method as a typical first order method, improved Euler method falling in second order category, and classical R-K method

$$\begin{aligned}
 u_{n+1} &= u_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4), \\
 k_1 &= f(u_n), \\
 k_2 &= f\left(u_n + \frac{h}{2}k_1\right), \\
 k_3 &= f\left(u_n + \frac{h}{2}k_2\right), \\
 k_4 &= f(u_n + hk_3)
 \end{aligned}
 \tag{3.2.5}$$

as a fourth order method.

Order of accuracy	1	2	3	4	5	6	7	8
Minimum stages needed	1	2	3	4	6	7	9	11

Table 3.2.1: Optimal order for R-K family methods.

The R-K methods enjoy the nice property that the order of convergence coincides with the order of LTE, due to the fact that the R-K family belongs to 1-step methods which are always stable. The R-K methods are also suitable with adaptive step sizes for better flexibility. We refer to [this article](#) for an introduction to two popular implementations of the R-K family.

3.3 Linear multistep methods (LMM)

LMM can be motivated by the central difference method

$$u_{n+1} = u_{n-1} + 2hf(u_n)$$

which is of second order and is also explicit. The price to pay is that one needs to determine the first term u_1 in order to “bootstrap” the beginning. One typical choice is to adopt the Euler method, i.e. $u_1 = u_0 + hf(u_0)$; nevertheless, detailed analysis is often required to verify if it leads to a well-posed scheme.

The general form for LMM reads

$$\sum_{j=0}^J a_j u_{n+j} = h \sum_{j=0}^J b_j f(u_{n+j}) \tag{3.3.1}$$

where u_0 is given but u_1, \dots, u_{J-1} requires extra initial values (if $J \geq 2$). The analysis for LTE is a prerequisite for accuracy, usually involving Taylor expansion and the ODE equation. The following condition is necessary to guarantee consistency:

$$\sum_{j=0}^J a_j = 0, \sum_{j=0}^J (J-j) a_j + \sum_{j=0}^J b_j = 0.$$

Without loss of generality, we set $a_J = 1$, i.e. normalizing the scheme based on the last term u_{n+J} . The term b_J determines if the scheme is explicit or implicit: $b_J \neq 0$ leads to implicit schemes since u_{n+J} appears on both sides and $b_J = 0$ corresponds to explicit schemes on the other hand.

It is worth pointing out that LMM may not convergence even if LTE is of order $p > 0$; in other words, stability analysis is often required.

Example 3.3.1. The following LMM scheme

$$u_{n+2} + u_{n+1} - 2u_n = 3hf(u_n)$$

has LTE = $\mathcal{O}(h)$ but is not stable. The proof for general cases is left as exercise, but we provide a simple illustration under $f \equiv 0$. The scheme reduces to $u_{n+2} = 2u_n - u_{n+1}$. Let us assume the initial condition reads $u_0 = 0, u_1 = \delta \ll 1$. Then, the first few terms read $u_2 = -\delta, u_3 = 3\delta, u_4 = -5\delta$ and we can expect an exponential growth in n rather than $t = nh$. This “contradicts” our wish for the scheme to be stable, i.e. insensitive to perturbations in the initial values.

A systematic way to analyze stability is provided as follows.

Theorem 3.3.1 (Dahlquist root condition). *Let $\rho(z) = \sum_{j=0}^J a_j z^j$ define a polynomial in the complex plane. Then, Eqn. 3.3.1 is stable iff.*

- $|z_k| \leq 1$ if $\rho(z_k) = 0$, and
- furthermore, $|z_k| < 1$ if $\rho(z_k) = \rho'(z_k) = 0$ (i.e. z_k is a multiple root).

Jan 23: Lecture 4

Numerical Ordinary Differential Equations (Part III: LMM)

4.1 Accuracy and stability of LMM

Recall that we introduced LMM in the last lecture

$$\sum_{j=0}^J a_j u_{n+j} = h \sum_{j=0}^J b_j f_{n+j} \quad (4.1.1)$$

where f_{n+j} is the shorthand notation for $f(u_{n+j})$ and u_0, \dots, u_{J-1} are properly given as initial conditions.

Definition 4.1.1. Let E be the time-remapping operator that increases the input time by h and we define

$$\rho(z) := \sum_{j=0}^J a_j z^j, \sigma(z) := \sum_{j=0}^J b_j z^j.$$

Remark. We rewrite Eqn. 4.1.1 as $\rho(E) u_n = h \sigma(E) f_n$.

The concepts of consistency and stability are crucial since there is a powerful tool that characterizes the stability of numerical schemes. To be specific, the Lax equivalence theorem states that for consistent approximations of well-posed problems, stability is equivalent to convergence.

The consistency is addressed in the following proposition.

Proposition 4.1.1. *The following statements are equivalent:*

- $LTE = \mathcal{O}(h^p)$ for some given $p \geq 1$;

- $\rho(1) = 0$ and $\sum_{j=0}^J j^q a_j = q \sum_{j=0}^J j^{q-1} b_j$ for $q = 1, \dots, p$.

Proof. The order of LTE can be determined, once again, by using the Taylor expansion. For the truth solution,

$$Eu(t) = u(t+h) = e^{hD}u(t) = \left[\sum_{q=0}^p \frac{h^q}{q!} D^q + \mathcal{O}(h^{p+1}) \right] u(t)$$

where D stands for the differential operator. Since f is the derivative of the truth solution u , we have

$$\begin{aligned} h\text{LTE} &= \rho(E)u - h\sigma(E)u' \\ &= \rho(e^{hD})u - h\sigma(e^{hD})Du \\ &= \sum_{j=0}^J a_j e^{jhD}u - \sum_{j=0}^J b_j h e^{jhD}Du \\ &= \left(\sum_{j=0}^J a_j \right) u + \sum_{q=1}^p \sum_{j=0}^J a_j \frac{(jh)^q}{q!} D^q u - \sum_{q=0}^{p-1} \sum_{j=0}^J b_j \frac{h(jh)^q}{q!} D^{q+1} u + \mathcal{O}(h^{p+1}) \\ &= \rho(1)u + \sum_{q=1}^p \left(\sum_{j=0}^J a_j j^q - q \sum_{j=0}^J b_j j^{q-1} \right) \frac{D^q u}{q!} h^q + \mathcal{O}(h^{p+1}). \end{aligned}$$

Thus, to achieve a p -th order in LTE, it is equivalent to put $\rho(1) = 0$ and $0 = \sum_{j=0}^J a_j j^q - q \sum_{j=0}^J b_j j^{q-1}$ for $1 \leq q \leq p$. □

For stability, we have the Dahlquist root condition (Thm. 3.3.1). Here's a quick illustration on an unstable multiple step scheme.

Example 4.1.1. Consider the following two-step scheme

$$u_{n+2} + u_{n+1} - 2u_n = 3hf_n.$$

- Since $\rho(z) = z^2 + z - 2$ and $\sigma(z) \equiv 3$, it follows that $\rho(1) = 0$.
- For $q = 1$, $\sum_{j=0}^2 j a_j = 3 = \sum_{j=0}^2 b_j$, so this scheme has a LTE of (at least) first order.
- For $q = 2$, $\sum_{j=0}^2 j^2 a_j = 5$ while $2 \sum_{j=0}^2 j b_j = 0$, so LTE is of less than second order.
- The roots to ρ read 1 and -2 , so this scheme is **not** stable.

Remark (Dahlquist first barrier theorem). The maximal order of accuracy for stable LMMs is $J+1$ for odd J and $J+2$ for even J . This can be shown by solving the equations in Prop. 4.1.1.

4.2 Practical methods

4.2.1 Adams methods

The general form of Adams method reads

$$u_{n+J} - u_{n+J-1} = h \sum_{j=0}^J b_j f_{n+j}$$

and choose $\{b_j\}$ for the maximal order. This scheme is stable since $\rho(z) = 0$ leads to $z = 1$ being a single root and $z = 0$ being the remaining multiple roots. We list a few popular variants.

- Explicit ($b_J = 0$, Adams-Bashforth method): for example the Euler method $u_{n+1} - u_n = hf_n$ and a second order scheme $u_{n+2} - u_{n+1} = h \left(\frac{3}{2}f_{n+1} - \frac{1}{2}f_n \right)$.
- Implicit ($b_J \neq 0$, Adams-Moulton methods): for example the trapezoidal rule $u_{n+1} - u_n = h \left(\frac{1}{2}f_{n+1} + \frac{1}{2}f_n \right)$ and a second order scheme $u_{n+2} - u_{n+1} = h \left(\frac{5}{12}f_{n+2} + \frac{8}{12}f_{n+1} - \frac{1}{12}f_n \right)$.

4.2.2 Predictor-corrector methods

The idea is to predict a good approximation to u_{n+J} for implicit Adam methods. The predictors can be calculated from explicit Adam methods. For example, let us consider the following coupled scheme:

$$\tilde{u}_{n+J} - u_{n+J-1} = h \sum_{j=0}^{J-1} b_j f(u_j), \quad (\text{A-B})$$

$$u_{n+J} - u_{n+J-1} = h \left[b_J f(\tilde{u}_{n+J}) + \sum_{j=0}^{J-1} b_j f(u_j) \right]. \quad (\text{modified A-M})$$

Although one extra evaluation of f is needed for each step if compared to the explicit Adam methods, P-C methods avoid solving the implicit equations and are thus cheaper than implicit schemes. We point out that there are quite some influences from the R-K family during the old times of development.

Example 4.2.1. The Heun method

$$\begin{aligned} \tilde{u}_{n+1} - u_n &= hf(u_n), \\ u_{n+1} - u_n &= h \left[\frac{1}{2}f(\tilde{u}_{n+1}) + \frac{1}{2}f(u_n) \right] \end{aligned}$$

can be regarded as a predictor-corrector method whose predictor is the forward Euler method, while the corrector is the Crank-Nicolson method.

Remark. The difference between the predictor and corrector can be used to estimate LTE and thus useful in determination of adaptive step sizes.

4.2.3 Backward difference methods (BDF)

In contrast to the Adams methods where ρ is fixed as $z^J - z^{J-1}$, the BDF methods adopt $\sigma(z) = z^J$ and choose $\{a_j\}$ for maximal order.

Example 4.2.2. The implicit Euler scheme $u_{n+1} - u_n = hf_{n+1}$ can be viewed as a first-order BDF method.

Remark. We shall point out that there is no immediate guarantee on stability for BDF methods (since stability depends on ρ), but they are useful in stiff problems.

4.3 Proof on convergence

4.3.1 LMM in one-step form

In preparation of proving convergence, it would be helpful to reduce the form of LMM to one step. Recall that LMM has the general form of

$$u_{n+J} = -a_{J-1}u_{n+J-1} - \cdots - a_0u_n + h \sum_{j=0}^J b_j f_{n+j}$$

and the equivalent matrix form

$$\begin{pmatrix} u_{n+J} \\ u_{n+J-1} \\ u_{n+J-2} \\ \cdots \\ u_{n+1} \end{pmatrix} = \begin{pmatrix} -a_{J-1} & -a_{J-2} & \cdots & -a_1 & -a_0 \\ 1 & 0 & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & 0 \end{pmatrix} \begin{pmatrix} u_{n+J-1} \\ u_{n+J-2} \\ u_{n+J-3} \\ \cdots \\ u_n \end{pmatrix} + h \begin{pmatrix} \sum_{j=0}^J b_j f_{n+j} \\ 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}. \quad (4.3.1)$$

Motivated by this, the intermediate components are concatenated into a long vector $U_n := (u_{n+J-1}^T, \dots, u_n^T)^T$. Then, Eqn. 4.3.1 is equivalent to

$$U_{n+1} = AU_n + hF(U_{n+1}, U_n). \quad (4.3.2)$$

The matrix A is often called as the companion matrix. We point out that any given LMM is uniquely represented by A and F . Furthermore, Eqn. 4.3.2 is also suitable for representing the R-K family. In fact, a R-K scheme reads $u_{n+1} = u_n + h \sum b_j k_j$ where k_j satisfies a system of equations, but one can always write the update in the form of $u_{n+1} = u_n + hF(u_{n+1}, u_n)$. Thus, R-K schemes are a special case of Eqn. 4.3.2 with A set to the identity matrix.

Jan 25: Lecture 5

Numerical Ordinary Differential Equations (Part IV: Stability)

5.1 Proof on stability

We are interested in analyzing the stability of the following LMM scheme

$$\begin{aligned}U_{n+1} &= AU_n + hF(U_{n+1}, U_n), \\U_0 &\text{ as given,}\end{aligned}\tag{5.1.1}$$

i.e. to measure the difference between U_n and the solution to

$$\begin{aligned}V_{n+1} &= AV_n + hF(V_{n+1}, V_n) + h\delta_{n+1}, \\V_0 &= U_0 + \delta_0,\end{aligned}$$

under perturbations. The stability statement follows a few critical assumptions.

Theorem 5.1.1. *Assuming*

- *the absolute value of each eigenvalue of A does not exceed 1 and is strictly smaller than 1 if being a multiple root, and*
- *F is a Lipschitz function.*

Then, the LMM scheme (Eqn. 5.1.1) is stable in the sense that there exists $h_0 > 0$ and $C > 0$

(depending on T and F) s.t. $0 < h \leq h_0$ implies

$$|U_n - V_n| \leq C \max_{0 \leq m \leq n} |\delta_m|.$$

We need some facts from linear algebra as prerequisite.

Lemma 5.1.1. *All finite dimensional matrix norms are equivalent, i.e. for any two given matrix norms $\|\cdot\|_I$ and $\|\cdot\|_{II}$, there exists $C_1, C_2 > 0$ s.t.*

$$C_1 \|M\|_I \leq \|M\|_{II} \leq C_2 \|M\|_I, \forall M.$$

Lemma 5.1.2. *Any matrix M can be similarity-transformed to Jordan canonical form (JCF), i.e. there exists invertible S s.t. $SMS^{-1} = J$ where J admits a block diagonal form*

$$J = \text{diag}(J_1, J_2, \dots, J_K), J_k = \begin{pmatrix} \lambda_k & 1 & 0 & \cdots & 0 \\ & \lambda_k & 1 & \cdots & 0 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & 1 \\ & & & & \lambda_k \end{pmatrix}.$$

Lemma 5.1.3. *For a given vector norm $\|x\|$, the induced matrix norm is defined as*

$$\|M\| := \sup_{x \neq 0} \frac{\|Mx\|}{\|x\|}.$$

Then, a similarity-transform maps an induced matrix norm to another induced norm, i.e.

$$\|M\|_I := \|SMS^{-1}\|_{II}$$

defines an induced matrix norm $\|\cdot\|_I$ if $\|\cdot\|_{II}$ is an induced norm.

to Thm. 5.1.1. First, we show that there exists an induced norm $\|\cdot\|_*$ s.t. $\|A\|_* \leq 1$. In fact, define

$$B := DSAS^{-1}D^{-1}, \|A\|_* := \|B\|_1$$

where S is the similarity matrix that transforms A to the corresponding JCF and $D := \text{diag}(1, d^{-1}, d^{-2}, \dots)$ is a (scalar-)diagonal matrix with $d > 0$. Recall that SAS^{-1} , as a JCF, admits a block diagonal form $\text{diag}(\widehat{J}_1, \dots, \widehat{J}_K)$ where \widehat{J}_k is a upper-bidiagonal matrix with diagonal entries filled by λ_k .

Thus, B also enjoys a block diagonal form

$$B = D (SAS^{-1}) D^{-1} = \text{diag}(R_1, \dots, R_K), R_k := \begin{pmatrix} \lambda_k & d & 0 & \cdots & 0 \\ & \lambda_k & d & \cdots & 0 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & d \\ & & & & \lambda_k \end{pmatrix}.$$

Recall that the induced 1-norm is the maximal sum of absolute values by rows, thus $\|A\|_* = \|B\|_1 \leq 1$ if

$$\|R_k\|_1 \leq 1, \forall k \iff d \leq 1 - \max_{\mu \text{ multiple root}} |\mu|.$$

Then we move on to the error estimate. Let $E_n := V_n - U_n$ which satisfies

$$\begin{aligned} E_{n+1} &= AE_n + h [F(V_{n+1}, V_n) - F(U_{n+1}, U_n)] + h\delta_n, \\ E_0 &= \delta_0. \end{aligned} \tag{5.1.2}$$

We apply the fact that F is Lipschitz to Eqn. 5.1.2:

$$\|E_{n+1}\|_* \leq \|A\|_* \|E_n\|_* + hL (\|E_{n+1}\|_* + \|E_n\|_*) + h\|\delta_{n+1}\|_*. \tag{5.1.3}$$

where L stands for the Lipschitz constant under the vector norm $\|\cdot\|_*$ ¹ (recall Lem. 5.1.3). We pick $h_0 > 0$ s.t. $h \leq h_0$ implies

$$\left| \frac{1+Lh}{1-Lh} \right| \leq 1 + 3Lh \text{ and } \frac{1}{1-Lh} \leq 2.$$

Let $e_n := \|E_n\|_*$ and $\Delta_n := \|\delta_n\|_*$; then Eqn. 5.1.3 reduces to $e_{n+1} \leq e^{3hL}e_n + 2h\Delta_{n+1}$ with $e_0 = \Delta_0$, leading to

$$e_n \leq e^{3hnL}e_0 + 2h \sum_{m=1}^n e^{3hL(n-m)}\Delta_m \leq (1+2T)e^{3TL} \max_{0 \leq m \leq n} \Delta_m.$$

for $T := nh$. □

Remark. Recall that the matrix $B := DSAS^{-1}D^{-1}$ is introduced to prove $\|A\|_* \leq 1$. The vector norm that induces $\|\cdot\|_*$ reads

$$\|x\|_* := \|(DS)x\|_1.$$

¹which is made rigorous by $\|(\cdot, \cdot)\|_{* \otimes * } := \|\cdot\|_* + \|\cdot\|_*$

This can be verified by a direct calculation

$$\sup_{x \neq 0} \frac{\|Ax\|_*}{\|x\|_*} = \sup_{x \neq 0} \frac{\|DSAx\|_1}{\|DSx\|_1} = \sup_{y:=DSx \neq 0} \frac{\|By\|_1}{\|y\|_1} = \|B\|_1 = \|A\|_*.$$

Corollary 5.1.1. *The stability result shown in Thm. 5.1.1 applies to any R-K method or any LMM that satisfies the Dahlquist root condition, assuming the source terms in the ODE are Lipschitz continuous.*

Remark. For R-K methods, $U_n = u_n$ and thus $A = I$; the Lipschitz continuity of F follows from the Lipschitz continuity of f and the property that sums and compositions preserves Lipschitz continuity, so the assumptions are naturally satisfied.

For LMM, the assumption on F can be verified similarly since F is a linear combination of f . The spectrum condition on A , however, depends on the roots of $\sigma(z)$ under scrutiny. This is to be continued in the next lecture.

Jan 30: Lecture 6

Numerical Ordinary Differential Equations (Part V: Stiff and symplectic systems)

6.1 Stiff problems

Sometimes, we encounter problems with very different time scales, for example in mechanics or chemistry reactions. Let us consider the following example:

$$\begin{aligned} u' &= 100(u_1 - u) & t > 0, \\ u(0) &= u_0, \end{aligned}$$

The solution has a closed form $u(t) = u_1 - (u_1 - u_0) \exp(-100t)$ where $u(t)$ converges rapidly to u_1 regardless of the initial value after a transient layer on the scale of $1/100$. Let us consider the two simplest schemes, namely forward and backward Euler. Both schemes are stable with local truncation error $h^2 u''(\xi)$ that converges to 0 as $h \rightarrow 0$. However, their long-term behavior is quite different:

- Forward Euler: let us put $h = 0.1$, then $u_0 = 0, u_1 = u_0 + h \cdot 100(1 - u_0) = 10, u_2 = -80, \dots$ which explodes quickly.
- Backward Euler: $u_0 = 0, u_1 = 0.91, u_2 = 0.99 \dots$ that is much stabler.

This motivates us to propose a new sense of stability that is useful for stiff problems.

Definition 6.1.1. For model problem $u' = \lambda u$ and fixed h , let us introduce the region of absolute stability R_a

$$R_a := \{z = h\lambda : |u_n| \searrow 0 \text{ as } n \rightarrow \infty\}.$$

Notice that λ can be a complex number.

Example 6.1.1. For Forward Euler, $u_{n+1} = u_n + h\lambda u_n = (1 + h\lambda) u_n$, thus

$$R_a^{\text{FE}} = \{z = h\lambda : |1 + z| < 1\}.$$

R_a^{FE} is a circle that is located to the left of the imaginary axis (as shown in Fig. 6.1.1). Note that for the analytical solution, $R_a^{\text{analytical}} = \{\text{Re}\lambda < 0\}$. For Implicit Euler, $u_{n+1} = u_n + h\lambda u_{n+1}$, giving $u_{n+1} = u_n / |1 + h\lambda|$, thus

$$R_a^{\text{IE}} = \{z = h\lambda : |1 - z| > 1\}.$$

The region R_a^{IE} includes complex numbers with positive real parts, implying that the implicit Euler has a damping effect, i.e. the numerical solution may diminish while the amplitude of true solution grows in time. For the trapezoidal rule, the region reads

$$R_a^{\text{T}} = \left\{ z = h\lambda : \left| \frac{1 + z/2}{1 - z/2} \right| < 1 \right\}.$$

which is exactly the same as the analytical solution.

Remark (R_a for some common methods). • Explicit RK: slightly larger than explicit Euler

- BDF: stable for super-stiff problems, but not A-stable for high-order variants since part of the imaginary axis is excluded; however, it is not a concerning issue if the eigenvalues are guaranteed to be real numbers. We refer to the book “Numerical Mathematics” (page 511) for more details.

Definition 6.1.2. A-stability refers to the property that

$$R_a \supset \{z = h\lambda : \text{Re}z < 0\}.$$

i.e. the stability region contains the left-half plane.

Fact. A-stable LMM is at most of second order. Besides, A-stable methods must be implicit.

Remark. Discretization of PDEs usually leads to stiff problems. In practice, lower order implicit methods (e.g. Trapezoidal or IE) are preferred.

Remark. The general approach to determine stability for ODE system $u' = f(u)$ is to consider eigenvalues z_j of ∇f and to make sure that they fall in the region of absolute convergence R_a .

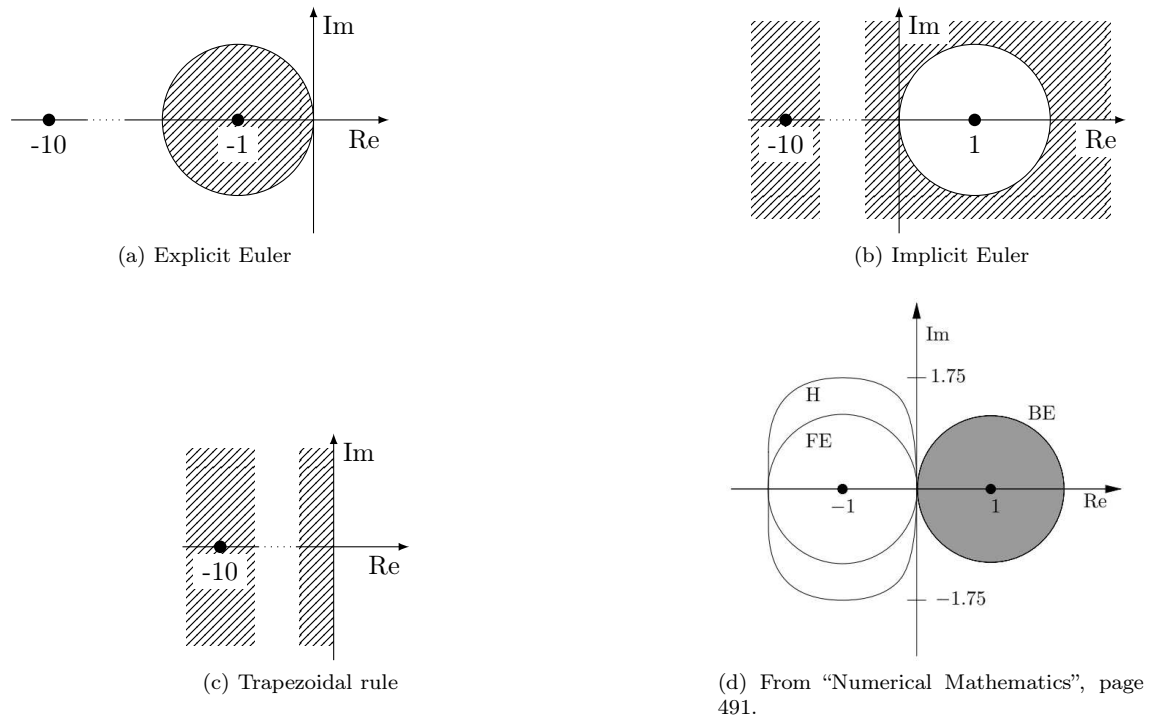


Figure 6.1.1: Region of absolute stability.

Remark. When solving the implicit scheme, the Newton’s method can be used to replace fixed point iterations. For example, for implicit Euler $u_{n+1} = u_n + hf(u_{n+1})$, fixed point iteration doesn’t work if $h\lambda$ is large; Newton’s method, on the other hand, does not suffer from large $h\lambda$, if the initial guess is close enough. The later assumption can be satisfied if we start with a good approximation by explicit schemes or using a smaller intermediate time step.

6.2 Symplectic systems

Another class of ODE systems has the so-called symplectic property. A common characteristic is conservation of energy, e.g. in astrophysics or of the abstract form $\frac{d}{dt}H(u, u') = 0$. This is generalized as the concept of Hamiltonian systems. Loosely speaking, symplectomorphism, i.e. a symplectic map, preserves the “volume” of an area of interest. In the astrophysics example, the planets that stay close will orbit around the sun with separating apart.

To illustrate this, we need a model problem that has imaginary eigenvalues. For example, we

consider

$$u' = i\omega u$$

where $\omega \in \mathbb{R}$ that typically refers to an angular velocity. The solution reads $u(t) = u(0) \exp(i\omega t)$, so that the norm of the solution stays constant. We wish to derive numerical schemes that are symplectomorphism. We exam the common choices:

- Forward Euler: recall that $u_{n+1} = (1 + h\lambda) u_n = (1 + h\lambda)^{n+1} u_0$. However, it holds true that $|1 + ih\omega| > 1$ whenever $h\omega \neq 0$, so the norm of u_n will always grow.
- Implicit Euler: similarly, since $1/|1 - ih\omega| < 1$ for $h\omega \neq 0$, so the norm diminishes over time.
- Trapezoidal rule: since the step size h and ω are real numbers,

$$\left| \frac{1 + ih\omega/2}{1 - ih\omega/2} \right| = 1$$

and thus the norm is kept constant. Notice that it is actually not a symplectic integrator; one can show that the numerical solution solves $u' = i\tilde{\omega}(\omega, h) u$ where the modified angular velocity depends on ω and h at the same time. Thus, it is possible that some eigen components evolve at a different speed than the other ones.

Symplectic integrators are widely used in Hamiltonian systems, especially in molecular dynamics. A second order example is the Verlet integrator: motivated from Newton's second law, let us assume unit mass and the governing equations read

$$\begin{aligned} x'(t) &= v(t), \\ v'(t) &= f(x(t)). \end{aligned}$$

Then the Verlet integrator reads

$$\begin{aligned} v_{n+1/2} &= v_n + \frac{h}{2} f(x_n), \\ x_{n+1} &= x_n + h v_{n+1/2}, \\ v_{n+1} &= v_{n+1/2} + \frac{h}{2} f(x_{n+1}). \end{aligned}$$

See [this file](#) for further discussion.

Feb 6: Lecture 7

Numerical Ordinary Differential Equations (Part VI: Miscellaneous remarks and DAE)

7.1 Miscellaneous remarks

7.1.1 Methods with large absolute stability region

If we are interested in methods with a large stability region rather than the precision, there are the so-called Chebyshev methods. We refer to "[Fourth order chebyshev methods with recurrence relation](#)" for more details.

7.1.2 Stability in higher dimension settings

We have derived the stability result for R-K & LMM with root condition. Assuming f Lipschitz, there exists h_0 s.t. for $h < h_0$, the difference between solution to

$$\begin{cases} U_{n+1} = AU_n + hF(U_{n+1}, U_n) \\ U_0 \text{ as given} \end{cases}, \begin{cases} V_{n+1} = AV_n + hF(V_{n+1}, V_n) + h\delta_{n+1} \\ V_0 = U_0 + \delta_0 \end{cases} \quad (7.1.1)$$

can be estimated by $\|V_n - U_n\| \leq C(T) \max \|\delta_n\|$. A direct consequence is convergence of the numerical scheme if the maximal LTE vanishes as the time step h goes to 0; furthermore, the order of convergence (w.r.t. h) coincides with the order of LTE. In fact, we can view V_n as the true solution and U_n as the numerical solution in Eqn. 7.1.1; then, δ_{n+1} is exactly the LTE. The initial

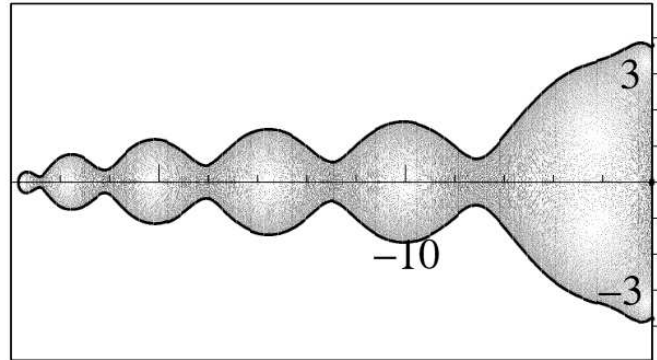


Figure 7.1.1: Stability domain of $R_9(x)$, from "Fourth order chebyshev methods with recurrence relation" (Fig 2.1).

value δ_0 is 0 for R-K methods (since they are of one-step), but could be non-zero for LMM. In practice, one needs to examine the error in the extra initial values brought by the bootstrapping procedure.

Remark. Recall that the iteration matrix

$$A = \begin{pmatrix} -a_{J-1} & -a_{J-2} & \dots & -a_1 & -a_0 \\ 1 & 0 & & & \\ & 1 & & & \\ & & & & \\ & & & 1 & 0 \end{pmatrix}$$

is introduced under the scalar setting $u(t) \in \mathbb{R}$ (Sec. 4.3.1) to transform a LMM scheme into Eqn. 7.1.1. This can be extended to the vector scenario; the iteration matrix reads

$$A = \begin{pmatrix} -a_{J-1}I_d & -a_{J-2}I_d & \dots & -a_1I_d & -a_0I_d \\ I_d & 0 & & & \\ & I_d & \ddots & & \\ & & \ddots & \ddots & \\ & & & I_d & 0 \end{pmatrix}$$

where we simply multiply each entry with an identity matrix.

7.1.3 R-K scheme as a one-step method

We provide a few more details in the course of formulating a R-K scheme as the one-step form. For a general implicit R-K method (recall Eqn. 3.2.3 and 3.2.4), let us put $K_n = (k_1, \dots, k_J)^T$ which follows

$$K_n = F\left(u_n + h\underline{a}K_n\right) \quad (7.1.2)$$

(the term \underline{a} refers to the coefficients in the Butcher table). Eqn. 7.1.2 has a unique solution for sufficiently small h ; this can be proved by using the contraction mapping theorem. In fact, $\Phi : K \mapsto F\left(u + h\underline{a}K\right)$ is a Lipschitz function due to

$$\begin{aligned} \|\Phi(K') - \Phi(K'')\| &\leq \left\| F\left(u + h\underline{a}K'\right) - F\left(u + h\underline{a}K''\right) \right\| \\ &\leq Lh \left\| \underline{a} \right\| \|K' - K''\| \end{aligned}$$

and its Lipschitz constant will be smaller than 1 if $Lh \left\| \underline{a} \right\| \leq 1$, thus applicable for applying the contraction mapping theorem. Let K^u denote the fixed point to Φ . Since $u_{n+1} = u_n + h \sum_j b_j k_j = u_n + h\underline{b}K^{u_n}$, it suffices to show that K^u is Lipschitz w.r.t. to u . A direct comparison yields

$$\begin{aligned} \|K^u - K^v\| &= \left\| F\left(u + h\underline{a}K^u\right) - F\left(v + h\underline{a}K^v\right) \right\| \\ &\leq \left\| F\left(u + h\underline{a}K^u\right) - F\left(u + h\underline{a}K^v\right) \right\| + \left\| F\left(u + h\underline{a}K^v\right) - F\left(v + h\underline{a}K^v\right) \right\| \\ &\leq Lh \left\| \underline{a} \right\| \|K^u - K^v\| + L \|u - v\|, \end{aligned}$$

leading to

$$\|K^u - K^v\| \leq \frac{L \|u - v\|}{1 - Lh \left\| \underline{a} \right\|}.$$

7.2 Differential algebraic equations

7.2.1 Example and applications

Let us consider an ODE coupled with an algebraic equation, such as

$$u' = f(u, v), \quad (7.2.1)$$

$$0 = g(u, v). \quad (7.2.2)$$

Typical applications include:

- Explicit constraints in mechanical systems, e.g. a pendulum system where the length of the

string stays constant.

- Control theory: usually u follows an ODE (for example Newton's law) and v is a control variable that serves as an adjustable coefficient.
- As a relaxation of stiff systems: for example, the v component in the solution to a stiff system

$$\begin{aligned}u' &= f(u, v), \\v' &= \frac{1}{\epsilon}g(u, v)\end{aligned}$$

typically converges to the root of $g(u, v) = 0$ on a faster scale when $\epsilon \ll 1$. Thus, it is convenient to formally put $v' = 0$ that leads to the DAE as mentioned above.

- PDE: usually emerge after space discretization, e.g. incompressible Navier-Stokes equation: the velocity follows the divergence-free constraint and the acceleration depends on the pressure variable which is a “missing piece”.

7.2.2 Approximation by implicit Euler method

A first attempt to solve Eqn. 7.2.1 and 7.2.2 is to use the explicit Euler method, i.e.

$$\begin{aligned}u_{n+1} &= u_n + hf(u_n, v_n), \\0 &= g(u_n, v_n).\end{aligned}$$

There is an issue with this scheme where no evaluation of v_{n+1} is yielded. Thus, we turn to the implicit version. We can apply the implicit Euler method to Eqn. 7.2.1 and plug the updated steps into the constraint (Eqn. 7.2.2), i.e.

$$\begin{aligned}u_{n+1} &= u_n + hf(u_{n+1}, v_{n+1}), \\0 &= g(u_{n+1}, v_{n+1}).\end{aligned}$$

7.2.3 Index of DAE

Another strategy of solving DAEs is to reduce them to ODEs. If Eqn. 7.2.2 admits a unique solution $v = \mathbf{g}^{-1}(u)$ which can be solved under a fair cost, then the DAE reduces to an ODE form $u' = f(u, \mathbf{g}^{-1}(u))$. We then define the index of a DAE as the number of reductions needed to transform into ODEs. Then, ODEs have index 0 and the aforementioned example has index 1.

Example 7.2.1. There are DAEs with higher indices. Consider

$$u' = f(u, v, w), \quad (7.2.3)$$

$$v' = g(u, v, w), \quad (7.2.4)$$

$$0 = h(u, v). \quad (7.2.5)$$

We cannot apply implicit Euler method to this system since w does not enter the algebraic equation explicitly. To circumvent this issue, let us differentiate Eqn. 7.2.5 to obtain

$$0 = \partial_u h(u, v) f(u, v, w) + \partial_v h(u, v) g(u, v, w) = \tilde{h}(u, v, w).$$

Since we need to differentiate (part of) the system to reduce to a index-1 DAE, this system is of index 2.

Feb 8: Lecture 8

Numerical Ordinary Differential Equations (Part VII: SDE and BVP)

Remark (Initial condition for DAEs). For problems in the form of Eqn. 7.2.1 and 7.2.2, $u(t_0)$ is usually given, then $v(t_0)$ may be solved if $g(u, \cdot) = 0$ admits a (possibly class of) solution. It is, however, more difficult to deal with if the DAE is given in the implicit form $F(u, u') = 0$ where $\frac{\partial F}{\partial(u')} = 0$ might even be singular. Methods for stiff ODES are suitable for addressing these problems.

8.1 Stochastic differential equations

Formally speaking, stochastic equations describe processes that involve random components, for example in a way that

$$u' = f(u, \omega) \tag{8.1.1}$$

where ω characterize the “randomness” that drives the system. Typical application of SDEs include weather prediction, finance, etc. A stochastic model can also emerge from physical and biological systems which shall be deterministic intrinsically due to Newton’s law, but it’s more handy and practical to model the molecule interactions as a stochastic process. In the same spirit, SDEs are applied in uncertainty quantification (UQ) where the posterior estimate often involves simulation of the random system.

From a numerical perspective, Eqn. 8.1.1 is not rigorous or regular since a wide class of stochastic processes are not continuously differentiable (or not differentiable to begin with). The following differential form

$$dX_t = f(X_t, t) dt + g(X_t, t) dW_t \tag{8.1.2}$$

is often adopted in SDE literature where X_t denotes the stochastic process and W_t is a given random process (usually the Wiener process or the Brownian motion). We usually refer $f dt$ as the drift term and $g dW_t$ as the diffusion term due to their physical meanings. Eqn. 8.1.2 shall be understood in the sense of the corresponding integral form, i.e. X_t solves

$$X_{t+s} = X_t + \int_t^{t+s} f(X_\xi, \xi) d\xi + \int_t^{t+s} g(X_\xi, \xi) dW_\xi$$

where the second integral is defined by Ito calculus which works differently from the usual calculus. A standard numerical method is the Euler-Maruyama scheme, namely

$$\widehat{X}_{n+1} = \widehat{X}_n + f(\widehat{X}_n, t_n) \Delta t + g(\widehat{X}_n, t_n) \Delta W_n$$

which is a natural extension to the explicit Euler scheme. Notice that the numerical solution is a random variable (or process if we consider the full trajectory), thus the definition of convergence requires close scrutiny. The weak error estimate, i.e. comparing the expectation at a given time t_n , is given by

$$\left| \mathbb{E}X_{t_n} - \mathbb{E}\widehat{X}_n \right| = \mathcal{O}(h)$$

if we assume an evenly spaced time grid with step h . The error estimate in the strong sense (i.e. path-by-path) can also be derived as

$$\mathbb{E} \left| \sup_{0 \leq t_n \leq T} |X_{t_n} - \widehat{X}_n|^2 \right|^{1/2} = \mathcal{O}(h^{1/2})$$

which follows a typical Monte-Carlo form. Numerical schemes of higher orders are possible (for example the SRK family), but they are much harder to design.

8.2 2-point boundary value problems

8.2.1 General theory

We wish to define a differential equation that involves the second order derivative of u , say modeling elasticity or conductivity. Since it is a system of second order, two boundary conditions are required: one can specify the values of u and u' at the starting step, but it is also possible to specify the

values at two boundary points, for example

$$u'' = f(u', u, x) \quad a < x < b, \quad (8.2.1)$$

$$u(a) = u_a, \quad (8.2.2)$$

$$u(b) = u_b. \quad (8.2.3)$$

Remark. The system above can be reduced to a first-order system. We introduce $u^{(1)} := u, u^{(2)} := u', u^{(3)} := x$ which leads to

$$\begin{aligned} [u^{(1)}]' &= u^{(2)}, \\ [u^{(2)}]' &= f(u^{(2)}, u^{(1)}, u^{(3)}), \\ [u^{(3)}]' &= 1. \end{aligned}$$

with the boundary condition $u^{(1)}(a) = u_a, u^{(1)}(b) = u_b, u^{(3)}(a) = a$. Thus, it naturally leads to the general form of BVPs

$$U' = F(U), \quad (8.2.4)$$

$$0 = G(U(a), U(b)). \quad (8.2.5)$$

The second order system (Eqn. 8.2.1, 8.2.2, and 8.2.3) can be solved via a direct discretization via FDM. For an evenly spaced grid $x_j := a + hj$ with $hJ = b - a$, let u_j denote the numerical approximation to $u(x_j)$. Then, the numerical scheme reads

$$\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = f\left(\frac{u_{j+1} - u_{j-1}}{2}, u_j, x_j\right), j = 1, \dots, J-1$$

$$u_0 = u_a,$$

$$u_J = u_b.$$

i.e. $J-1$ equations for $J-1$ unknowns u_1, \dots, u_{J-1} . For BVPs in the general form (Eqn. 8.2.4 and 8.2.5), the trapezoidal rule is preferred

$$\frac{1}{h}(U_{j+1} - U_j) = \frac{1}{2}[F(U_{j+1}) + F(U_j)], j = 0, \dots, J-1,$$

$$G(U_0, U_J) = 0.$$

Remark. FDM of higher orders are, however, not common and not widely adopted since it is not easy to derive the discretized boundary conditions. Higher accuracy is often obtained from Richardson

extrapolation instead.

8.2.2 Well-posedness

The well-posedness result for BVPs is harder than for IVPs since the solutions might not be unique.

Example 8.2.1. Let us consider the following system

$$\begin{aligned}u'' + \pi^2 u &= 0, 0 < x < 1, \\0 &= u(0) = u(1).\end{aligned}$$

The system above admits a class of solutions $u(x) = C \sin \pi x$ where the amplitude C is an arbitrary constant.

Let us analyze the numerical aspect of FDM discretization.

Example 8.2.2. One can verify that there is the unique solution to

$$\begin{aligned}-u'' + u &= f(x), 0 < x < 1, \\0 &= u(0) = u(1).\end{aligned}$$

The linear discretized system derived from FDM reads

$$\begin{aligned}-(u_{j+1} - 2u_j + u_{j-1}) + h^2 u_j &= h^2 f(x_j), \\0 &= u_0 = u_J\end{aligned}$$

that is equivalent to the matrix form

$$\begin{pmatrix} 2+h^2 & 1 & & & & \\ 1 & 2+h^2 & 1 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 2+h^2 & 1 \\ & & & & 1 & 2+h^2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_{J-2} \\ u_{J-1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_{J-2}) \\ f(x_{J-1}) \end{pmatrix} \quad (8.2.6)$$

The iteration matrix is strongly diagonally dominant, thus Eqn. 8.2.6 also admits a unique solution.

Remark. A similar argument can be applied to Eg. 8.2.1 as well, where the only difference is that

the iteration matrix reads

$$\begin{pmatrix} -2 + \pi^2 h^2 & 1 & & & & & \\ 1 & -2 + \pi^2 h^2 & 1 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 + \pi^2 h^2 & 1 & \\ & & & & 1 & -2 + \pi^2 h^2 & \end{pmatrix}.$$

The dominance no longer holds, thus it is not surprising that we encounter non-uniqueness. The numerical solution is then sensitive to the step size h . A similar phenomenon arises in the discretization to wave equations.

8.2.3 Richardson extrapolation

As mentioned earlier, FDM schemes of higher orders are not easy to derive. As an alternative, Richardson extrapolation is often more favorable. It can be motivated by the following derivation. Let us compare the numerical solution that uses time step h to the true solution

$$U_n^{(h)} = U(t_n) + h^2 e_2(t_n) + h^4 e_4(t_n) + \dots \quad (8.2.7)$$

which can be obtained via asymptotic expansion and order matching. If we cut the step size by half, we have

$$U_{2n}^{(h/2)} = U(t_n) + \frac{1}{4} h^2 e_2(t_n) + \frac{1}{16} h^4 e_4(t_n) + \dots$$

Then, a higher order estimate can be drawn by composing $U_n^{(h)}$ and $U_{2n}^{(h/2)}$ carefully:

$$\frac{4U_{2n}^{(h/2)} - U_n^{(h)}}{3} = U(t_n) - \frac{1}{4} h^4 e_4(t_n) + \dots \quad (8.2.8)$$

Eqn. 8.2.8 yields a higher order approximation and is also suitable for monitoring the local error. This often serves as a prototype for adaptive methods.

Remark. To obtain an expansion in the form of Eqn. 8.2.7, we start with the expansion with full terms

$$U_n^{(h)} = U(t_n) + h e_1(t_n) + h^2 e_2(t_n) + \dots$$

which is plugged into the FDM scheme. An order matching procedure then yields a set of ODEs that connects U, e_1, e_2 and their derivatives.

Feb 13: Lecture 9

Numerical Partial Differential Equations (Part I: 2-Point BVP, FDM, and FEM)

9.1 Finite difference method and 2-point boundary value problem

Remark. It is possible to derive direct discretization of higher order on the first order system $U' = F(U)$ with boundary condition $G(U(a), U(b)) = 0$ (Eqn. 8.2.4 and 8.2.5). But, the Richardson extrapolation is often preferred for stability issues; it also does not require extra numerical boundary conditions as is needed in higher order scheme. The adaptivity also helps to performance error estimate locally.

The discretization of boundary value problems often leads to a linear or possibly non-linear system of equations. In the non-linear case, the initial guess for iterative solver can be hard to find. A feasible approach is to introduce some homotopy over the solution space. Consider the following sequence of problems:

$$\begin{aligned} U'_\theta(x) &= \theta F(U_\theta(x)) + (1 - \theta) H(U_\theta(x)) \\ G(U_\theta(a), U_\theta(b)) &= 0 \end{aligned} \tag{9.1.1}$$

notice that Eqn. 9.1.1 falls back to the original problem if we let $\theta = 1$. When $\theta = 0$, the composed problem reads $U' = H(U)$, so we can pick some function H such that this starting problem is easy

to solve. Once the initial guess is obtained, we can then increase θ little by little and hope that the solution converges when θ reaches 1.

Another strategy is to reformulate the problem as an initial value problem coupled with an algebraic equation, a.k.a. the shooting method. The idea is to assume the boundary value $U(a) = s$, solve the IVP, and examine the terminal value $\tilde{U}(b)$. This terminal value, in general, depends on s , so the shooting method poses an additional equation $G(s, \tilde{U}(b; s)) = 0$.

Example 9.1.1. Let us consider the following second-order system

$$\begin{aligned} u'' &= f(u', u, x) & a < x < b, \\ u(a) &= u_a, \\ u(b) &= u_b. \end{aligned}$$

Once the initial derivative $s_1 = u'(a)$ is fixed, we can solve the IVP to obtain a trajectory $\tilde{u}(x; s_1)$. Then, it remains to solve $\tilde{u}(b; s_1) = u_b$.

The choice of s can be tricky since it solves a non-linear equation which is implicitly posed by the non-linear ODE solver. One simple approach is to use a bisection solver that only relies on the function value. Fixed point iteration is also viable. The preferred approach is Newton's method that utilizes the Jacobian of the implicit equation which can be obtained by solving a derived system of ODEs.

In general, BVPs do not have an easy guarantee of the well-posedness of the solutions. It may seem that the shooting method circumvents the need to identify the unique solution. We shall point out that the shooting method introduces an additional terminal condition equation. Nevertheless, sometimes it is a viable strategy to show the existence and uniqueness of the solution to BVP.

9.2 Finite element method

9.2.1 Strong and weak form

Now, we shift our focus to finite element method. Loosely speaking, the solution function is approximated by point-wise values in the finite difference method; in finite element method, the solution is approximated by a piece-wise linear function. To motivate this, consider the following second-order system

$$\begin{aligned} -u'' + a(x)u &= f(x) & x_L < x < x_R, & \quad (\text{S}) \\ u(x_L) &= u(x_R) = 0. \end{aligned} \tag{9.2.1}$$

Let us approximate the true solution by a combination of trial functions, i.e.

$$u(x) \approx u_h(x) = \sum_{j=1}^J \alpha_j \varphi_j(x)$$

where φ_j are called trial functions, given and fixed for a specific FEM. We rewrite the DE into the weak form to derive equations for $\{\alpha_j\}$ ¹. We multiply the strong form (Eqn. 9.2.1) by test functions $v(x)$ and integrate over the domain (x_L, x_R) :

$$\int_{x_L}^{x_R} (-u''v + auv) dx = \int_{x_L}^{x_R} fv dx.$$

We can apply integration by parts (IBP) on LHS

$$\int_{x_L}^{x_R} (-u''v) dx = (-u'v)|_{x_L}^{x_R} + \int_{x_L}^{x_R} u'v' dx.$$

To avoid the boundary term, let us pick $v(x)$ that vanishes at the two boundary points x_L and x_R , leading to

$$\int_{x_L}^{x_R} (u'v' + auv) dx = \int_{x_L}^{x_R} fv dx. \quad (9.2.2)$$

We introduce the following definition

$$A(u, v) := \int_{x_L}^{x_R} (u'v' + auv) dx,$$

$$F(v) := \int_{x_L}^{x_R} fv dx.$$

Then, Eqn. 9.2.2 is put equivalently as

$$A(u, v) = F(v) \quad (\text{V}) \quad (9.2.3)$$

Notice that A is a bilinear form, i.e. being linear in the two arguments. F is a linear functional, meaning that it maps functions to scalar values.

Let us inspect the meaning of the strong and weak formulation closely.

¹the name “weak” comes from the fact that solutions are less regular (say, less continuous, less integrable...) than those we have seen in ODEs

Strong form (Eqn. 9.2.1)	Weak form (Eqn. 9.2.3)
find $u \in C^2(x_L, x_R)$ s.t. u vanishes at x_L and x_R while	find u in $H_0^1([x_L, x_R])$ (to be clarified later) s.t.
$-u'' + a(x)u = f(x)$	$A(u, v) = F(v)$
holds for $x_L < x < x_R$	for any $v \in H_0^1([x_L, x_R])$

It seems we only require $u, v \in C^1$ since only the first derivative is involved. In fact, the requirement is even weaker that u can be a H_0^1 function. The definition for Sobolev spaces in detail is beyond the scope of this class, but formally speaking, H_0^1 contains functions that can be differentiated while vanishes on the boundary, i.e.

$$H_0^1([x_L, x_R]) := \{u : u' \in L^2([x_L, x_R]), u(x_L) = u(x_R) = 0\}.$$

We shall emphasize that the solution to the weak form does not necessarily solve the strong form (it is a “weaker” solution after all).

9.2.2 Discretization

The general idea of the FEM (or Galerkin method) is to discretization the transformed variational problem (Eqn. 9.2.3). For the discretization setup, we solve the same variational problem, but restricting the trial and test functions to be of the linear combination form, i.e.

$$A(u, v) = F(v), \forall v \in V_h \quad (V_h) \quad (9.2.4)$$

Due to linearity, we do not need to verify Eqn. 9.2.4 for every possible v in V_h ; it suffices to examine the problem by just using $\varphi_1, \dots, \varphi_J$, leading to

$$\sum_{j=1}^J \alpha_j A(\varphi_j, \varphi_k) = f(\varphi_k), \forall 1 \leq k \leq J.$$

There are J linear equations for J unknowns where the coefficients

$$A(\varphi_j, \varphi_k) = \int_{x_L}^{x_R} (\varphi_j' \varphi_k' + a \varphi_j \varphi_k) dx =: A_{j,k}^s + A_{j,k}^m$$

are built by suitable numerical integration methods, e.g. the Gaussian integral quadrature. The first part $A_{j,k}^s$ of the integral is often referred to as stiffness matrix, the second part $A_{j,k}^m$ being mass matrix, where the naming came after practical physical studies dated back to early days of FEM

development in engineering.

A popular choice for the basis function is the P_1 element, where each basis function looks like a hat function.

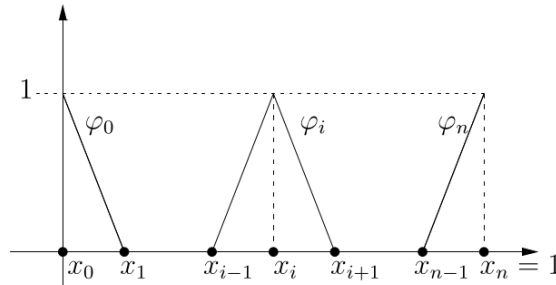


Fig. 12.3. Shape functions of X_h^1 associated with internal and boundary nodes

Figure 9.2.1: Hat functions and P_1 element, from From “Numerical Mathematics”, page 561.

For P_1 elements, the space of trial functions is exactly the piecewise linear functions that FDM adopts. However, FEM approach tells what happens in between the consecutive sampling points. For higher order elements, it is not necessarily the case that interpolation is linear.

We examine the coefficients for the stiffness matrix if using P_1 element. A direct calculation yields

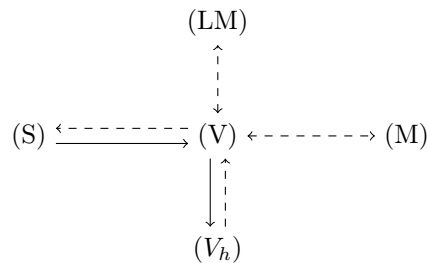
$$A_{j,k}^s = \begin{cases} 2/h & j = k \\ -1/h & |j - k| = 1 \\ 0 & \text{otherwise} \end{cases}.$$

The shape of A^s is a tridiagonal matrix, which resembles the second order derivative discretized by the finite difference method.

Feb 15: Lecture 10

Numerical Partial Differential Equations (Part II: FEM)

Let us review the diagram that connects different forms arising in the method of finite elements.



10.1 Connecting (V) and (LM)

To derive well-posedness of the BVP, the following properties of the variational components A and F are often desired.

1. Coercivity of A : there exists C_1 positive s.t. $A(u, u) \geq C_1 \|u\|_{H^1}^2$.¹
2. Continuity of A : there exists C_2 finite and positive s.t. $|A(u, v)| \leq C_2 \|u\|_{H^1} \|v\|_{H^1}$.
3. Continuity of F : there exists C_3 positive s.t. $|F(v)| \leq C_3 \|v\|_{H^1}$.

¹The H^1 -norm is defined as

$$\|u\|_{H^1} = \sqrt{\int_{x_L}^{x_R} (u')^2 + u^2 dx} = \sqrt{\|u\|_{L^2}^2 + \|u'\|_{L^2}^2}.$$

4. Symmetry of A : $A(u, v) = A(v, u)$.
5. $A(u, v)$ is a bilinear form and F is a linear functional.

The last two properties are easy to verify. The first one on coercivity is satisfied if $a(x)$ has a uniform positive lower bound. In fact,

$$\begin{aligned} A(u, u) &= \int_{x_L}^{x_R} [(u')^2 + a(x)u^2] dx \\ &\geq \min\left(\min_{x_L \leq x \leq x_R} a(x), 1\right) \int_{x_L}^{x_R} [(u')^2 + u^2] dx \\ &= C_1 \|u\|_{H^1}^2. \end{aligned}$$

The second and third property is usually a consequence of Cauchy's inequality:

$$\begin{aligned} |F(v)| &= \left| \int_{x_L}^{x_R} f v dx \right| \leq \sqrt{\int_{x_L}^{x_R} f^2 dx} \sqrt{\int_{x_L}^{x_R} v^2 dx} = \|f\|_{L^2} \|v\|_{L^2}, \\ |A(u, v)| &\leq C \sqrt{A(u, u)} \sqrt{A(v, v)}. \end{aligned}$$

Now we can connect to the Lax-Milgram theorem, which states that there exists a unique solution to the variational form in H^1 if the properties 1, 2, 3, 5 are satisfied; notice that symmetry is not a mandatory requirement.

10.2 From (V) to (S)

To go from the weak form back to the strong form, recall that the weak solution is merely in H^1 , so one would expect that stronger conditions are needed. In this particular example, if the source term f is a L^2 function, then u can be shown to be a H^2 function. Then, by performing IBP in the reverse direction and handling the boundary term properly, we arrive at that

$$\int_{x_L}^{x_R} (-u'' + au - f) v dx = 0 \tag{10.2.1}$$

holds for any $v \in H^1$. Since $g := -u'' + au - f$ is in L^2 , we can pick v to be g and the integral reads $\|g\|_{L^2}^2$, leading to the conclusion that $g = 0$ in L^2 . A stronger argument on a point-wise basis can be derived if $f \in C^1$; g is also a continuous function in this case. Let us assume, for the sake of negating the conclusion to be shown, that $g(\bar{x})$ is not 0 for some $\bar{x} \in (x_L, x_R)$. Then, by continuity of g , it must be bounded away from 0 in some neighborhood \mathcal{N} of \bar{x} . One can thus pick a proper positive mollifier v that is supported in \mathcal{N} to show a contradiction by examining the integral.

10.3 Connecting (V) and (M)

The five properties (including symmetry) implies that the variational form is equivalent to the minimization form

$$u = \arg \min_{u \in H_0^1} \left[\frac{1}{2} A(u, u) - F(u) \right] \quad (\text{M}).$$

To see this, let us pick a perturbation direction $v \in H_0^1$ and the associated amplitude $\epsilon > 0$. The perturbed value reads

$$\frac{1}{2} A(u + \epsilon v, u + \epsilon v) - F(u + \epsilon v) = \left[\frac{1}{2} A(u, u) - F(u) \right] + \epsilon [A(u, v) - F(v)] + \frac{\epsilon^2}{2} A(v, v).$$

Now, if (V) is valid, then $A(u, v) = F(v)$ holds true, so the ϵ term vanishes and the perturbation leads to an additional ϵ^2 term, showing that u is the minimizer. On the other hand, if (M) holds true, that u is indeed a minimizer but we do not know if u necessarily solves the variational form. If not, for example there exists v s.t. $A(u, v) < F(v)$, then we can pick ϵ sufficiently small so that $\epsilon A(v, v) \leq -[A(u, v) - F(v)]$, then

$$\epsilon [A(u, v) - F(v)] + \frac{\epsilon^2}{2} A(v, v) \leq \frac{\epsilon}{2} [A(u, v) - F(v)] < 0$$

which contradicts with the fact that u is a minimizer, thus eliminating the possibility that $A(u, v)$ does not match $F(v)$.

10.4 Connecting (V) and (V_h)

The two variational problems are not that different since they share an almost identical form, except that the function spaces are different. We require that the numerical (usually finite dimensional) function space V_h is a subspace of V . As a consequence, the true solution u should also fit in the discretized problem, i.e.

$$A(u, v_h) = F(v_h), \forall v_h \in V_h.$$

Recall that the numerical version reads $A(u_h, v_h) = F(v_h)$, leading to

$$A(u - u_h, v_h) = 0$$

which implies that the residual in the approximation $u - u_h$ is orthogonal to any discrete test function v_h . We will work on the approximation error estimate in the next lecture.

Feb 20: Lecture 11

Numerical Partial Differential Equations (Part III: FEM and Poincare's inequality)

11.1 FEM error estimate

Recall that we have established some basics for solving variational problems in previous lectures. The variational problem, a.k.a. weak form (usually of infinite dimension), possesses the form

$$\text{find } u \in V \text{ s.t. } a(u, v) = f(v) \quad \forall v \in V, \quad (\text{V})$$

while the discretized problem in the language of FEM/Galerkin method, as a finite dimensional problem, reads

$$\text{find } u \in V_h \text{ s.t. } a(u, v) = f(v) \quad \forall v \in V_h. \quad (\text{V}_h)$$

The usual setup for 2-point BVP is that $V = H_0^1([x_L, x_R])$ and we usually require that V_h is a subspace of V . A direct consequence is to residual error estimate. Since $u_h \in V_h$ and $u \in V$ solves the discretized problem (V_h) , for any $v \in V_h$,

$$a(u, v) = f(v) = a(u_h, v). \quad (11.1.1)$$

Then, one can show an estimate as follows.

Theorem 11.1.1. *We assume that the assumptions 1-3 and 5 in Sec. 10.1 are met for the varia-*

tional problems. Then, there exists a constant $C > 0$ s.t.

$$\|u - u_h\|_V \leq C \inf_{w_h} \|u - w_h\|_V. \quad (11.1.2)$$

Remark. Eqn. 11.1.2 can be interpreted as “ u_h is almost the best approximation of u in V_h , up to a constant C ”. Nevertheless, one still needs to develop approximation theory that guarantees that the approximation error vanishes as h goes to 0.

(*Proof to Thm. 11.1.1*). Since $u_h, w_h \in V_h$, we plug $v = u_h - w_h$ into Eqn. 11.1.1 and use the fact that a is a bilinear form:

$$0 = a(u - u_h, u_h - w_h) = a(u - u_h, u - w_h) - a(u - u_h, u - u_h).$$

Since a is coercive and bounded, we have

$$C_1 \|u - u_h\|_V^2 \leq a(u - u_h, u - u_h) = a(u - u_h, u - w_h) \leq C_2 \|u - u_h\|_V \|u - w_h\|_V,$$

leading to

$$\|u - u_h\|_V \leq \frac{C_2}{C_1} \|u - w_h\|_V.$$

□

One consequence of Thm. 11.1.1 is to bound the approximation error for P_1 elements. Since Eqn. 11.1.2 holds for any $w_h \in V_h$, we can build linear interpolation approximations, should these fall in V_h . To be specific, let φ_j denote hat functions s.t. $\tilde{u} = \sum \alpha_j \varphi_j$ interpolates u at the nodes $\{x_j\}$.

One can derive the error bound from the following intuitive calculation. For each interval $[x_i, x_{i+1}]$, let us pick the point ξ_i s.t.

$$u'(\xi_i) = \tilde{u}'(\xi_i) = \frac{u(x_{i+1}) - u(x_i)}{x_{i+1} - x_i}$$

thanks to Lagrange’s intermediate value theorem. Then, the point-wise error in u' is bounded by

$$|u'(x) - \tilde{u}'(x)| = \left| u'(\xi_i) - \tilde{u}'(\xi_i) + \int_{\xi_i}^x u''(y) dy \right| \leq h \max |u''|$$

which is useful to show error in u as

$$|u(x) - \tilde{u}(x)| \leq h |u'(x_i) - \tilde{u}'(x_i)| + \left| \frac{h^2}{2} u''(\zeta_i) \right| \leq Ch^2 \max |u''|.$$

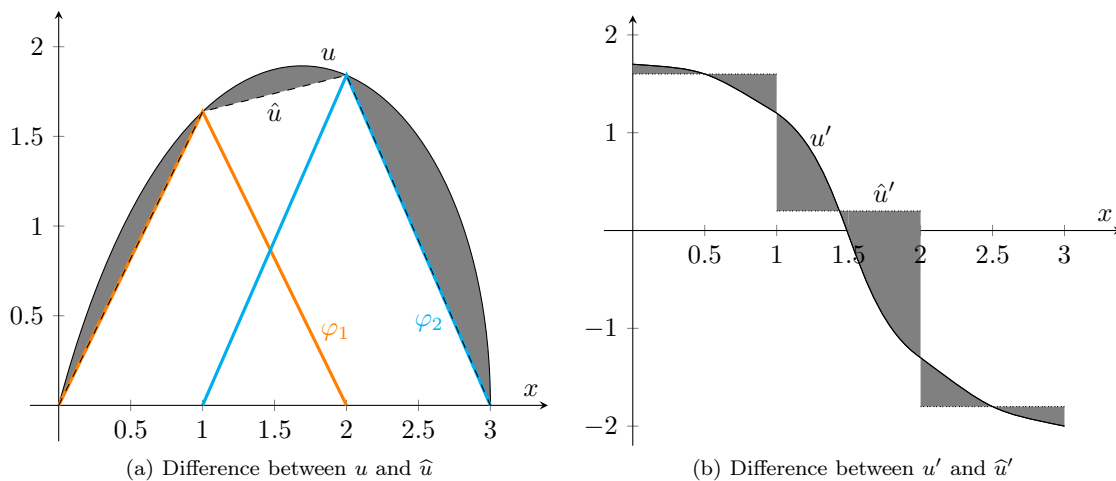


Figure 11.1.1

Thus, the H^1 norm of the difference $u - u_h$ can be obtained via a simple integral estimate, i.e.

$$\|u - u_h\|_{H^1} \leq Ch \max |u''|. \quad (11.1.3)$$

Remark. The usual result for FE approximation, especially in higher dimensions, is to replace the L^∞ norm on the RHS by a L^2 norm, i.e.

$$\|u - u_h\|_{H^1} \leq Ch \|u''\|_{L^2} = Ch |u|_{H^2}.$$

Notice the notation difference between $|\cdot|_{H^2}$ and $\|\cdot\|_{H^2}$; the former is defined as $\|\cdot\|_{L^2}$ and is thus a semi-norm.

Remark. It may seem that we can derive a L^2 estimate from the derivation that leads to Eqn. 11.1.3 since we have a point-wise estimate. This is, however, not the case for higher dimensions, at least not as easily as in the 1 dimensional case. We refer the readers to the Aubin-Nitsche-duality method.

Let us revisit the variational form. Recall that we have derived an orthogonal condition (Eqn. 10.2.1) earlier:

$$\int_{x_L}^{x_R} (-u'' + au - f) v \, dx = 0.$$

This also holds true for $u_h, v \in V_h$ for the discretized problem. In general, we can not conclude that $-u_h'' + au_h - f = 0$ since u_h does not even have enough regularity to yield a second order derivative. Nevertheless, it is reasonable to claim that the residual $-u_h'' + au_h - f$ is “orthogonal” to

any $v \in V_h$ (in the sense of distributions). This can be viewed as a relaxed condition as compared to the requirement that the original strong form is met at the node locations $\{x_j\}$. The latter approach, often known as the collocation method, is sometimes applied to integral equations and has come back recently in some machine learning techniques such as PINNs (Physical-Informed Neural Network).

11.2 Remarks on coercivity

Recall that we require the coefficient $a(x)$ to be lower bounded by $a_0 > 0$ to conclude coercivity for the bilinear form $a(u, v)$. We will show that it can be relaxed to lower bound condition for $a_0 \geq 0$, but apparently not any further since counter-examples may arise, such as $-u'' - u = 0$ for suitable domains. One critical case of interest is when $a(x) = 0$; the differential equation now reads

$$-u'' = f(x), u(x_L) = u(x_R) = 0. \quad (11.2.1)$$

It may seem trivial to solve Eqn. 11.2.1 by simply integrating it twice. However, this approach breaks down for general PDE problems, where Eqn. 11.2.1 is also known as the Poisson problem.

The alternative approach is to establish the coercivity condition by studying the connection between u and its derivative u' . To begin with, let us express the value of u by integrating the derivative

$$u(x) = \int_{x_L}^x u'(\xi) \, d\xi = \int_{x_L}^x u'(\xi) \cdot 1 \, d\xi.$$

Then, we apply the Cauchy's inequality

$$|u(x)| = \sqrt{\int_{x_L}^x u'(\xi)^2 \, d\xi} \cdot \sqrt{\int_{x_L}^x 1^2 \, d\xi} \leq \|u'\|_{L^2} \sqrt{x_R - x_L},$$

leading to

$$\|u\|_{L^2} \leq \|u'\|_{L^2} |x_R - x_L|. \quad (11.2.2)$$

Notice that this derivation only relies on the condition $x_L = 0$ without knowing anything about x_R .

Eqn. 11.2.2 is also known as the Poincaré's inequality. The general form of this theorem is that if $u \equiv u_0$ on part of the boundary $\Gamma \subset \partial\Omega$, $|\Gamma| > 0$ while $u, u_0 \in H^1$, then

$$\|u - u_0\|_{L^2} \leq C \|\nabla u\|_{L^2}.$$

To connect this back to the BVP, let us study the variational form

$$a(u, v) = \int_{x_L}^{x_R} u'v' \, dx.$$

By Poincaré's inequality, since

$$\int_{x_L}^{x_R} u^2 \, dx \leq (x_R - x_L)^2 \int_{x_L}^{x_R} (u')^2 \, dx,$$

we can add $\int_{x_L}^{x_R} (u')^2 \, dx$ on both sides to obtain

$$\|u\|_{H^1}^2 \leq \left[(x_R - x_L)^2 + 1 \right] \|u'\|_{L^2}^2 \leq \left[(x_R - x_L)^2 + 1 \right] a(u, u).$$

Feb 22: Lecture 12

Numerical Partial Differential Equations (Part IV: Boundary Conditions for BVPs)

Remark. Regarding homework problem 2, we can formulate the shooting problem as $G(s, Y(1; s)) = 0$, should we start with assuming $Y(0) = s$. This equation can be efficiently solved via Newton's method, but this approach requires to know the gradient of $G(s, Y(1; s))$. A closer examination reveals that

$$\frac{d}{ds}G(s, Y(1; s)) = \partial_1 G + \partial_2 G \partial_s Y(1; s)$$

where the second term is interpreted in the following manner: $Y(x; s)$ solved the original differential equation with initial value $Y(0) = s$. If we assume that $Y(x; s)$ depends on s in a continuously differentiable manner, then $\dot{Y} = f(Y)$ leads to

$$\partial_s \dot{Y}(x; s) = \partial_s \dot{Y}(x; s) = \partial_s f(Y(x; s)) = f'(Y) \partial_s Y$$

which is a matrix differential equation. The initial condition is simply $\partial_s Y = I$ since we assume $Y(0; s) = s$.

Remark. When discretizing the infinite-dimensional problem (V) to (V_h) , the symmetry assumption is not mandatory but is often preferred since it helps to solve the discretized system. For a symmetric and positive definite system, one can deploy conjugate gradient method or Cholesky decomposition for less time needed and more accurate results.

12.1 Neumann BC

Recall that in the last lecture, we have established the Poincaré's inequality, namely $\|u\|_{L^2} \leq C \|u'\|_{L^2}$ for $u(x_L) = 0$. We claim that there is a similar version for solutions to the Neumann's boundary condition, i.e.

$$u'(x_L) = u'(x_R) = 0.$$

Let us derive the weak form first. For suitable test functions v (properties of which will be determined soon), we have

$$\int_{x_L}^{x_R} (u'v' + a(x)uv) dx - [u'v]_{x_L}^{x_R} = \int_{x_L}^{x_R} f v dx.$$

Thus, the variational form for Neumann boundary condition reads

$$\text{find } u \in H^1 \text{ s.t. } a(u, v) = F(v), \forall v \in H^1. \quad (12.1.1)$$

Notice that we have dropped the subscript 0 in the Sobolev spaces, meaning that we do not require the solution or the test function to vanish on the boundary.

Similarly to the Dirichlet problem, the assumptions for Lax-Milgram theorem are satisfied when $a(x) \geq a_0 > 0$. The situation gets a bit tricky when $a(x)$ is no longer lower bounded away from 0. Consider the solution to $-u'' = f(x)$ that satisfies the Neumann BC; in contrast to the Dirichlet problem, there is no unique solution since adding a constant to one existing solution leads to another solution. This also implies that Poincaré's inequality does not hold in the same form. In fact, it motivates the notion of \overline{H}^1 spaces where functions have mean zero over the domain, thus avoiding the ambiguity. One can derive a similar version of Poincaré inequality on this space.

Another issue of the variational form (Eqn. 12.1.1) is that it says nothing about the boundary condition. However, we claim that the Neumann BC is encoded implicitly. In fact, if we assume enough regularity, then Eqn. 12.1.1 leads to

$$\int_{x_L}^{x_R} \left(\underbrace{-u'' + a(x)u - f(x)}_{=:g(x)} \right) v dx + [u'v]_{x_L}^{x_R} = 0$$

for any $v \in H^1$. If g does not vanish at some point $x_0 \in (x_L, x_R)$, one can choose a suitable mollifier v that supports on the neighborhood of x_0 where g is bounded away from 0, leading to contradiction. For the boundary term, we can simply choose a suitable mollifier with thin support containing each boundary point, leading to $u' = 0$. The Neumann BC is also known as the natural BC for this reason.

12.2 Inhomogeneous Dirichlet BC, etc.

Let us consider a variation of the zero Dirichlet conditions, namely

$$u(x_L) = u_L, u(x_R) = u_R.$$

The variational form still reads $a(u, v) = F(v)$ for any $v \in H_0^1$, but the problem is that u does not live in a linear subspace; in fact, the sum of two functions that meet the given BC does not satisfy the same any longer. One way to circumvent this issue is to introduce some function g that satisfies the desired condition $g(x_L) = u_L, g(x_R) = u_R$. Then, we let $\tilde{u} := u - g$ and the strong form reads

$$\tilde{u}'' - a(x)\tilde{u} = f(x) - g''(x) + a(x)g =: \tilde{f}(x).$$

Thus, \tilde{u} is suitable for applying the established theories. In practice, one can simply add two hat functions that handle the two boundary points which introduce two extra linear equations.

For more complex BC, such as

$$\begin{aligned} -u'' + a(x)u &= f(x), \\ u'(x_L) &= u'_L, \\ u'(x_R) &= Cu(x_R). \end{aligned}$$

The boundary term reads

$$[u'v]_{x_L}^{x_R} = Cu(x_R)v(x_R) - u'_L v(x_L)$$

where the first term is a linear form and the second term is merely a linear function of v . The variational form is thus

$$\begin{aligned} a(u, v) &= \left[\int_{x_L}^{x_R} u'v' + a(x)uv \, dx \right] - Cu(x_R)v(x_R), \\ F(v) &= \int_{x_L}^{x_R} fv \, dx - u'_L v(x_L). \end{aligned}$$

Feb 27: Lecture 13

Numerical Partial Differential Equations (Part V: Inhomogeneous BC and Higher Order Problems)

13.1 Inhomogeneous BC (continued)

Let us continue the discussion on inhomogeneous Dirichlet BC, say $u(x_L) = u_L \neq 0$. If we adopt P_1 elements, we can pose the numerical trial functions as

$$u_h = u_L \varphi_0 + \sum_{j=1}^J \alpha_j \varphi_j$$

so that the left boundary is naturally u_L .

For inhomogeneous Neumann BC, e.g. $u'(x_L) = 0$ and $u'(x_R) = 2$, we shall take this into account when deriving the weak form. Notice that

$$\begin{aligned} \int_{x_L}^{x_R} [-u'' + a(x)u]v \, dx &= \int_{x_L}^{x_R} [u'v' + a(x)uv] \, dx - [u'v]_{x_L}^{x_R} \\ &= \int_{x_L}^{x_R} [u'v' + a(x)uv] \, dx - 2v(x_R), \end{aligned}$$

thus the extra $2v(x_R)$ will be placed on the right hand side with the linear functional $\int_{x_L}^{x_R} f v \, dx$.

However, one needs to show that the combined functional is bounded, i.e.

$$|2v(x_R)| \leq C \|v\|_{H^1}. \quad (13.1.1)$$

Recall that the H^1 -norm is a combination of L^2 -norm and the L^2 -norm applied on the derivative. This is a particular example of the so-called “Trace Theorem” that is derived from theories involving Sobolev spaces; nevertheless, we shall point out that Eqn. 13.1.1 is not true if the H^1 -norm is replaced by either building component ($\|v\|_{L^2}$ or $\|\nabla v\|_{L^2}$). Here, we provide a quick proof using calculus.

Proposition 13.1.1. *For $u \in H^1((x_L, x_R))$, there exists a constant $C > 0$ s.t. $|u(x_R)| \leq C \|u\|_{H^1}$.*

Proof. Let us consider a point $\bar{x} \in (x_L, x_R)$. By Newton-Leibniz theorem,

$$|u(x_R)| \leq |u(\bar{x})| + \left| \int_{\bar{x}}^{x_R} u' dx \right|.$$

Recall that by Cauchy’s inequality,

$$\left| \int_{\bar{x}}^{x_R} u' \cdot 1 dx \right| \leq \sqrt{\int_{\bar{x}}^{x_R} |u'|^2 dx} \sqrt{\int_{\bar{x}}^{x_R} 1^2 dx} \leq \|u'\|_{L^2} \sqrt{x_R - x_L}.$$

Thus,

$$\begin{aligned} (x_R - x_L) |u(x_R)| &\leq \int_{x_L}^{x_R} |u(\bar{x})| d\bar{x} + \int_{x_L}^{x_R} \left| \int_{\bar{x}}^{x_R} u' dx \right| d\bar{x} \\ &\leq \sqrt{\int_{x_L}^{x_R} |u(\bar{x})|^2 d\bar{x}} \sqrt{x_R - x_L} + (x_R - x_L) \|u'\|_{L^2} \sqrt{x_R - x_L} \\ &\leq C (\|u\|_{L^2} + \|u'\|_{L^2}) \leq 2C \|u\|_{H^1}. \end{aligned}$$

□

13.2 Higher order problems

The aforementioned methodology also applies to problems involving higher order derivatives. Let us consider the following model problem

$$\begin{aligned} u^{(4)} - k^2 u'' &= f(x) & x_L < x < x_R, \\ u = u' &= 0 & x = x_L \text{ or } x_R. \end{aligned}$$

To derive the weak formulation, let us pick a proper test function v and apply integration by parts:

$$\begin{aligned} \int_{x_L}^{x_R} u^{(4)} v \, dx &= [u''']_{x_L}^{x_R} - \int_{x_L}^{x_R} u''' v' \, dx \\ &= [u''']_{x_L}^{x_R} - [u'' v']_{x_L}^{x_R} + \int_{x_L}^{x_R} u'' v'' \, dx, \\ \int_{x_L}^{x_R} -k^2 u'' v \, dx &= -[k^2 u' v]_{x_L}^{x_R} + \int_{x_L}^{x_R} k^2 u' v' \, dx. \end{aligned}$$

To make the boundary term vanish, we can require v to satisfy the same BC as the solution u , leading to the weak form

$$\text{find } u \in H_0^2 \text{ s.t. } a(u, v) = F(v) \quad \forall v \in H_0^2 \quad (\text{V}) \quad (13.2.1)$$

where

$$a(u, v) := \int_{x_L}^{x_R} (u'' v'' + k^2 u' v') \, dx, \quad F(v) := \int_{x_L}^{x_R} f v \, dx.$$

This system is suitable for modeling clamped elastic rod which is fixed at both ends as well as the boundary angles; the source term f can be interpreted as an external force.

Let us consider the basis function in the corresponding FEM problem to Eqn. 13.2.1. The regularity requirement is that $\varphi_j \in H_0^2$, so the hat functions no longer works. Also, we need the basis functions to express the value of derivatives at the nodes freely. Cubic functions are a suitable candidate towards this goal. We introduce φ_j and ψ_j s.t.

$$\begin{aligned} \varphi_j(x_i) &= \delta_{ji}, \varphi_j'(x_i) = 0; \\ \psi_j(x_i) &= 0, \psi_j'(x_i) = \delta_{ji}. \end{aligned}$$

Then, the solution can be written as the combination $u_h = \sum \alpha_j \varphi_j + \sum \beta_i \psi_i$.



Figure 13.2.1

Remark. Notice that cubic functions are required to construct basis for weak formulation that involves second order derivatives. This is usually quite expensive after being generalized to high dimensions, so one shall avoid using higher order derivatives when modeling problems with large dimensions.

Remark. However, higher order basis functions can still be useful, even in weak formulation that only involves $u'v'$ terms. This is due to the fact that they have better approximation properties, thus leading to more accurate result and faster convergence. For example, we can construct P_2 elements by using local quadratic functions.

13.3 Non-linear problems

For non-linear problems, we can still apply the Galerkin method, at least formally. For example, when the source term depends on u and u' in an arbitrary manner:

$$-u'' = f(u', u, x), u(x_L) = 0, u(x_R) = 0.$$

We pick the same basis function and trial space $u = \sum_{j=1}^J \alpha_j \varphi_j$ that leads to a system of J non-linear equations

$$\sum_{j=1}^J \alpha_j \int \varphi_j' \varphi_k' dx - \int f\left(\sum_{j=1}^J \alpha_j \varphi_j', \sum_{j=1}^J \alpha_j \varphi_j, x\right) \varphi_k = 0. \quad (13.3.1)$$

We shall point out that there is no universal established theory for general non-linear f and the well-posedness results are studied for each particular case.

Solving Eqn. 13.3.1 is not an easy job either. Recall that the Gaussian elimination can be deployed for linear problems and Cholesky for symmetric linear problems. For non-linear systems, the Newton method or linearized fixed point iteration is often used.

Mar 1: Lecture 14

Numerical Partial Differential Equations (Part VI: FEM for PDEs, Poisson Equation)

14.1 Introduction to PDEs

Example 14.1.1. We study the following model problem (often known as “Poisson equation”)

$$\begin{aligned} -\Delta u &= f(x) & x \in \Omega \subseteq \mathbb{R}^d, \\ u(x) &= 0 & x \in \partial\Omega. \end{aligned} \tag{14.1.1}$$

Remark 14.1.1. In general, we can classify common PDEs depending on the highest order of the derivative, being linear or non-linear, or the characteristic of the solution. For example, let us consider the following second order equation

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu = g(x, y).$$

We can put this problem into three different groups depending on the determinant $\Delta = b^2 - 4ac$:

- elliptic ($\Delta < 0$): standard form is $-\Delta u = g$.
- parabolic ($\Delta = 0$): standard form is $(\partial_t - \partial_x^2)u = g$, or the generalization could be $\partial_t = A\partial_{xx}u$ for $\sigma(A) > 0$.
- hyperbolic ($\Delta > 0$): standard form is $(\partial_t^2 - \partial_x^2)u = g$.

Example 14.1.2. The coupled system $u_t = v_x$ and $v_t = u_x$ can be written as

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_x$$

or equivalently the vector form $U_t = AU_x$ where $U = \begin{pmatrix} u \\ v \end{pmatrix}$ and $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Notice that the eigenvalues to A are 1 and -1 , thus A is diagonalizable. This is also the usual condition that leads to well-posedness.

Similarly to 2-point BVP, extra conditions are often needed for well-posedness. For elliptic problem, the BCs are posed on $\partial\Omega$, e.g. zero Dirichlet condition $u(x) = 0$ for all $x \in \partial\Omega$. For parabolic and hyperbolic problems, the BCs are usually on part of $\partial\Omega$ since Ω is often the cylinder product of space domain Ω_{space} and time domain $[0, T]$. Sometimes, we also call $\partial\Omega_{\text{space}} \times [0, T]$ as the “boundary condition” (without raising ambiguity), e.g. Dirichlet condition for absorbing or constant thermal source, or Neumann condition for reflecting phenomenons; we also pose “initial conditions” on $\Omega_{\text{space}} \times \{0\}$ to inject information at the time of beginning.

14.2 Derivation of the weak form

Let us derive the weak form to Eqn. 14.1.1. We pick a proper test function v which is multiplied to the differential equation followed by integration on Ω , leading to

$$\int_{\Omega} (-\Delta u) v \, dx = \int_{\Omega} f v \, dx. \tag{14.2.1}$$

We wish to apply the integral by parts formula as in the 1d case; it is possible to treat the LHS as a nested integral and apply IBP on each variable if Ω is a rectangular domain, but there is a way to handle more general scenarios. To this end, recall the Green’s formula

$$\int_{\Omega} (-\Delta u) v \, dx = \int_{\partial\Omega} (-\partial_n u) v \, ds + \int_{\Omega} (\nabla u) \cdot (\nabla v) \, dx \tag{14.2.2}$$

where ds stands for the surface infinitesimal element and n is the normal direction that points outward. If we pick v that vanishes on the boundary as u does, a combination of Eqn. 14.2.1 and 14.2.2 leads to

$$a(u, v) = F(v) \quad (\text{V})$$

where

$$a(u, v) := \int_{\Omega} (\nabla u) \cdot (\nabla v) \, dx, F(v) := \int_{\Omega} f v \, dx.$$

Let us examine the space of test functions v . Mimicking the definition in the 1d case, we should define the norm in terms of v and ∇v , i.e.

$$H_0^1(\Omega) := \left\{ v : \int_{\Omega} (v_x)^2 + (v_y)^2 + v^2 \, dx \, dy < \infty, v(x, y) = 0 \, \forall (x, y) \in \partial\Omega \right\}.$$

Under the similar assumptions (a being a bounded, symmetric and coercive bilinear form, F being bounded), we can derive the equivalent minimization form and the discretized problem. These assumptions will be addressed later.

14.3 Discretization in multi-dimensions

We wish to generalize the P_1 -elements where $u \in P_1$ is a piece-wise linear function that is defined by values on a given set of points. In the case of two dimensions, a linear function that supports on a triangle can be determined by the values at the three vertices, or equivalently in the form $a_i x + b_i y + c_i$ where i indexes the triangle element. A prerequisite to this approach is to compute a triangulation of Ω , namely to partition Ω into a set of triangles properly. Then, we seek for basis functions $\{\varphi_j(x, y)\}$ with a small support that vanishes at all nodes but (x_j, y_j) .

Thus, we define the finite element function space as

$$V_h := \left\{ \sum_{j=1}^J \alpha_j \varphi_j \right\} \tag{14.3.1}$$

and the corresponding weak form of the finite element variational form

$$\text{find } u_h \in V_h \subset H_0^1 \text{ s.t. } a(u_h, v_h) = F(v_h) \, \forall v_h \in V_h \quad (V_h).$$

By bi-linearity of a , (V_h) is also equivalent to

$$\text{find } \{\alpha_j\} \text{ s.t. } \sum_{j=1}^J \alpha_j a(\varphi_j, \varphi_k) = F(\varphi_k) \, \forall k.$$

We quickly verify that $V_h \subset H_0^1$ as is defined in Eqn. 14.3.1. Since φ_j is an affine function, it is naturally in L^2 ; the gradient $\nabla \varphi_j$ is (piece-wise) constant, thus also in L^2 .¹

The evaluation of $a(\varphi_j, \varphi_k)$ is typically done separately on each triangular element. In this example, this term vanishes if node j and node k are not connected by a common edge, leading to a sparse structure. We shall point out that the evaluation of $\int_{\Omega} (\nabla \varphi_j) \cdot (\nabla \varphi_k) \, dx$ involves

¹In fact, we should also make sure that $\nabla \left(\sum_{j=1}^J \alpha_j \varphi_j \right)$ does not introduce delta functions; this can be checked by arguing that φ_j is a continuous function.

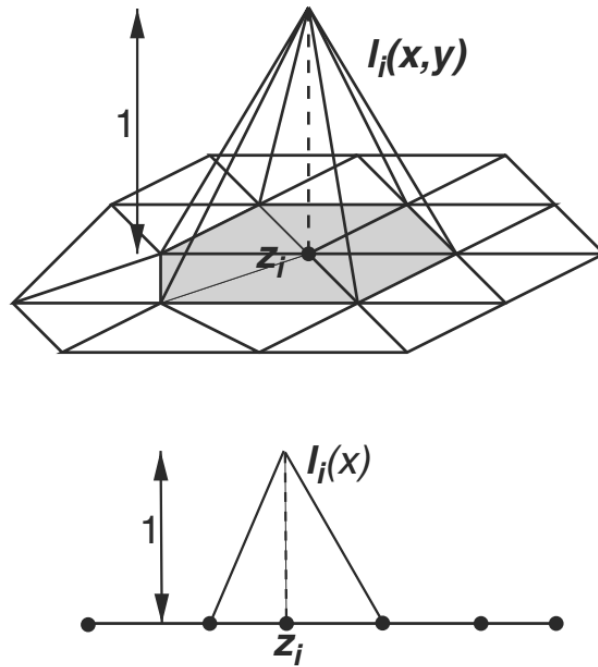


Figure 14.3.1: Shape of φ_j in 2 and 1 dimensional cases. From “Numerical Analysis” page 354, Figure 8.7 right.

some numerical quadratures in general (although it reads integrating a constant in this particular example); it is impractical to develop such quadrature for every triangular element, so it is preferred to map the element to a standard form and develop numerical integration methods there, e.g. the Gaussian quadrature.

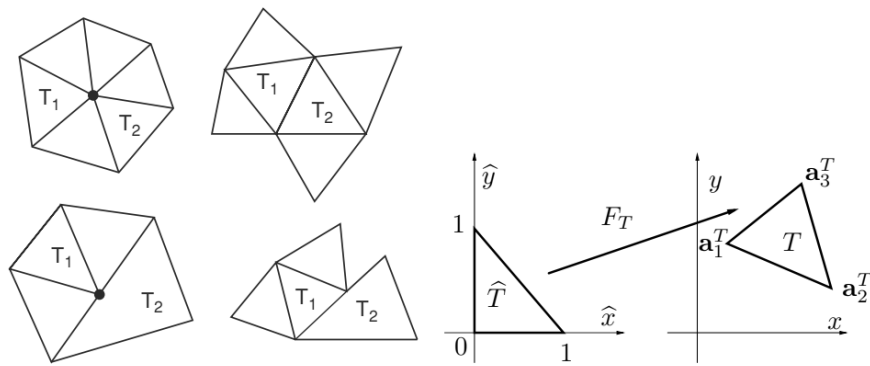


Fig. 8.6. The left side picture shows admissible (*above*) and nonadmissible (*below*) triangulations while the right side picture shows the affine map from the reference triangle \hat{T} to the generic element $T \in \mathcal{T}_h$

Figure 14.3.2: From “Numerical Analysis”, page 353.

Mar 6: Lecture 15

Numerical Partial Differential Equations (Part VII: Practical Concerns for FEM)

15.1 Meshes

Recall the general outline of the FEM approach: for the model problem

$$\begin{aligned} -\Delta u &= f(x) & x \in \Omega, \\ u &= 0 & x \in \partial\Omega, \end{aligned}$$

we derive the weak form $a(u, v) = F(v)$. To solve it numerically, we pick a set of basis functions $\{\varphi_j\} \subset H_0^1(\Omega)$ which spans the space V_h of trial (and also test) functions, i.e. we seek for solutions in the form $u = \sum_j \alpha_j \varphi_j$ and test the variational form for each φ_k . This leads to a system of linear equations $\sum_j \alpha_j a(\varphi_j, \varphi_k) = F(\varphi_k)$.

The set of basis functions is relied on the mesh that lies on the underlying domain. Triangulation is often needed to generate such mesh, although sometimes rectangular partitions are also viable. The topology of the mesh is defined by the matching nodes and edges that are shared between elements and we usually require admissible partitions (see Fig. 15.1.1). The quality of a triangulation is measured by how well functions can be approximated by the numerical subspace which is often determined by two major factors:

1. Geometry of each element: “regular” triangle elements are “better” than “irregular” ones that look too flat or spread out; this can be measured by the ratio between the diameter and the

radius of the inscribed circle;

- Density of the mesh: high density, or equivalently smaller element diameter, is often needed for the area where $\nabla^2 u$ is “larger” to lower the approximation error.

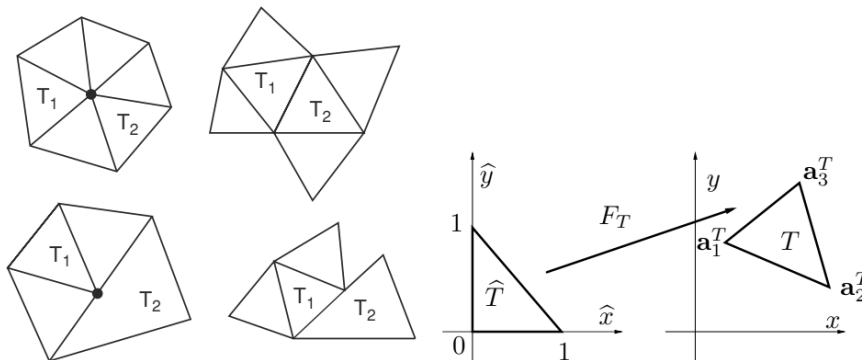


Fig. 8.6. The left side picture shows admissible (*above*) and nonadmissible (*below*) triangulations while the right side picture shows the affine map from the reference triangle \hat{T} to the generic element $T \in \mathcal{T}_h$

Figure 15.1.1: From “Numerical Analysis”, page 353.

The meshing algorithm is often a challenging problem by itself since there is no universal approach. Practical implementations usually depend on different types of heuristics and/or CAD-files that defines the surface of the domain. A general guideline is to create new partitions based on the remaining volume of the domain/element. This often leads to a recursive partitioning algorithm that starts from a regular mesh and finer adjustments. There are more constraints to consider, however, for example boundary matching and discontinuity handling.

15.2 Assembly of linear system

The linear system, derived from testing the variational form, can be written as

$$\begin{pmatrix} \ddots & & & \\ & a(\varphi_j, \varphi_k) & & \\ & & \ddots & \\ & & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ \alpha_k \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ f(\varphi_j) \\ \vdots \end{pmatrix}$$

where the first term on the left is referred to as the stiffness matrix. Let us examine the entries more closely

$$a(\varphi_j, \varphi_k) = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_k \, dx = \sum_{i \in I} \int_{K_i} \nabla \varphi_j \cdot \nabla \varphi_k \, dx$$

where i is the index of each triangle element K_i . A data structure is thus required to hold the structure between nodes and elements and edges. We demonstrate this by using an array.

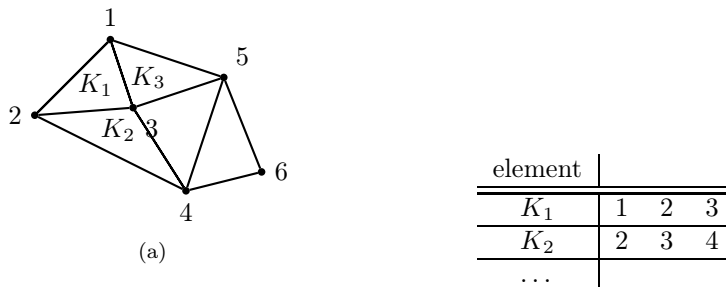


Figure 15.2.1

To improve performance, we can calculate the bilinear form for each K_i separately and memorize the contribution to each node. For example, the K_1 element has three nodes: 1, 2, and 3. Other hat functions will not be involved with this business because they are not supported on K_1 . Then, we can calculate the local stiffness matrix

$$\begin{pmatrix} \tilde{a}(\varphi_1, \varphi_1) & \tilde{a}(\varphi_1, \varphi_2) & \tilde{a}(\varphi_1, \varphi_3) \\ & \tilde{a}(\varphi_2, \varphi_2) & \tilde{a}(\varphi_2, \varphi_3) \\ & * & \tilde{a}(\varphi_3, \varphi_3) \end{pmatrix}.$$

The asterisk terms are not needed for symmetric cases, but extra calculation is needed if there is a first order term in the PDE problem.

The evaluation of the local bilinear form $\tilde{a}(\varphi_j, \varphi_k)$ depends on numerical quadratures. To reduce the complexity and avoid proposing a different scheme for each element, it is often preferred to map elements to a standard element (see Fig. 15.1.1). Let us introduce mapping F that maps the standard/reference element \hat{K} to K . Let us put a hat on the coordinate to denote the coordinate in the reference space. By the formula of changing coordinates,

$$\int_K g(x) \, dx = \int_{\hat{K}} \hat{g}(x) |\det J_F| \, d\hat{x}$$

where we put $\hat{g}(x) := g(F(\hat{x}))$

$$J_F := \begin{pmatrix} \frac{\partial F_1}{\partial \hat{x}_1} & \frac{\partial F_1}{\partial \hat{x}_2} \\ \frac{\partial F_2}{\partial \hat{x}_1} & \frac{\partial F_2}{\partial \hat{x}_2} \end{pmatrix}.$$

Now, we move to the reference space and study the Gaussian quadrature formulae. The 1-node version reads

$$\frac{1}{2} \hat{g}\left(\frac{1}{3}, \frac{1}{3}\right)$$




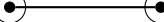
which is accurate for linear \hat{g} . The 3-node version reads

$$\frac{1}{6} \left[\hat{g} \left(\frac{1}{6}, \frac{1}{6} \right) + \hat{g} \left(\frac{1}{6}, \frac{2}{3} \right) + \hat{g} \left(\frac{2}{3}, \frac{1}{6} \right) \right]$$

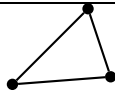
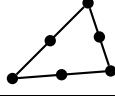
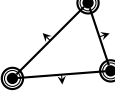
which is accurate for quadratic \hat{g} .

Remark 15.2.1. It is also possible to propose non-linear mapping F to handle curved boundaries in K_i . Since linear functions are ruled out from the picture, F is at least quadratic; in this case, we can pick the basis functions from P_2 elements (a.k.a. Lagrange form). Recall that P_1 elements can be viewed as elevating the element only at the specific node. For P_2 elements, the basis functions have six nodes to interpolate. Compared to P_1 elements, they look more “concentrated” at the nodes and “inflated” at the midpoint of edges. It is also easy to verify that P_2 elements are continuous by the Lagrange interpolation formula.

Let us compare difference ways of constructing elements: for 1d problems,

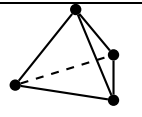
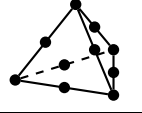
element space	illustration	d.o.f.	regularity
P_1		2	C^0
P_2		3	C^0
P_3		4	C^0
\tilde{P}_3		4	C^1

For the 2d case,

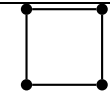
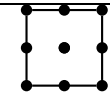
element space	illustration	d.o.f.	regularity
P_1		3	C^0
P_2		6	C^0
P_5 (Argyris element)		21 ¹	C^1

For the 3d case, we omit the C^1 element since it is too complicated.

¹All 0th, 1st, and 2nd derivatives at each node are needed, along with the normal derivative at the midpoint, leading to $(1 + 2 + 3) \times 3 + 3 = 21$ degrees of freedom

element space	illustration	d.o.f.	regularity
P_1		4	C^0
P_2		10	C^0

Rectangular elements are also possible if the mesh does not need to come from triangulation, allowing more flexibility.

element space	illustration	d.o.f.	regularity
Q_1 (bilinear)		4	C^0
Q_2		9	C^0

Mar 8: Lecture 16

Numerical Partial Differential Equations (Part VIII: Remarks on FEM and Parabolic Problems)

16.1 Computational issues of FEM

In the previous lectures, we have covered how to assemble the linear system by testing the variational form. It remains a problem how these systems are solved efficiently.

- Direct solvers: applied for 1D, most 2D, and sometimes 3D problems. This class of methods includes Gaussian elimination, Cholesky decomposition.
- Iterative solvers: Krylov subspace methods (e.g. Conjugate gradient for symmetric problems, GMRES otherwise); used for some 2D and most 3D problems.
- Multigrid solvers: geometric multigrid solvers utilize the structure of the underlying mesh to efficiently propagate the residual error; there is also an algebraic version that decomposes the solution space.

There are also other techniques if we have access to special properties of the coefficient matrix, e.g. sparsity. For example, for the Poisson problem, the matrix is tridiagonal in 1D and block-tridiagonal in 2D; it is banded in general.

For direct solvers, the cost to solve a banded system of size n and band width m is usually $\mathcal{O}(nm^2)$, so it is crucial to keep m small. This is not an issue for 1D problems since m does not scale with the discretization, but it does scale for higher dimension problems. The root cause of

scaling is that the nodes are more “closely connected” as the dimension increases. To circumvent the fill-in effect, it is sometimes necessary to reorder the nodes so that the connection is structured in a hierarchical fashion.

Compared to these direct solvers, the iterative solvers are not affected by the bandwidth since the matrix-vector product scales with the size n only. However, the trade-off is that one needs to apply the outer iteration before the solver converges.

Similarly to non-linear systems, Newton or quasi-Newton solvers are also possible, but we postpone this to the future lectures.

It is also possible to generalize the aforementioned methods, say to time dependent systems (parabolic, hyperbolic PDEs) or initial value problems. Mixed methods are also developed for problems with a special structure. For example, the Stokes equation assumes that the flow is incompressible, i.e. the solution is divergence-free. This requires a careful design on the finite element space to preserve the desired properties.

16.2 Parabolic problems

Let us consider the model problem that describes how heat propagates in a medium,

$$\begin{aligned} u_t &= \Delta u & x \in \Omega, t > t_0 \\ u(x, t) &= 0 & x \in \partial\Omega, t > t_0 \\ u(x, t_0) &= u_0(x) & x \in \Omega \end{aligned}$$

This equation characterizes the inner temperature where the material is attached to a thermal source with constant temperature on both ends. Other boundary conditions are also possible. The Neumann condition $\partial_n u = 0$ indicates that the heat flux can not escape from the (insulated) boundaries, corresponding to an adiabatic environment.

The weak form can be derived by testing against different test functions. We can pick test functions that is supported on the time and space domain at the same time (space-time FEM), but it is only applied in a few special cases. The “standard” approach is to use only space test functions. Similarly to what we did in the elliptic case,

$$\begin{aligned} \int_{\Omega} u_t(x, t) v(x) \, dx &= - \int_{\Omega} \nabla_x u(x, t) \nabla_x v(x) \, dx, \\ u(x, t_0) &= u_0(x). \end{aligned}$$

If we assume that the basis function evolve independently to each other and put $u(x) \approx \sum_j \alpha_j(t) \varphi_j(x)$,

it leads to

$$\sum_j \alpha_j'(t) \underbrace{\int_{\Omega} \varphi_j(x) \varphi_k(x) dx}_{=:B} = - \sum_j \alpha_j \underbrace{\int_{\Omega} \nabla_x \varphi_j(x) \nabla_x \varphi_j(x) dx}_{=:A}$$

which can be written as a first order form

$$B\alpha' = -A\alpha, t > t_0. \quad (16.2.1)$$

The initial condition is determined by finding the best $\alpha_j(t_0)$ such that the finite element space best approximates the given initial condition.

Remark 16.2.1. We shall point out that there is an implicit time marching even for formally explicit time method (such as Euler). This is due to the fact that it is not practical to invert the sparse matrix B in Eqn. 16.2.1 since B^{-1} may not be sparse. Thus, implicit methods are often applied to parabolic problems for this reason.

Mar 20: Lecture 17

Numerical Partial Differential Equations (Part IX: Time Dependent Problems and Mixed Methods)

17.1 Time dependent problems

Let us continue on the heat equation where we seek solutions to

$$\begin{aligned}u_t &= \Delta u & x \in \Omega, t > t_0 \\u(x, t) &= 0 & x \in \partial\Omega, t > t_0 \\u(x, t_0) &= u_0(x) & x \in \Omega.\end{aligned}$$

As we have mentioned earlier, it is possible to apply full FEM to this problem which means to pick test functions that depends on t and x . However, the more practical choice is to leave the time derivative to be a strong derivative and apply FEM to space variables only. This leads to the so-called semi-weak form: find $u \in H_0^1(\Omega) \times C^1(t_0, T)$ ¹ s.t.

$$\begin{aligned}\int_{\Omega} u_t v \, dx + \int_{\Omega} \nabla u \cdot \nabla v \, dx &= 0 \\u(x, t_0) &= u_0(x)\end{aligned}$$

for all $v \in H_0^1(\Omega)$.

¹A better notation could be $C^1((t_0, T); H_0^1(\Omega))$

17.1.1 Space first, then time

Discretization can be done in two equivalent fashions. The first approach handles space variable first, then time variable. We put u_h as a superposition of a given set of basis functions φ_j :

$$u_h(x, t) = \sum_{j=1}^J \alpha_j(t) \varphi_j(x)$$

where the discretized test function v_h may take $\varphi_k, k = 1, \dots, J$. This leads to the following system

$$\sum_{j=1}^J \dot{\alpha}_j(t) \int_{\Omega} \varphi_j \varphi_k \, dx + \sum_{j=1}^J \alpha_j(t) \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_k \, dx = 0, j = 1, \dots, J. \quad (17.1.1)$$

If we put $\vec{\alpha} = (\alpha_1, \dots, \alpha_J)^T$, $B_{kj} = \int_{\Omega} \varphi_j \varphi_k \, dx$ and $A_{kj} = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_k \, dx$, Eqn. 17.1.1 reduces to

$$B \vec{\alpha}' + A \vec{\alpha} = 0. \quad (17.1.2)$$

The initial condition $\vec{\alpha}(t_0)$ is determined by approximating the given function u_0

$$u_0(x) \approx \sum_{j=1}^J \alpha_j(t_0) \varphi_j(x).$$

We shall point out that applying FDM to Eqn. 17.1.2 leads to an implicit scheme in general, regardless of the type of the FDM chosen.

17.1.2 Time first, then space

Another approach is to discretize time first and apply Galerkin formulation later. The idea is to work directly on the slice at each time-step t_n . Let us put $u_n(x) \approx u(x, t_n)$. A popular choice is to apply trapezoidal rule and the corresponding PDE scheme is known as Crank-Nicolson method. To be specific, the time derivative u_t is approximated by $\frac{1}{\Delta t}(u_{n+1} - u_n)$ while the gradient term ∇u is replaced by $\frac{1}{2}(\nabla u_{n+1} + \nabla u_n)$, leading to

$$\int_{\Omega} \frac{1}{\Delta t} (u_{n+1} - u_n) v \, dx + \int_{\Omega} \frac{1}{2} (\nabla u_{n+1} + \nabla u_n) \cdot \nabla v \, dx = 0. \quad (17.1.3)$$

Once u_n is known or solved, we can solve u_{n+1} from Eqn. 17.1.3 as an elliptic problem.

Eqn. 17.1.3 is also useful for stability analysis. The usual trick is to pick v as the combination

of $\{u_n\}$, which reads $\frac{1}{2}(u_{n+1} + u_n)$ in this case. This substitution yields

$$\int_{\Omega} \frac{1}{2\Delta t} (u_{n+1}^2 - u_n^2) \, dx + \int_{\Omega} \frac{1}{4} |(\nabla u_{n+1} + \nabla u_n)|^2 \, dx = 0.$$

Notice that the second integral is always non-negative, so the first integral is never positive, leading to

$$\|u_{n+1}\|_{L_2}^2 = \int_{\Omega} u_{n+1}^2 \, dx \leq \int_{\Omega} u_n^2 \, dx = \|u_n\|_{L_2}^2,$$

implying that the numerical L_2 norm is non-increasing unconditionally. This can be combined with the FEM-elliptic estimate for controlling error in space discretization, leading to the overall error estimate.

17.2 Mixed methods

The general underlying idea is to adopt different spaces and basis functions for different components in the PDE. To illustrate the point, we study the Stokes equation that describes incompressible slow stationary flow, which can be viewed as a particular case of Naive-Stokes equation with ∂_t and $(u \cdot \nabla)u$ dropped. The system reads

$$\begin{aligned} -\mu\Delta u + \nabla p &= f & x \in \Omega, \\ \nabla \cdot u &= 0 & x \in \Omega, \\ u &= 0 & x \in \partial\Omega \end{aligned}$$

where $u = (u^{(1)}(x_1, x_2), u^{(2)}(x_1, x_2))$ stands for velocity and $p(x_1, x_2)$ is a scalar field that represents pressure. Notice that we require the divergence-free condition for incompressibility; a direct approach is to build this into the basis function by explicitly constructing the function space wisely. This can be done in some special cases where we have some information of the solution, say singularity etc. To derive the weak form in this spirit, we use two space-test functions $v^{(1)}, v^{(2)}$, leading to

$$\mu \int_{\Omega} \nabla u^{(k)} \cdot \nabla v^{(k)} \, dx + \int_{\Omega} (\nabla p)^{(k)} v^{(k)} \, dx = \int_{\Omega} f^{(k)} v^{(k)} \, dx, \quad k = 1, 2.$$

We add the two equations (corresponding to $k = 1, 2$) and apply IBP to the $\int_{\Omega} (\nabla p)^{(k)} v^{(k)} \, dx$ term

$$\mu \int_{\Omega} \left[\nabla u^{(1)} \cdot \nabla v^{(1)} + \nabla u^{(2)} \cdot \nabla v^{(2)} \right] \, dx - \int_{\Omega} p (\nabla \cdot v) \, dx = \int_{\Omega} f \cdot v \, dx.$$

If we coerce v also a divergence-free function, then the second integral vanishes, leading to a standard elliptic problem. To make this rigorous, we introduce the space that holds u and v :

$$\tilde{H} := \left\{ \left(v^{(1)}, v^{(2)} \right) : v^{(k)} \in H_0^1(\Omega), \nabla \cdot v^{(k)} = 0, k = 1, 2 \right\}.$$

The variational form reads

$$\text{find } u \in \tilde{H} \text{ s.t. } \mu \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f \cdot v \, dx \quad \forall v \in \tilde{H}.$$

We point out that it is not numerically practical since elements in \tilde{H} are not easy to obtain due to the divergence-free condition, even in the simplest 2D case. This motivates the mixed methods. We couple the variational form

$$\mu \int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\Omega} p(\nabla \cdot v) \, dx = \int_{\Omega} f \cdot v \, dx \quad (17.2.1)$$

with

$$\int_{\Omega} (\nabla \cdot u) q \, dx = 0 \quad (17.2.2)$$

to compensate the lack of the divergence-free requirement. Thus, the variational problem is to find $u \in H_0^1$ and $p \in L^2$ s.t. Eqn. 17.2.1 and 17.2.2 hold for any $v \in H_0^1, q \in L^2$.

The standard theory, however, does not apply directly due to the lack of coercivity of the variational form which is also replaced by inf-sup condition

$$\inf_{q_h \in Q_h} \left[\sup_{v_h \in V_h} \frac{\int q_h (\nabla \cdot v_h) \, dx}{\|v_h\|_{H^1} \|q_h\|_{L^2}} \right] \geq C > 0.$$

This also asks for a stronger regularity of the discretized function space since the common choice ($u_h, v_h \in P_1, p, q \in P_0$) does not converge. The lowest common requirement is to use $u_h, v_h \in Q_2$ and $p, q \in Q_0$.

Another example that demonstrates the mixed method is to put the Poisson equation

$$\begin{aligned} -\Delta u &= f & x \in \Omega \\ u &= 0 & x \in \partial\Omega \end{aligned}$$

as a first-order coupled system

$$\begin{aligned} \sigma &= \nabla u & x &\in \Omega \\ -\nabla \cdot \sigma &= f & x &\in \Omega \\ u &= 0 & x &\in \partial\Omega. \end{aligned}$$

We will continue on this case in the next lecture.

Mar 22: Lecture 18

Numerical Partial Differential Equations (Part X: Mixed Methods and FDM)

18.1 Mixed methods (cont'd)

Let us study another application of the mixed methods. The Poisson equation $-\Delta u = f$ can be put equivalently as $-\nabla \cdot \sigma = f$ if we identify $\sigma = \nabla u$. The weak form reads

$$\begin{aligned} \int_{\Omega} \sigma \cdot \tau + u (\nabla \cdot \tau) \, dx &= 0 \\ - \int_{\Omega} (\nabla \cdot \sigma) v \, dx &= \int_{\Omega} f v \, dx \end{aligned}$$

for all proper test functions τ, v . One motivation to use this alternative form is that we can lower the regularity requirement on u (notice that we take no derivative on u in this weak form). An appropriate choice of the function spaces is $u, v \in L^2(\Omega)$ and

$$\sigma, \tau \in H(\operatorname{div}) := \left\{ \varphi \in [L^2(\Omega)]^d : \|\operatorname{div} \varphi\|_{L^2} < \infty \right\}.$$

There are some caveats to this formulation: there is no standard coercivity (replaced by inf-sup conditions) and the numerical discretization space V_h needs to satisfy more conditions than simply being a subspace. One numerical realization of the aforementioned function space is to pick $u, v \in P_0$ (piece-wise constant functions) and σ, τ as the so-called Raviart-Thomas elements (a.k.a. edge elements); these elements are defined along an edge that involves a pair of triangular elements

(where the configuration is shown in Fig. 18.1.1)

$$\varphi(x) := \begin{cases} C(x - p^+) & x \in T_+ \\ C(x - p^-) & x \in T_- \end{cases}.$$

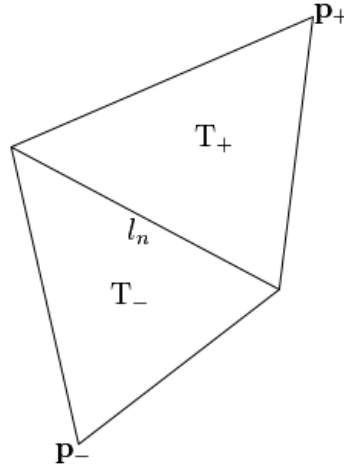


Figure 18.1.1: Raviart-Thomas basis functions, from wikipedia.

18.2 FDM for PDEs

As we have done to the ODE systems, we can also approximate the PDE solutions by the node values and use finite difference to replace the differentials in the equation. We point out that this is similar to Lagrange elements in the FEM framework since point-wise evaluations of solutions are focused. For example, if we extract the values $u(x_i, y_j)$ on a rectangular mesh, the finite difference reads

$$-\Delta u \approx - \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} \right)$$

which is often known as the five-point stencil. The local truncation error can be derived via Taylor's expansion (similarly to the ODE case) and possesses the form $\max |u^{(4)}| (\Delta x^2 + \Delta y^2)$. The discretized linear system also exhibits a fractional diagonal structure, similar to one from FEM. In fact, without loss of generality we assume $\Delta x = \Delta y = h$, then the discretized equation

reads

$$4u_{i,j} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = h^2 f_{i,j}.$$

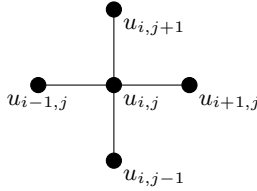


Figure 18.2.1: The five point stencil.

To translate this system into the matrix-vector form, we adopt the row-wise ordering, i.e.

$$U := (u_{1,1}, u_{1,2}, \dots, u_{2,1}, u_{2,2}, \dots, u_{n,1}, \dots, u_{n,n})^T.$$

Then, the matrix coefficient is filled in the following manner

$$A = \begin{pmatrix} 4 & -1 & & \dots & -1 & & \dots \\ -1 & 4 & -1 & & & & -1 \\ & -1 & 4 & & & & -1 \\ \vdots & \vdots & \vdots & \ddots & & & \\ -1 & & & & \ddots & & \\ & -1 & & & & \ddots & \\ & & -1 & & & & \ddots \\ \vdots & & & & & & \ddots \end{pmatrix}.$$

Generally speaking, FDM is easier to implement for domains of regular geometries and free from the need to calculate local integrals. Nevertheless, there are some drawbacks to this method.

- FDM does not work well with curved boundaries since the boundary might not align with the assumed grid well. There are several ways to circumvent this issue: one is to extrapolate the grid value so that it extends to the boundary, but this can lead to oscillatory behaviors. Another approach is to alter the curved boundary into the artificial, zig-zag shape one and develop approximation theory there.
- FDM also suffers from Neumann BC since the normal direction might not align with the mesh, let alone the discrete finite normal difference might lead to numerical difficulties.

One general strategy to circumvent the domain issues is to transform the domain to a rectangular

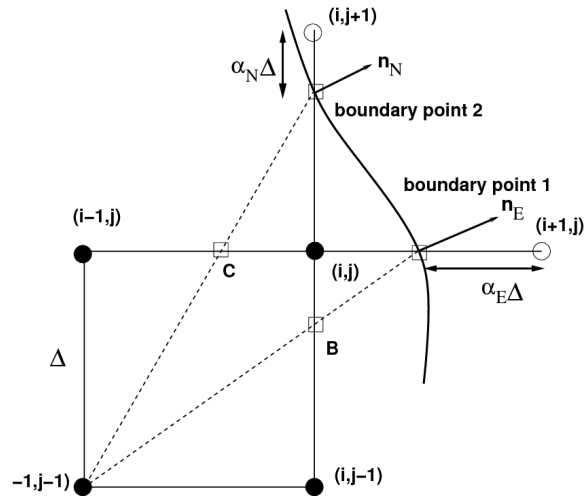


Figure 18.2.2: An illustration of one irregular boundary handling scheme. From “Numerical solution of the 2D Poisson equation on an irregular domain with Robin boundary conditions”, 08’ Z. Jomaa, C. Macaskill.

shape before applying FDM.

18.2.1 Analysis to FDM

One strategy to derive error estimate is to start with the discretized system $AU = h^2 f$ and examine the spectrum of A (non-singularity, eigen gap...); this is valid but not very practical. Another approach is the so-called energy method which mimics the FEM analysis. To illustrate the point, let us examine the 1-D system $-u_{xx} + u = f$ with $u(x_L) = u(x_R) = 0$. The three-point stencil leads to

$$-\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right) + u_i = f_i. \quad (18.2.1)$$

Let us define some discrete operators

- Translation $Tu_i = u_{i+1}$,
- Forward difference $\Delta_+ := T - I$, i.e. $\Delta_+ u_i = u_{i+1} - u_i$, and
- Backward difference $\Delta_- := I - T^{-1}$, i.e. $\Delta_- u_i = u_i - u_{i-1}$.

With those notations, we can put Eqn. 18.2.1 as

$$-(\Delta_+ \Delta_- u_i) + h^2 u_i = h^2 f_i. \quad (18.2.2)$$

Let us multiply Eqn. 18.2.2 by u_i and sum over all interior nodes:

$$\sum_{i=1}^{J-1} [-(\Delta_+ \Delta_- u_i) + h^2 u_i] u_i = h^2 \sum_{i=1}^{J-1} f_i u_i.$$

We mimic the “integration by parts” to achieve “summation by parts”:

$$\begin{aligned} \sum_{i=0}^{J-1} (\Delta_+ a_i) b_i &= \sum_{i=0}^{J-1} (a_{i+1} - a_i) b_i = \sum_{i=1}^J a_i b_{i-1} - \sum_{i=0}^{J-1} a_i b_i. \\ &= \sum_{i=1}^J a_i (b_{i-1} - b_i) + a_J b_J - a_0 b_0 \\ &= ab|_0^J - \sum_{i=1}^J a_i (\Delta_- b_i). \end{aligned}$$

Now, if we identify $a = \Delta_- u$ and $b = u$, we have

$$\sum_{i=1}^J (\Delta_- u_i)^2 + h^2 \sum_{i=1}^{J-1} u_i^2 = h^2 \sum_{i=1}^{J-1} f_i u_i.$$

Although we use finite difference to approximate differentials, we are not aiming for a H^1 error estimate here, so we silently drop the first term and apply Cauchy’s inequality to the RHS, leading to

$$\left| \sum_{i=1}^{J-1} u_i^2 \right| \leq \left| \sum_{i=1}^{J-1} f_i^2 \right|.$$

This can be abbreviated as $\|u\|_{L^2(1, J-1)} \leq \|f\|_{L^2(1, J-1)}$ if we put

$$\|u\|_{L^2(1, J-1)} := \sqrt{h \sum_{i=1}^{J-1} u_i^2}.$$

Mar 27: Lecture 19

Numerical Partial Differential Equations (Part XI: FDM & Stability Analysis)

19.1 FDM (cont'd)

Let us continue on FDM for PDE problems.

Example 19.1.1. Consider

$$\begin{aligned} -\Delta u + u &= f(x) & x \in \Omega, \\ u &= 0 & x \in \partial\Omega \end{aligned}$$

inside a rectangular domain $\Omega \subset \mathbb{R}^2$ (where the general case will be addressed later).

In the spirit of finite difference, we use point-wise value U_{j_1, j_2} to represent the function $u(x)$ and use divided difference $\Delta_{\pm}^{(1,2)}$ to approximate the true differential, i.e.

$$-\left(\frac{\Delta_+^{(1)} \Delta_-^{(1)}}{(\Delta x_1)^2} + \frac{\Delta_+^{(2)} \Delta_-^{(2)}}{(\Delta x_2)^2} \right) U_j + U_j = f(x_j)$$

and $U_j = 0$ for index j that corresponds to boundary nodes. The use of $\Delta_+ \Delta_-$ makes sure that the discretized system is symmetric.

In the last lecture, we have shown the L^2 -stability in the 1D case

$$\|U\|_2 \leq \|f\|_2$$

by summation by parts; this is also true for higher dimensions where one needs to handle the generalized summation by parts properly. We also point out that this stability result can be used to show convergence by LTE analysis, similarly to the convergence proof for the ODE solver. In fact, we can apply FDM to $u(x_j) - U_j$, leading to

$$-\left(\frac{\Delta_+^{(1)} \Delta_-^{(1)}}{(\Delta x_1)^2} + \frac{\Delta_+^{(2)} \Delta_-^{(2)}}{(\Delta x_2)^2}\right) (u(x_j) - U_j) + (u(x_j) - U_j) = \text{LTE}_j,$$

thus

$$\|u(x_j) - U_j\|_2 \leq \|\text{LTE}\|_2 = \mathcal{O}\left((\Delta x_1)^2 + (\Delta x_2)^2\right)$$

where we recall that $\|U\|_2 = \sqrt{h^2 \sum_j U_j^2}$.

Compared to FEM, it is actually easier to formulate a FDM scheme and in fact more accurate (due to strong conditions on f). However, it is much harder to generalize to arbitrary domain for FDM, where FEM can work with any domain as long as the mesh is defined. There are a few workarounds for arbitrary shapes to work with FDM. One may attempt to pick some random points in the interior and interpolate the values locally to estimate the Laplacian Δu , but there is little theory for convergence for this approach. Alternatively, one can enforce zero values on the nearest node points to the boundary and adopt a similar l^2 -stability analysis; this, however, leads to a lower accuracy. The other approach is to extrapolate the values to the intersection of the curved boundary with the mesh grids but this is also more tedious. Neumann BC is also much more difficult to deal with.

19.2 FDM for IVP

Example 19.2.1. We recall the heat equation

$$\begin{aligned} u_t &= u_{xx} & t > t_0, x_L < x < x_R, \\ u(x_L, t) &= u(x_R, t) = 0 & t > t_0, \\ u(x, t_0) &= u_0(x) & x_L < x < x_R. \end{aligned}$$

Once again, we use point-wise value U_j^n to approximate $u(x_j, t_n)$ on a uniform mesh $x_j = x_0 + j\Delta x$ and $t_n = t_0 + n\Delta t$. There are various choices for the divided differences (as shown in

FDM and time-dependent FEM), and we adopt the simplest one for showcase

$$\begin{aligned} \frac{U_j^{n+1} - U_j^n}{\Delta t} &= \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} && \text{(interior)} && (19.2.1) \\ U_0^n = U_J^n &= 0 && \text{(BC)} \\ U_j^0 &= u_0(x_j) && \text{(IC)} \end{aligned}$$

We can call this a four-point stencil, compared to the standard five-point stencil in the elliptic problems.

19.3 Techniques for analyzing convergence

The convergence, requiring vanishing LTE as $\Delta x, \Delta t \rightarrow 0$, follows a similar analysis based on Taylor expansion. For stability analysis, there are three techniques: direct norm estimate (that yields a sufficient condition), comparison of dependent domain (that gives arise to a necessary condition), and von Neumann analysis (that yields a sufficient and necessary condition).

19.3.1 Direct norm estimate

If we put $\alpha := \frac{\Delta t}{\Delta x^2}$, then Eqn. 19.2.1 reduces to

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

Thus, the l^∞ -norm can be estimated via

$$\|U^{n+1}\|_\infty := \max_j |U_j^{n+1}| \leq (2\alpha + |1 - 2\alpha|) \max_j |U_j^n| = \|U^n\|_\infty.$$

To ensure that the l^∞ -norm is non-exploding, α needs to be no larger than 1/2. The method of induction implies that

$$\|U^n\|_\infty \leq \|U^0\|_\infty = \|u_0\|_{L^\infty},$$

i.e. continuous dependence on initial data. Notice that this analysis produces a sufficient condition, but it turns out to be a necessary condition as well from the von Neumann analysis.

For the l^2 -analysis, we combine the techniques in analyzing Crank-Nicolson for FEM and summation by parts. The C-N FDM scheme reads

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

i.e.

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{1}{2\Delta x^2} \Delta_+ \Delta_- (U_j^{n+1} + U_j^n). \quad (19.3.1)$$

Mimicking the L^2 -analysis in FEM, we multiply Eqn. 19.3.1 with $U_j^{n+1} + U_j^n$ and apply summation by parts, leading to

$$\|U^{n+1}\|_2^2 - \|U^n\|_2^2 = -\frac{\alpha}{2} \|\Delta_- (U_j^{n+1} + U_j^n)\|_2^2 \leq 0.$$

Thus, the l^2 -norm $\|U^n\|_2$ is non-increasing.

Mar 29: Lecture 20

Numerical Partial Differential Equations (Part XII: Stability Analysis)

20.1 Techniques for analyzing convergence (cont'd)

20.1.1 Direct norm estimate (cont'd)

Let us revisit the heat equation with a general conductivity coefficient and a heat source

$$\begin{aligned} u_t &= \sigma(x) \Delta u + f(x, t) & t > 0, x_L < x < x_R \\ 0 &= u(x_L, t) = u(x_R, t) & t > 0 \\ u_0(x) &= u(x, t_0) & x_L < x < x_R. \end{aligned}$$

We adopt a FDM discretization scheme

$$U_j^{n+1} = \alpha_j U_{j+1}^n + (1 - 2\alpha_j) U_j^n + \alpha_j U_{j-1}^n + \Delta t f_j^n \quad (20.1.1)$$

where $\alpha_j := \sigma(x_j) \frac{\Delta t}{\Delta x^2}$. The local stability result

$$\|U^{n+1}\|_\infty \leq \|U^n\|_\infty + \Delta t \|f^n\|_\infty \quad (20.1.2)$$

holds as long as $\max_j |\alpha_j| \leq \frac{1}{2}$, albeit α might vary along the mesh. Adding Eqn. 20.1.2 for $n = 1, \dots, N$ leads to

$$\|U^N\|_\infty \leq \|u_0\|_\infty + T \max_n \|f^n\|_\infty$$

for $T := N\Delta t$. This can be used to derive an error estimate for the scheme (Eqn. 20.1.1) since the heat equation is posed in a linear form. Let

$$V_j^n := u(x_j, t_n) - U_j^n,$$

then V_j^n solves the following discrete system

$$V_j^{n+1} = \alpha_j V_{j+1}^n + (1 - 2\alpha_j) V_j^n + \alpha_j V_{j-1}^n + \Delta t \text{LTE}_j^n$$

which leads to

$$\|V^N\|_\infty \leq T \max_n \|\text{LTE}^n\|_\infty = \mathcal{O}(\Delta t + \Delta x^2).$$

A sufficient stability condition reads

$$\frac{\sigma \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

for $\sigma := \max_x \sigma(x)$.

20.1.2 Domain of dependence

The domain of dependence argument, on the other hand, derives a necessary condition $\Delta t/\Delta x \rightarrow 0$. The numerical dependence domain, if $\Delta t/\Delta x$ does not vanish in the limit, covers a finite range of interval at the initial condition, violating the fact that the heat kernel has infinite support; this naturally leads to the Fourier analysis.

20.1.3 von Neumann analysis

Recall the Fourier transform for Schwartz functions (extended isometrically to $L^2(\mathbb{R})$)

$$\begin{aligned} \widehat{u}(\omega) &:= \int_{-\infty}^{\infty} u(x) e^{-i \cdot 2\pi\omega x} dx, \\ u(x) &= \int_{-\infty}^{\infty} \widehat{u}(\omega) e^{i \cdot 2\pi\omega x} d\omega. \end{aligned}$$

The Fourier transform is useful since the differential operator is closely associated to a scalar function in the frequency domain, i.e.

$$\widehat{\frac{d}{dx} u(x)} = i \cdot 2\pi\omega \widehat{u(x)}. \quad (20.1.3)$$

However, since we are dealing with a finite domain, it is sometimes more useful to consider another version of Fourier transform¹

$$\begin{aligned}\widehat{U}_k &:= \sum_{j=0}^{J-1} U_j e^{-i \cdot 2\pi k j / J}, \\ U_j &:= \frac{1}{N} \sum_{k=0}^{J-1} \widehat{U}_k e^{i \cdot 2\pi k j / J}.\end{aligned}$$

Recall that we have introduced the translation operator $TU_j = U_{j+1}$. The analogs to Eqn. 20.1.3 is given by

$$\begin{aligned}\left(\widehat{TU}\right)_k &= \sum_{j=0}^{J-1} (TU_j) e^{-i \cdot 2\pi k j / J} = \sum_{j=0}^{J-1} U_{j+1} e^{-i \cdot 2\pi k j / J} \\ &= \sum_{j=1}^J U_j e^{-i \cdot 2\pi k (j-1) / J} = e^{i \cdot 2\pi k / J} \widehat{U}_k.\end{aligned}$$

Another handy tool is the so-called Parseval identity

$$\|u(x)\|_{L^2(\mathbb{R})} = \|\widehat{u}(\omega)\|_{L^2(\mathbb{R})}$$

and the corresponding discrete version²

$$\|U\|_2 = \|\widehat{U}\|_2.$$

Back to the heat equations. To keep simplicity, let us assume that $\sigma(x) \equiv \sigma$ being a constant and $f \equiv 0$. We apply DFT to the discrete scheme

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

that yields

$$\widehat{U}_k^{n+1} = \alpha e^{i\xi_k} \widehat{U}_k^n + (1 - 2\alpha) \widehat{U}_k^n + \alpha e^{-i\xi_k} \widehat{U}_k^n \quad (20.1.4)$$

where we put $\xi_j := 2\pi k / J$. Due to the Euler's formula

$$\frac{e^{i\xi_k} + e^{-i\xi_k}}{2} = \cos \xi_k,$$

¹the Fourier basis works with the periodic BC rather than the zero Dirichlet BC posed earlier (where one should actually adopt the sine basis).

²there is sometimes a \sqrt{N} factor due to different definitions of DFT and 2-norms.

Eqn. 20.1.4 is reduced to

$$\widehat{U}^{n+1} = (2\alpha \cos \xi + 1 - 2\alpha) \widehat{U}^n.$$

Since the Parseval's identity implies the equivalence between stability of U^n and \widehat{U}^n , we wish the amplification factor

$$|2\alpha \cos \xi_k + 1 - 2\alpha| \leq 1$$

for any k , leading to the sufficient and necessary condition $\alpha \leq \frac{1}{2}$. If this condition is violated, then the amplification factor for $k^* = \lfloor J/2 \rfloor$ is most likely larger than 1, leading to exploding Fourier coefficients and highly oscillatory numerical solutions (recall that in DFT, k^* corresponds to the highest/finest frequency).

Apr 3: Lecture 21

Numerical Partial Differential Equations (Part XIII: von Neumann Analysis)

21.1 Example on transport equations

Now let us apply von Neumann analysis to hyperbolic equations.

Example 21.1.1. To pose a first order transport equation, we focus on

$$\begin{aligned} u_t + au_x &= 0 & x_L < x < x_R, t > t_0 \\ & u \text{ periodic BC} \\ u(x, t_0) &= u_0(x) & x_L < x < x_R. \end{aligned}$$

21.1.1 A straight-forward but unstable attempt

A naive discretization scheme is to adopt forward Euler in time and central difference in space, leading to

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a}{2\Delta x} (U_{j+1}^n - U_{j-1}^n) = 0. \quad (21.1.1)$$

Let us apply DFT to Eqn. 21.1.1 and recall that translation operator is turned into scalar multiplication in Fourier domain:

$$\hat{U}^{n+1}(\xi) = \left(1 - \frac{a\Delta t}{2\Delta x} (e^{i\xi} - e^{-i\xi}) \right) \hat{U}^n(\xi).$$

The amplification factor

$$\left| 1 - \frac{a\Delta t}{2\Delta x} (e^{i\xi} - e^{-i\xi}) \right| = \sqrt{1 + \left(\frac{a\Delta t \sin \xi}{\Delta x} \right)^2} > 1 \quad (21.1.2)$$

for $\sin \xi \neq 0$, thus this scheme is unconditionally unstable.

21.1.2 Lax-Friedrichs scheme

Eqn. 21.1.2 actually suggests a way to improve stability which is to reduce the real part of the amplification factor. This leads to the Lax-Friedrichs scheme:

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

The local truncation error reads $\mathcal{O} \left(\Delta t + \frac{\Delta x^2}{\Delta t} + \Delta x^2 \right)$. A similar analysis derives the corresponding factor

$$\left| \frac{1}{2} (e^{i\xi} + e^{-i\xi}) - \frac{a\Delta t}{2\Delta x} (e^{i\xi} - e^{-i\xi}) \right| = \sqrt{\cos^2 \xi + \left(\frac{a\Delta t \sin \xi}{\Delta x} \right)^2}$$

which we wish to be no larger than 1, leading to

$$\left| \frac{a\Delta t}{\Delta x} \right| \leq 1.$$

We can also derive this equation from a domain of dependence argument. The transport equation can be solved in closed form as $u(x, t) = u_0(x - at)$, implying that the solution at (x, t) depends on the information of u_0 around $x - at$. If $\frac{a\Delta t}{\Delta x}$ is not properly small, then the numerical dependence domain may not be large enough to cover the characteristic line (see Fig. 21.1.1). This is also known as the Courant-Friedrichs-Lewy condition¹.



Figure 21.1.1: The domain of dependence argument.

¹see [Courant-Friedrichs-Lewy condition](#)

21.1.3 Lax-Wendroff scheme

Another line of improvement is to add artificial viscosity, leading to

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

The local truncation error is $\mathcal{O}(\Delta t^2 + \Delta x^2)$ and the stability condition is $|\frac{a\Delta t}{\Delta x}| \leq 1$ as well.

21.2 General analysis framework

In general, we can put explicit one time step schemes into the following form

$$U_j^{n+1} = \mathcal{L}_\Delta U_j^n = \sum_{k=-K}^K a_k T^k U_j^n$$

where \mathcal{L} is an abstract operator which we assume is the linear combination of a few translation operators. In the Fourier domain, this is translated to

$$\widehat{U}^{n+1} = \widehat{\mathcal{L}}_\Delta \widehat{U}^n = \sum_{k=-K}^K a_k e^{i\xi k} \widehat{U}^n$$

where $\xi := 2\pi\Delta$ that depends on the discretization mesh. The stability condition

$$\|\mathcal{L}_\Delta(\xi)\| \leq 1, \forall 0 \leq \xi < 2\pi$$

is equivalent to

$$\|\widehat{\mathcal{L}}_\Delta(\xi)^n\| \leq C, \forall 0 \leq \xi < 2\pi, n\Delta t \leq T. \quad (21.2.1)$$

Notice that Eqn. 21.2.1 implies a necessary condition

$$\|\widehat{\mathcal{L}}_\Delta(\xi)\| \leq 1, \forall 0 \leq \xi < 2\pi.$$

A similar analysis can be carried out for implicit schemes, where we assume the following form

$$\mathcal{L}_\Delta^{(1)} U_j^{n+1} = \mathcal{L}_\Delta^{(2)} U_j^n.$$

The corresponding stability condition reads

$$\left\| \left\{ \left(\widehat{\mathcal{L}}_\Delta^{(1)} \right)^{-1} \widehat{\mathcal{L}}_\Delta^{(2)}(\xi) \right\}^n \right\| \leq C, \forall 0 \leq \xi < 2\pi, n\Delta t \leq T. \quad (21.2.2)$$

This technique can also be applied to multistep schemes if we go down the rabbit hole. For example, let us consider

$$\mathcal{L}_\Delta^{(1)} U_j^{n+1} = \mathcal{L}_\Delta^{(2)} U_j^n + \mathcal{L}_\Delta^{(3)} U_j^{n-1}$$

and the corresponding form in the Fourier domain

$$\widehat{\mathcal{L}}_\Delta^{(1)}(\xi) \widehat{U}^{n+1} = \widehat{\mathcal{L}}_\Delta^{(2)}(\xi) \widehat{U}^n + \widehat{\mathcal{L}}_\Delta^{(3)}(\xi) \widehat{U}^{n-1}.$$

The techniques from the LMM stability analysis can be also applied here, with a slight caveat that the iteration coefficient depends on the wave number ξ .

21.3 Generalization

We shall point out that the analysis mentioned above has a few restrictions, namely linear equations with constant coefficients and periodic BC. One might ask if we can remove or weaken some of the restrictions.

21.3.1 Variable coefficients

For example

$$\mathcal{L}_\Delta U_j := \sum_{k=-K}^K A_k(x_j) T^k U_j$$

where A_k depends on the mesh nodes so a direct Fourier transform is not viable. Nevertheless, we can formally define

$$\widehat{\mathcal{L}}_\Delta := \sum_{k=-K}^K A_k(x_j) e^{i\xi k}$$

as if A_k were a constant. Then, one can carry out analysis to compare how far $\widehat{\mathcal{L}}_\Delta$ is from the true Fourier translated operator. In fact, we have

Fact 21.3.1 (Lax-Nirenberg Theorem). *If*

$$\widehat{\mathcal{L}}_\Delta(x, \xi) \in C^{2,2}, \left\| \widehat{\mathcal{L}}_\Delta(x, \xi) \right\| \leq 1, \forall x, \xi,$$

then $\|\mathcal{L}_\Delta\| \leq 1 + C\Delta x$.

One can derive a useful method by applying L-N theorem to the L-F scheme, namely solving $u_t + a(x) u_x = 0$ while monitoring the stability condition $\left| \frac{a(x)\Delta t}{\Delta x} \right| \leq 1$.

21.3.2 Non-linearity

In general, linearized stability and smooth solution $u(t, x)$ for consistent schemes implies convergence. For example, the conservation law

$$u_t + f(u)_x = 0$$

is a non-linear equation in general (and we shall point out that the solution might have discontinuities). The linearized equation, under $u = \mathbf{u} + \epsilon v$ with smooth \mathbf{u} and small ϵ , reads

$$(\mathbf{u}_t + f(\mathbf{u})_x) + \epsilon [v_t + (f'(\mathbf{u})v)_x] + \mathcal{O}(\epsilon^2) = 0.$$

If we assume that \mathbf{u} is already a smooth solution, then

$$\underbrace{v_t + f'(\mathbf{u})v_x}_{(*)} + f'(\mathbf{u})_x v + \mathcal{O}(\epsilon) = 0 \quad (21.3.1)$$

where we apply stability analysis for the star part. Take L-F scheme as an example, we assume

$$U_j^n = \mathfrak{U}(x_j, t_n) + \epsilon V_j^n$$

that solves

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

The linearized part implies the first-order matching equation

$$V_j^{n+1} = \frac{V_{j+1}^n + V_{j-1}^n}{2} - \frac{\Delta t}{2\Delta x} f'(\mathfrak{U}(x_j, t_n))_x (V_{j+1}^n - V_{j-1}^n).$$

Then one can apply a similar argument mimicking L-N theorem.

Apr 5: Lecture 22

Numerical Partial Differential Equations (Part XIV: Topics on Non-linear Problems)

22.1 Linearization and convergence

Let us continue the discussion on linearization of non-linear PDEs and stability analysis. To put in a very general sense, the numerical discretization scheme can be written as $F(U) = 0$ while the evaluation on the true PDE solution results in a local error, i.e. $F(\mathfrak{U}) = \text{LTE}$. If we put $e := \mathfrak{U} - U$ and apply Taylor expansion, we have

$$0 = F(U) = F(\mathfrak{U} - e) = F(\mathfrak{U}) - \nabla F(\mathfrak{U}) \cdot e + G(e)$$

where $G(e) \lesssim \|e\|^2$. Thus, the error e can be obtained by solving

$$\nabla F(\mathfrak{U}) \cdot e = \text{LTE} + G(e)$$

which shares the exact same form as the linearized equation [21.3.1](#) as previously derived. Formally speaking,

$$e = [\nabla F(\mathfrak{U})]^{-1} [\text{LTE} + G(e)] \tag{22.1.1}$$

and the stability translates to the equivalence between unique solution to Eqn. 22.1.1 and boundedness of $[\nabla F(\mathfrak{U})]^{-1}$. The consistency argument is then used to show that

$$\|\text{LTE}\| \lesssim L(\Delta) \rightarrow 0 \text{ as } \Delta \rightarrow 0.$$

This can be combined with the fact that G has quadratic growth and fixed point argument to show that $e \rightarrow 0$ as $\Delta \rightarrow 0$.

22.2 Problems with less regularity

In general, the local truncation error is derived from regularity of the PDE solution (similar to the approximation estimate in FEM). However, there are certain equations that possess solutions with “insufficient” regularities.

Example 22.2.1. Non-linear conservation law

$$u_t + f(u)_x = 0$$

in 1D scalar form or for higher dimensions

$$\vec{u}_t + \nabla \cdot \vec{f}(\vec{u}) = 0.$$

These equations are often used in continuum mechanics, fluid dynamics, transport equations, etc.

Example 22.2.2. Hamiltonian-Jacobi equation

$$u_t + H(\nabla u, u) = 0$$

which is widely adopted in control theory and finance related problems.

We shall point out that some HJ equations, for example $u_t + H(\nabla u) = 0$, can be translated to a conservation law. In fact, let $v := u_x$, then

$$v_t = u_{xt} = -\nabla_x H(\nabla u) = -\nabla_x H(v).$$

The importance of such “irregular” equations is that it enables to model phase transition phenomenon. To be specific, say the non-linear conservation law (NLCL), the solutions may develop discontinuities in finite time even for smooth initial data. Thus, the standard method for FDM breaks down due to the exploding local truncation error. For a similar reason, the FEM approach also fails since there is not enough regularity to guarantee approximation properties. The famous

example to illustrate this point is the Burgers' equation, i.e.

$$u_t + \frac{1}{2} (u^2)_x = 0. \tag{22.2.1}$$

If we assume that the solution $u \in C^1$, we have

$$u_t + uu_x = 0.$$

Recall that for a linear transport equation $v_t + av_x = 0$, the solution is transport exactly along the characteristics, i.e. $v(x, t) = v_0(x - at)$. If we assume the initial condition u_0 take value 1 for $x < -1$ and 0 for $x > 1$, then the values 1 and 0 will be kept along the characteristics as the system evolves. However, after a finite time the left line will meet the other vertical line, contradicting with the assumption that $u \in C^1$.

22.3 Numerical methods for conservation laws

To circumvent this issue while keeping the good property on transporting, we can allow the solutions to have some discontinuities along certain domains, often known as shocks. The site of shock will also be transported. To compute the behavior of shock, we introduce test functions $v \in C_0^\infty$ and apply it on $u_t + f(u)_x = 0$, leading to

$$- \int (uv_t + f(u)v_x) dx dt = 0.$$

This form allows discontinuities of u in either space or time and it is possible to show existence of solution in L^1 . However, uniqueness does not immediate follow since multiple solutions might solve the original NLCL. Extra conditions are often needed (a.k.a. entropy conditions), for example Lax condition (the characteristic must “go into” the shock rather than “leaving” the shock). The other approach is to add artificial viscosity ϵ and to show that the entropy solution in the limit as $\epsilon \rightarrow 0$. Take Burgers' equation (Eqn. 22.2.1) as an example:

$$u_t^\epsilon + \frac{1}{2} ((u^\epsilon)^2)_x = \epsilon u_{xx}^\epsilon.$$

For initial data $u_0(x) = \text{sgn}(x)$, $u(\cdot, t) = u_0$ is a weak solution but not the entropy solution, which shall be

$$u_{\text{entropy}}(x, t) = \begin{cases} -1 & x < -t \\ x/t & |x| \leq t \\ 1 & x > t \end{cases} .$$

The finite difference method can also be adapted to solve NLCL. The upwind scheme reads

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \begin{cases} -\frac{\Delta_- f(U_j^n)}{\Delta x} & \text{if } f'(U_j^n) > 0, \\ -\frac{\Delta_+ f(U_j^n)}{\Delta x} & \text{if } f'(U_j^n) < 0 \end{cases}$$

where we recall $\Delta_- f(U_j^n) = f(U_j^n) - f(U_{j-1}^n)$ and $\Delta_+ f(U_j^n) = f(U_{j+1}^n) - f(U_j^n)$. The Lax-Friedrichs scheme can be used in both cases. To construct a general conservation discretization form, it follows

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} [F(U_{j-s+1}^n, \dots, U_{j+s}^n) - F(U_{j-s}^n, \dots, U_{j+s-1}^n)] =: G(U_{j-s}^n, \dots, U_{j+s}^n). \quad (22.3.1)$$

A convergence result is stated as follows:

Fact 22.3.1 (Lax-Wendroff theorem). *If $\{U_j^n\}$ converges a.e. to a piece-wise smooth $v(x, t)$ under FDM scheme (Eqn. 22.3.1), then $v(x, t)$ is a weak solution.*

The implication is convergence to entropy solution for scalar NLCL problems if FDM scheme is consistent ($F(U, \dots, U) = U$ in Eqn. 22.3.1) and monotone ($\frac{\partial G}{\partial U_j^n} > 0$). A typical estimate on the L^1 error is given by

$$\|U_j^n - u(x_j, t_n)\|_{L^1} \lesssim \Delta^{1/2}$$

for Δt and Δx bounded by Δ . We shall point out that the numerical location of shock may be wrong if the FDM scheme is not in conservation form, although the numerical solution may still look physical.

Apr 10: Lecture 23

Numerical Partial Differential Equations (Part XV: NLCL and FVM)

23.1 Notes on FDM and NLCL

Let us briefly recall some key points for FDM application to PDEs with solutions not in C^1 . The theoretical basis that guarantees uniqueness is either given by defining the weak entropy solution (in nonlinear conservation laws) or weak viscosity solution (in H-J equations). The non-linear conservation law $u_t + f(u)_x = 0$ can be discretized via FDM as

$$\frac{\Delta_+^t U_j^n}{\Delta t} + \frac{\Delta_+^x F(U_{j-s}^n, \dots, U_{j+s-1}^n)}{\Delta x} = 0. \quad (23.1.1)$$

The consistency is defined by $F(U, \dots, U) = f(U)$ via Taylor expansion. We show that this is useful in showing convergence to a weak solution.

Theorem 23.1.1 (Lax-Wendroff). *For the FDM on the conservation form (Eqn. 23.1.1), if the scheme is consistent and $U_j^n \rightarrow u(x_j, t_n)$ boundedly a.e. to a piece-wise C^1 function u , then u is a weak solution to the NLCL.*

Proof. To show that u is a weak solution, let us multiply Eqn. 23.1.1 by $v(x_j, t_n)$, $v \in C_0^1$ and sum over all j, n :

$$\sum_{j,n} \left(\frac{\Delta_+^t U_j^n}{\Delta t} + \frac{\Delta_+^x F(\dots, U_j^n, \dots)}{\Delta x} \right) v(x, t) = 0.$$

We apply summation by parts:

$$-\sum_{j,n} \left[U_j^n \frac{\Delta^t v}{\Delta t} + F(\dots, U_j^n, \dots) \frac{\Delta^x v}{\Delta x} \right] = 0. \quad (23.1.2)$$

Since U converges to u uniformly, we can pass Eqn. 23.1.2 in the limit to conclude that

$$-\sum_{j,n} \left[u_j^n \frac{\Delta^t v}{\Delta t} + F(\dots, u_j^n, \dots) \frac{\Delta^x v}{\Delta x} \right] = 0. \quad (23.1.3)$$

Since u, v are piece-wise C^1 , Eqn. 23.1.3 can be viewed as a Riemann sum that converges in the limit to $-\int u \partial_t v + F(\dots, u, \dots) \partial_x v = 0$, thus showing u is a weak solution. \square

Remark. For scalar problems, the uniform convergence condition can be replaced by the monotone argument, i.e. the coefficients in $U_j^n - \frac{\Delta t}{\Delta x} \Delta_x^+ F(U_{j-s}^n, \dots, U_{j+s+1}^n)$ shall be all positive.

As we mentioned earlier, to address the non-uniqueness of NLCL, an artificial viscosity is added to the equation so that

$$u_t^\epsilon + f(u^\epsilon)_x = \epsilon u_{xx}^\epsilon$$

admits a solution u^ϵ which converges to u . Although this is a well-defined subject, it requires solving a family of solution by taking $\epsilon \rightarrow 0$ or at least using a sufficiently small ϵ to get an approximation, which is often too expensive and prohibitive. To circumvent this, we can adopt the Lax entropy condition that states the characteristics of an entropy solution must go into shocks; this can be generalized to systems that inspect “how many” characteristics go in and out the shock. The other approach is to introduce the so-called “entropy functions” that usually comes from a physical background where the entropy pair (η, q) solve the inequality $\eta(u)_t + q(u)_x \leq 0$.

23.2 Boundary conditions

Let us study how to properly pose BCs for transport problems. If we assume a positive a in

$$\begin{aligned} u_t + au_x &= 0 & x_L < x < x_R, t > t_0 \\ u(x_L, t) &= u_L(t) & t > t_0 \\ u(x, t_0) &= u_0(x) & x_L < x < x_R, \end{aligned}$$

the equation is actually well-defined even without specifying the values on the other side. In fact, the solution is transported along $x - at = C$ and thus the values on the right boundary can be “inferred” by propagating the characteristics. On the other side, if $u(x_R, t)$ are specified, there may not exist a solution unless it matches the value propagated from the characteristics.

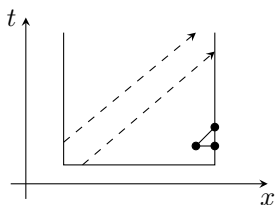


Figure 23.2.1: Characteristics and the upwind scheme.

This matches the behavior of upwind differencing since the upper node depends on the two left-bottom nodes, so no information is needed from the right side. However, we encounter some difficulties when applying Lax-Friedrichs scheme $\frac{U_j^{n+1} - \frac{U_{j+1}^n + U_{j-1}^n}{2}}{\Delta t} + \frac{a}{2\Delta x} (U_{j+1}^n - U_{j-1}^n) = 0$ since the right-most equation requires a out-of-bound node U_{J+1}^n . There are two ways to extrapolate, namely constant extrapolation $U_{J+1}^n = U_J^n$ or linear extrapolation $U_{J+1}^n = 2U_J^n - U_{J-1}^n$.

The stability may be obtained from the direct norm estimate. We can also develop the von Neumann analysis that inspects the normal mode $U_j^n = \lambda^n \chi^j$ where $|\chi| \leq 1$ and λ is to be determined.

23.3 Finite volume methods

The conservation form (Eqn. 23.1.1) is the design principle for FVM. To derive a FVM scheme, let us integrate $u_t + f(u)_x = 0$ over a space-time rectangle

$$\begin{aligned} 0 &= \iint (u_t + f(u)_x) dx dt \\ &= \int_{x_{j-1/2}}^{x_{j+1/2}} [u(x, t_{n+1}) - u(x, t_n)] dx + \int_{t_n}^{t_{n+1}} [f(u(x_{j+1/2}, t)) - f(u(x_{j-1/2}, t))] dt. \end{aligned} \quad (23.3.1)$$

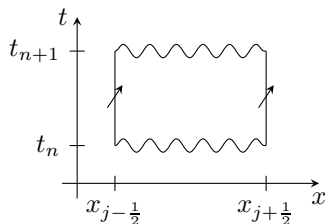


Figure 23.3.1: A FVM scheme illustration.

We can interpret the first term

$$\int_{x_{j-1/2}}^{x_{j+1/2}} [u(x, t_{n+1}) - u(x, t_n)] dx = \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_{n+1}) dx - \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_n) dx$$

as the net change of the material, should we imagine u as the density function. Motivated by this, we define

$$U_j^n := \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_n) dx,$$

leading to

$$U_j^{n+1} = U_j^n - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} [f(u(x_{j+1/2}, t)) - f(u(x_{j-1/2}, t))] dt.$$

To close the numerical scheme, we shall replace $u(x_{j+1/2}, t)$ by the numerical averages U_j^n, U_{j+1}^n . The most general form is to replace $\int_{t_n}^{t_{n+1}} f(u(x_{j+1/2}, t)) dt$ by $\Delta t F(U_j^n, U_{j+1}^n)$, leading to

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} [F(U_j^n, U_{j+1}^n) - F(U_{j-1}^n, U_j^n)]. \quad (23.3.2)$$

It may seem that [23.3.2](#) coincides with the FDM conservation form, but it has a different interpretation based on fluxes and interval averages.

This can be generalized to systems, higher dimensions, or other shapes of volumes. FVM may also be applied to higher order PDEs, for example the viscosity equation $u_t + f(u)_x = \epsilon u_{xx}$ where we can define the flux as $f(u) - \epsilon u_x$ (which still needs to be approximated in the form as $F(U_{j+1}^n, U_j^n)$). Higher order schemes (meaning better accuracy) are, however, difficult to develop and naturally leads to the discontinuous Galerkin method.

Apr 12: Lecture 24

Numerical Partial Differential Equations (Part XVI: FVM and DG)

24.1 FVM (cont'd)

24.1.1 Rankine-Hugoniot condition

The Rankine-Hugoniot condition is proposed to describe the evolution of shock waves. Let us consider the Riemann problem, namely transport equation $u_t + f(u)_x = 0$ with a discontinuous initial condition

$$u(x, 0) = u_0(x) := \begin{cases} u_L & x \leq 0 \\ u_R & x > 0 \end{cases}.$$

We speculate that the discontinuity is propagated at a speed s where the shock wave is, i.e. the solution is in the form

$$u(x, t) = u_0(x - st).$$

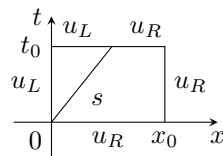


Figure 24.1.1: Set-up to derive the Rankine-Hugoniot condition.

Then, using the relation between spatial averages and temporal fluxes (Eqn. 23.3.1), we have

$$\left[\left(\int_0^{st_0} u_L dx + \int_{st_0}^{x_0} u_R dx \right) - \int_0^{x_0} u_R dx \right] + \left[\int_0^{t_0} f(u_R) dt - \int_0^{t_0} f(u_L) dt \right] = 0$$

and subsequently

$$s = \frac{f(u_R) - f(u_L)}{u_R - u_L}.$$

This can a resemblance to the slope $f'(u)$ for continuous solutions.

24.1.2 Upwind scheme for systems

Let us now investigate the upwind scheme for constant coefficient linear transport equation $u_t + au_x = 0$:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} a \begin{cases} \Delta_+^x U_j^n & a < 0 \\ \Delta_-^x U_j^n & a > 0 \end{cases} \quad (24.1.1)$$

or equivalently

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} [\max(a, 0) \Delta_-^x + \min(a, 0) \Delta_+^x] U_j^n. \quad (24.1.2)$$

Eqn. 24.1.1 (or the equivalent form 24.1.2) can be generalized to some linear transport systems in the form

$$\vec{U}_t + A \vec{U}_x = 0. \quad (24.1.3)$$

We assume that S diagonalizes A , i.e. $SAS^{-1} = \Lambda$ resulting in a diagonal matrix $\text{diag}(\lambda_i)$ of eigenvalues. One can show from PDE theories that 24.1.3 is well-posed under diagonalizable assumption. Let us introduce

$$\begin{aligned} \Lambda_+ &:= \text{diag}(\max(\lambda_i, 0)), A_+ := S^{-1} \Lambda_+ S, \\ \Lambda_- &:= \text{diag}(\min(\lambda_i, 0)), A_- := S^{-1} \Lambda_- S, \end{aligned}$$

then the generalized upwind scheme reads

$$\vec{U}_j^{n+1} = \vec{U}_j^n - \frac{\Delta t}{\Delta x} (A_+ \Delta_-^x + A_- \Delta_+^x) \vec{U}_j^n.$$

24.1.3 Behavior around discontinuities

To study the non-linear setting in general, we replace the transport solution of $u_t + au_x = 0$ with the Riemann solution, namely solutions that has a discontinuity separating two constant regions. We discuss two particular cases that corresponds to shock and rarefaction waves.

- For shock waves, i.e. $f'(u_L) > f'(u_R)$ where two characteristics go into each other. To determine how the solution evolves at a specific site x_0 , let us use the Rankine-Hugoniot condition

$$s = \frac{f(u_R) - f(u_L)}{u_R - u_L}.$$

If $s > 0$, then $u(x_0, t) = u_L$; otherwise, $u(x_0, t) = u_R$ when $s < 0$.

- For rarefaction waves, the general idea is to use the values determined by the entropy solution. For example, let us revisit the Burgers equation $u_t + (\frac{1}{2}u^2)_x = 0$ with initial condition $u_L < 0$ for $x < 0$ and $u_R > 0$ for $x > 0$. The entropy solution

$$u(x, t) = \begin{cases} u_L & x < u_L t \\ x/t & u_L t \leq x < u_R t \\ u_R & u_R t \leq x \end{cases}$$

stays zero along $x = 0, t > 0$. This motivates the following conclusion:

- $u = u_L$ for $f'(u_L) > 0$;
- $u = u_R$ for $f'(u_R) < 0$;
- $u = 0$ otherwise, i.e. when $f'(u_L) < 0 < f'(u_R)$.

The generalization to systems is much more complicated since the behavior of Riemann solutions are more complex.

The FVM scheme, proposed in Eqn. 24.1.1 is of first order due to the fact that Δ_{\pm} is a first order approximated derivative. It is possible to derive higher accuracy schemes in the 1d case by developing interpolation theories, but it is harder to generalize these ideas to higher dimensions due to the complicated geometry of the finite volume cells.

24.2 DG method

Nevertheless, the idea of finite volumes motivates the discontinuous Galerkin method (or DG for short). DG is similar to FEM, but allowing more flexibility to work with non-continuous solutions. Take the NLCL $u_t + f(u)_x = 0$ as an example; we multiply a test function $v(x)$ and integrate both sides

$$\int_{\Omega} [u_t v + f(u)_x v] dx = 0. \quad (24.2.1)$$

As in the FEM setting, we decompose u into basis functions

$$u \approx u_h = \sum_j \alpha_j(t) \varphi_j$$

and pick $v = \varphi_k$. The issue with this formulation is that φ_j may be discontinuous and thus $f(u_h)_x$ may become Dirac functions (which is not in L^2). To circumvent this issue, we divide the whole domain Ω into a few subdomains Ω_m and we require that $\{\varphi_j\}$ is continuous in each Ω_m ; discontinuities are still allowed along the boundaries of Ω_m . Now, Eqn. 24.2.1 reads

$$\sum_m \int_{\Omega_m} \left[\sum_j \alpha'_j \varphi_j \varphi_k + f \left(\sum_j \alpha_j \varphi_j \right)_x \varphi_k \right] dx = 0, \forall k.$$

We proceed with the integration by parts argument, leading to

$$\sum_m \left[\sum_j \alpha'_j \int_{\Omega_m} \varphi_j \varphi_k dx - \int_{\Omega_m} f \left(\sum_j \alpha_j \varphi_j \right) (\varphi_k)_x dx + \int_{\partial\Omega_m} f \left(\sum_j \alpha_j \varphi_j \right) \varphi_k d\sigma(x) \right] = 0.$$

Recall that φ_k may not be discontinuous along the boundaries $\partial\Omega_m$, so the integral is done in the sense under continuation from the interior.

We now discuss the candidates for basis functions. The \tilde{P}_0 family consists of piece-wise constant functions and the corresponding basis functions are indication functions $\mathbf{1}_{\Omega_m}$ on each subdomain; in fact, this recovers the FVM as we mentioned earlier. The \tilde{P}_1 family consists of piece-wise linear functions, so the basis functions are half-hat functions that takes value 0 on one side and 1 on the other side.

DG and FEM are very similar since both methods are built upon weak forms. One primary distinction lies in that DG have more unknowns for the same order of accuracy, thus resulting better approximation for piece-wise functions (however, there is no gain in approximating continuous functions). Besides, DG allows for explicit methods in time since the stiffness matrix is in a block diagonal form (recall that the stiffness is a tri-diagonal matrix for FEM with P_1 element).

Apr 17: Lecture 25

Numerical Partial Differential Equations (Part XVII: DG and Particle Methods)

25.1 DG (cont'd)

We have introduced the modified space of finite element functions \tilde{P}_0, \tilde{P}_1 in the previous lecture. Loosely speaking, \tilde{P}_1 allows simple concatenation of linear segments without requiring the segments to connect (compared to P_1). It is also worth pointing out that the \tilde{P}_0 element leads to the vanilla FVM scheme.

Let us discuss how to properly handle the integral in the Galerkin weak form for DG methods. For NLCL $u_t + au_x = 0$, let us recall

$$\sum_m \sum_j \left[\alpha'_j(t) \int_{x_{m-1/2}}^{x_{m+1/2}} \varphi_j \varphi_k \, dx + a \alpha_j(t) \int_{x_{m-1/2}}^{x_{m+1/2}} (\varphi_j)_x \varphi_k \, dx \right] = 0.$$

Notice that $\int_{x_{m-1/2}}^{x_{m+1/2}} \varphi_j \varphi_k \, dx = 0$ unless the basis index j, k match the domain index m . The jump $[u(x_{m+1/2})]$ is used to derive the numerical flux

$$\begin{cases} a \lim_{x \rightarrow x_{m+1/2}^-} u_m(x) & a > 0 \\ a \lim_{x \rightarrow x_{m+1/2}^+} u_{m+1}(x) & a < 0 \end{cases}.$$

The idea of upwind matching can also be applied to even if there is no temporal variables. For

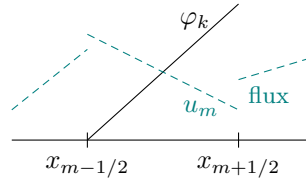


Figure 25.1.1

example, let us put $\eta = 0$ in kinetic equations $\eta u_t + \nabla \cdot (au) = f(x_1, x_2)$, leading to a stationary equation $\nabla \cdot (au) = f$. Here, we can still apply upwind matching for a triangular partitioned mesh, as long as one of the finite differences ∇^\pm along the edges are picked based on the sign of a_1, a_2 .

For elliptic problems, FEM is usually preferable since the regularity of the solution is often guaranteed, so the extra flexibility of DG methods is redundant and is more expensive. Nevertheless, it is possible to derive the DG formulation. Let us study $-u_{xx} = f$ as an example; the DG weak form reads

$$\sum_m - \int_{x_{m-1/2}}^{x_{m+1/2}} u_{xx} v \, dx = \sum_m \int_{x_{m-1/2}}^{x_{m+1/2}} f v.$$

For the left hand side, integration by parts yields a boundary term due to discontinuities

$$\int_{x_{m-1/2}}^{x_{m+1/2}} u_{xx} v \, dx = [u_x v]_{x_{m-1/2}}^{x_{m+1/2}} - \int_{x_{m-1/2}}^{x_{m+1/2}} u_x v_x \, dx.$$

There are many options to define the numerical flux to approximate $[u_x v]_{x_{m-1/2}}^{x_{m+1/2}}$. One option is to perform local averages which, however, leads to weaker stability; the alternative is to use a penalty method (i.e. the Nitsche's method).

25.2 Particle methods

In sharp contrast to the numerical methods we have covered, the particle method represent the solution by a distribution of particles $\{x_j\}$ rather via point-wise values or basis functions. The idea behind is to mimic the physical meaning for kinetic equations that the solution is the density of some physical particles.

The conversion from the value-based representation to the particle-based representation is done by a statistical sampling, say one can invert the samples from a uniform distribution by the cumulative function in a 1d setting. The conversion in the opposite direction can be done via a non-parametric estimation

$$\hat{u}(x) = \frac{1}{nh} \sum_j K\left(\frac{x - x_j}{h}\right)$$

where n stands for the number of particles and K is some proper kernel functions (Gaussian distribution, uniform distribution, etc).

Let us investigate the power of particle methods by studying the transport equation, for example

$$\begin{aligned} u_t + v \cdot \nabla u &= 0, \\ u(x, 0) &= u_0(x) \end{aligned}$$

where we assume v is a constant velocity vector. The analytical solution in close form reads

$$u(x, t) = u_0(x - vt)$$

which implies that the solution is transported along the characteristics $x - vt = C$. Now, since the value represents density, so we can transport the particles instead. So, we can initialize a few particles $\{x_j(0)\}$ by sampling from u_0 and solve the family of ODEs instead $\dot{x}_j(t) = v$. We can easily generalize this to non-constant coefficients, for example $u_t + a(x)u_x = 0$ corresponding to the family of ODEs $\dot{x}(t) = a(x(t))$.

The advantage of particle methods is that there is no numerical diffusion or dispersion involved as in FEM and FDM. The disadvantages, however, are rooted in the nature of its formulation, that only restricted class of equations are supported and it is difficult to get high accuracy.

Inhomogeneous terms can also be addressed in this framework, say $u_t + u_x = bu$ that models particle annihilation/creation. It is a bit tedious to directly simulate the change in the number of particles, but one can circumvent this by properly weighting the particles. Let us introduce $\mu_j(t) := \exp(bt)$ for each particle $x_j(t)$; the weights are chosen such that it solves $\dot{\mu}_j(t) = b\mu_j(t)$. Then, one can verify that the solution can be recovered by

$$\hat{u}(x) = \frac{1}{nh} \sum_j \mu_j(t) K\left(\frac{x - x_j(t)}{h}\right).$$

Many numerical methods can be built on the idea of particle methods. For example, let us study the Vortex method that is used in incompressible Euler equation for fluids

$$\begin{aligned} \vec{u}_t + (\vec{u} \cdot \vec{\nabla}) \vec{u} + \vec{\nabla} p &= 0, \\ \vec{\nabla} \cdot \vec{u} &= 0 \end{aligned}$$

where $\vec{u}(x, y, t)$ stands for velocity and $p(x, y, t)$ for pressure. The particle method does not apply directly, however it is possible after a change in the dependent variable (often known as the vorticity stream-function formulation). We define the vorticity $\omega := \vec{\nabla} \times \vec{u}$ as the curl of the velocity; then

it follows that

$$\omega_t + \vec{u} \cdot \vec{\nabla} \omega = 0, \tag{25.2.1}$$

$$\Delta \Psi = -\omega, \tag{25.2.2}$$

$$\vec{u} = \vec{\nabla} \Psi. \tag{25.2.3}$$

The Vortex method applies particle methods to Eqn. 25.2.1 to estimate ω from the particle distribution and map it back to the grid; then, Eqn. 25.2.2 is solved by a grid-based solver and subsequently u is computed from Eqn. 25.2.3.

Apr 24: Lecture 26

Numerical Partial Differential Equations (Part XVIII: Spectral Methods)

26.1 Complexity analysis

We append a few remarks on discussion of machine learning techniques in PDE problems. One possible application is enabled by the class of neural operators that learns a solution map to the problem of question. These neural operators may sound very different to traditional solvers, nevertheless, there is actually a resemblance from some particular angles. For example, let us consider an initial value problem where the variable data is the values u^0 at the initial time. A one-step explicit scheme (FDM, FEM, FVM...) iterates for $n_t = 1/\Delta t$ times

$$u^n = A_{\Delta t} u^{n-1}, \dots, u^1 = A_{\Delta t} u^0$$

to march along the temporal grid. Under a $\Delta x \sim \Delta t$ scaling (such as transport problems), the total time complexity is $\mathcal{O}(rn_x n_t) = \mathcal{O}(rn_x^2)$ with the bandwidth of A being r . On the other side, we can also directly compute A^n that leads to a $\mathcal{O}(r^2 n_x \log n_x)$ complexity, which can then be applied to any initial data and the evaluation is of $\mathcal{O}(n_x^2)$ expensive. The complexity could be larger if the stability condition requires a $\Delta t \sim \Delta x^2$ scaling; the iterative method requires $\mathcal{O}(rn_x^3)$ while the direct matrix power approach requires the same $\mathcal{O}(r^2 n_x \log n_x)$ complexity.

26.2 Spectral method

The origin of spectral methods dates back to the Fourier's approach to heat equations (1822)

$$\begin{aligned} u_t &= \sigma u_{xx}, \\ u(-\pi, t) &= u(\pi, t) = 0, \\ u(x, 0) &= u_0(x). \end{aligned} \tag{26.2.1}$$

By applying the Fourier transform/series to Eqn. 26.2.1, we arrive at a class of ODEs

$$\begin{aligned} \widehat{u}_t &= \sigma (ik)^2 \widehat{u}, \\ \widehat{u}(0) &= \widehat{u}_0. \end{aligned}$$

Once the ODE problems are solved, the solution of question $u(x, t)$ can be recovered by applying the inverse Fourier transform to $\widehat{u}(k, t)$.

This method works well for problems with a constant coefficient; however, it is hard to generalize to variable coefficient problems that has a convolution on the Fourier side, making the numerical application costly. Instead, we apply the Fourier transform (or FFT in practice) to evaluate the derivatives only. For example, consider the transport problem with periodic BC

$$\begin{aligned} u_t + a(x) u_x &= 0, \\ \text{Periodic BC} \\ u(x, 0) &= u_0(x). \end{aligned}$$

Recall that $\widehat{u_x} = ik\widehat{u}$; this motivates us to propose the FFT-based differential scheme

$$[a(x) u_x]_j^n = a(x_j) \{ \mathcal{F}^{-1} [ik \mathcal{F} (u^n)] \}_j \tag{26.2.2}$$

where \mathcal{F} stands for the discrete Fourier transform.

A few remarks/caveats on the Fourier method:

- Compared to the FDM/FEM discretization that has a fixed $\mathcal{O}(\Delta x^p)$ accuracy, the Fourier-based difference performs better than an algebraic rate if the solution is smooth.
- A minor downside is that time complexity of the Fourier-based scheme is $\mathcal{O}(\frac{1}{\Delta x} \log \frac{1}{\Delta x})$ while the FDM/FEM scheme is usually of $\mathcal{O}(\frac{1}{\Delta x})$.
- Another major issue is that the Fourier transform can only work on periodic domains (which, nevertheless, can be circumvented by adopting a Chebyshev basis).

- Besides, the solution must be smooth for the Fourier-method to be competitive.
- The Fourier-side-difference can be modified to handle stability, for example we apply a decay in the frequency domain

$$\mathcal{F}^{-1} [\theta(k) ik \mathcal{F}(u)]$$

where θ has a proper decay as $|k| \rightarrow \infty$ or a simple cutoff at a particular wave-number. This usually ensures the stability but could harm the accuracy.

The spectral method can be applied in the following settings. In atmosphere simulation, the Fourier method is coupled with a FDM/FEM/FVM method where the former handles the tangent evolution and the latter deals with the dynamics in the normal direction; this idea is also widely used in generic analysis of turbulence. The lattice structure in quantum mechanics is also suitable to apply spectral methods to study wave functions or scattering phenomena.

A distinct feature of the spectral method is that it is a global method while the previous methods we have covered (FDM/...) is a local method. The difference scheme (Eqn. 26.2.2) can be understood as a finite difference scheme supported on an infinitely-wide stencil; the complexity is drastically reduced thanks to the fast Fourier transform.

26.3 Conclusion remarks

We also point out that these numerical schemes can be combined in a few flexible manners. Apart from the FDM-FEM coupling for time-space problems, it is also possible to use spectral methods as a local basis on each little finite element that leads to a FEM-SM coupling. FEM can also be coupled with PM where a hat-shaped function is transported with the particle that forms a basis for the weak form. FVM can be coupled with PM that is often adopted in the vorticity stream-function formulation (Eqn. 25.2.1).

As an overview, FDM is often applied to 1d problems; for engineering purposes, FEM/DG and FVM (for fluid dynamics) are often preferred; FEM/FDM are also applied in problems with a physics/chemistry background; PM and SM are adopted in some special settings.

Lecture A

Neural Operators and PDEs

In this lecture, we briefly discuss the neural operators and the applications in PDEs. The two major applications can be understood in the following framework: let us consider the following parameterized equation

$$\mathcal{L}(u; a) = f, \tag{A.0.1}$$

$$\mathcal{B}(u; a) = g; \tag{A.0.2}$$

possible examples are the elliptic equation $-\nabla \cdot (a\nabla u) = 0$ that models permeability/conductivity, the parabolic equation $\partial_t u - \nabla \cdot (a\nabla u) = 0$ for heat equations, or the transport equation $\partial_t u + a(x)\partial_x u = 0$. One application is to simply solve Eqn. A.0.1 with the power of neural networks, without any prior knowledge on the solution u ; we call this the “neural solver”. The other application does not aim to solve Eqn. A.0.1 accurately on its own, but rather try to obtain an approximated solution with a relatively lighter effort, often known as a “neural operator”. As the name indicates, we build a mapping/operator that learns the relation between functions, i.e. mapping the coefficient a and physical location x to the solution u . There are other applications as well and the aforementioned methods can be combined under certain scenarios.

A.1 Neural network 101

A.1.1 Artificial neural networks

Inspired by the biological neural networks of animal brains, the concept of artificial neural networks (or ANN for short) refers to a collection of interacting nodes structured by a given set of mathematical rules. Information is passed in from some input modules and processed between the nodes

before it is taken out from the network. A particular example is the so-called feed-forward neural network (FNN) where the flow of information forms a directed acyclic graph, namely in one single direction without going back to the previous nodes. FNNs are easy to model by compositions of basic functions and can be trained efficiently by the back propagation technique to be mentioned later.

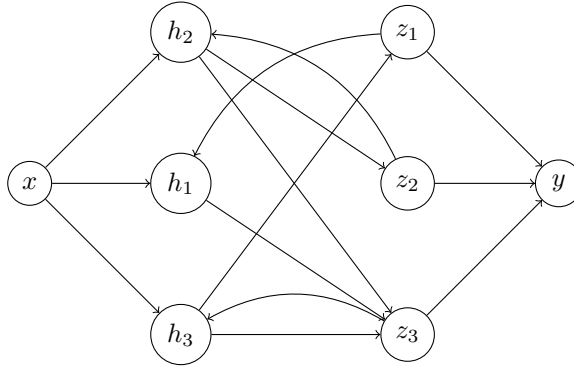


Figure A.1.1: Illustration on general ANN structures.

Let us begin with a class of networks that have simple structures, yet rich theoretical properties, known as the multilayer perceptrons (MLP). A perceptron is a multivariate function, defined as the composition of an affine map $w^T x + b$ and a non-linear activation σ :

$$p(x; w, b) := \sigma(w^T x + b), \quad x, w \in \mathbb{R}^N, b \in \mathbb{R}.$$

The design of the perceptron can be motivated by combination of incoming information. These perceptrons are used to build layers, namely

$$L\left(x; \left\{w^{(k)}, b^{(k)}\right\}_{k=1}^K\right) := \begin{pmatrix} p(x; w^{(1)}, b^{(1)}) \\ \dots \\ p(x; w^{(K)}, b^{(K)}) \end{pmatrix}.$$

The layers are supposed to handle the incoming information x from different perspectives, if well-trained on the data observed. To build a MLP with given number of input dimensions d_{in} and output dimension d_{out} , we propose hidden layers of width m and depth n as follows:

$$M(x; \theta) := \tilde{L}_{n+1} \circ L_n \circ \dots \circ L_2 \circ L_1$$

where θ is the collection of weights and each map L_i has its own weights $w^{(i, k_i)}$ and biases $b^{(i, k_i)}$. Notice that we put a tilde on the final output layer, meaning that no activation is applied on that

layer.

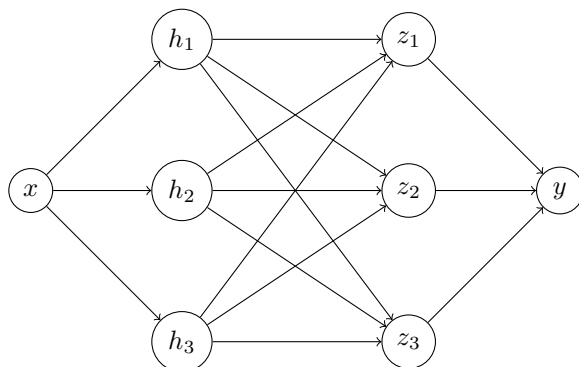


Figure A.1.2: Illustration on general MLP structures.

MLPs are not born to be perfect, however. The weights and biases need to be adjusted accordingly based on the training data. A training set is defined as the collection of input-output pairs (or feature-label pairs in other context), namely

$$T := \{(x_i, y_i) : x_i \in R^{d_{in}}, y_i \in R^{d_{out}}, 1 \leq i \leq N_t\}.$$

A (strictly) optimal MLP perfectly maps each x_i to the corresponding y_i . Nevertheless, it is nearly impossible to find the optimal MLP in the strict sense since there might be noisy or even polluted adversarially. One of the best practices is to introduce the so-called loss function J that measures how “bad” the prediction $\hat{y}_i := M(x_i)$ is compared to the given output y_i . A common choice is the mean squared error

$$\text{MSE}(\{\hat{y}_i\}, \{y_i\}) = \frac{1}{N_t} \sum_{i=1}^{N_t} |\hat{y}_i - y_i|^2.$$

Once the loss function is given, we aim to minimize the loss for all possible choices of the parameters $\theta := \{w^{(i,k_i)}, b^{(i,k_i)}\}$. The simplest approach is to perform gradient descent

$$\theta \leftarrow \theta - \alpha \partial_{\theta} J.$$

There is a technique to boost the efficiency in computing $\partial_{\theta} J$ called backpropagation; we refer to [Backpropagation - Wikipedia](#) for interested readers.

A.1.2 Universal approximation theory

One might ask if ANNs are capable of solving PDE problems; after all, the neural operator/solver is infeasible if ANNs can’t approximate the solutions well. In fact, as we see below, well-designed

ANNs can approximate any continuous functions to any given precision, so this is really a powerful tool.

To simplify the notation, let us consider MLPs with single hidden layer for single input and output, namely

$$M(x; \{w, b\}, K) := \sum_{k=1}^K w^{(2,k)} \sigma \left(w^{(1,k)} x + b^{(1,k)} \right) + b^{(2)}, \quad x \in \mathbb{R}. \quad (\text{A.1.1})$$

Theorem A.1.1 ([Siegel & Xu, 2020]). *Assuming σ is Riemann integrable with polynomial growth $|\sigma(x)| \lesssim (1 + |x|)^p$ and σ is not a polynomial. For any given $f \in C(\Omega)$ on a compact set Ω and $\epsilon > 0$, there exists $\{w, b\}$ and K s.t.*

$$\max_{x \in \Omega} |M(x; \{w, b\}, K) - f(x)| < \epsilon.$$

This is also known as the universal approximation theorem. The proof is a bit technical, so we'd like to motivate this idea from a simplified setting, i.e. σ being the rectified linear unit (ReLU) function

$$\text{ReLU}(x) := \max(0, x).$$

We point out that ReLUs can be used to build the hat functions we have seen in P_1 element by

$$\varphi(x) := \sigma(x+1) - 2\sigma(x) + \sigma(x-1).$$

Then, due to the knowledge that P_1 is dense in $C([a, b])$, we conclude that

$$M_1(x) := \sum_{j=1}^N f(x_j) \varphi\left(\frac{x - x_j}{h}\right), \quad x_j := a + (j - 1/2)h, \quad N := \frac{b - a}{h}$$

approximate $f \in C([a, b])$ to precision $h \max_x |f'(x)|$. We point out that the universal approximation theorem can be extended to the regime where the MLP takes multiple inputs and yields multiple outputs.

A.1.3 Variants and extensions

There are a few variants to the MLP structure, for example the dropout technique (randomly disconnecting connection between neurons), batch normalization (center and normalize the latent information), the resnet structure (bypass skip connection), etc. There are also an active interest to study networks with particular structures to better capture the desired information, for example convolutional neural network for image processing and recognition, recurrent neural network for

data with a time-series nature, transformers for sequential to sequential problems, etc.

A.2 Neural operators

A.2.1 Motivation

In a broader sense, we can compare a traditional PDE solver with a specialized neural network since they both take in the coefficient data and output solutions to the problem. Another similarity is that the feed-forward propagation resembles the explicit time marching behavior. A major difference lies in the fact that PDE solvers are often accurate and born with good error control while feedforward neural networks do not possess guarantee for error estimation. Based on this observation, we can build a neural network that learns the mapping that is implicitly determined by the PDE solver as a surrogate for downstream applications, for example inverse problems or optimal experimental design.

A.2.2 A simplified setting

We follow the idea proposed in [Kovachki, Nikola, et al, 2021] but we work on a simplified model. Let us assume the PDE problem of interest reads

$$\mathcal{L}(u; a) = f \tag{A.2.1}$$

where \mathcal{L} , parametrized by a , is a linear/non-linear differential operator of u and f is a source term. Let us assume that Eqn. A.2.1 implicitly determines a functional

$$\mathcal{S} : (a, f) \mapsto u$$

which can be well approximated by an established PDE solver

$$\mathcal{S}_0 : D_{in} = \left(\{a(x_i)\}_{i=1}^{N_x}, \{f(x_i)\}_{i=1}^{N_x} \right) \mapsto D_{out} = \{u(x_i)\}_{i=1}^{N_x}.$$

We confine $D_{in} \in \mathbb{R}^{2N_x}$ to be in a bounded set $\Omega \subset \mathbb{R}^{2N_x}$. We adopt the MLP with one-hidden layer as proposed in Eqn. A.1.1. By the universal approximation theorem, for any given precision requirement $\epsilon > 0$, there exists a MLP realization M s.t.

$$\sup_{D_{in} \in \Omega} \|S_0(D_{in}) - M(D_{in})\|_{\infty} < \epsilon.$$

To find an optimal network, we prepare the training set by randomly sampling $\{a_i\}$ and $\{f_i\}$ from the set Ω and the PDE solver to compute the solution $D_{out} = \{u_i\}$. The MLP can then be trained

by a gradient method with the back-propagation technique.

A.2.3 Discussion

A few modifications can be made the framework mentioned above. For example, in DeepONet [Lu, Jin, Karniadakis, 2019], the coefficient vector and solution vector are encoded separately so that it allows prediction on physical locations other than the training set. Other improvements are mentioned in [Kovachki, Nikola, et al, 2021], including low rank approximations, Fourier transform that enables frequency learning, a hierarchical decomposition powered by multipole graph structure, etc.

A.3 Neural solver without prior knowledge

A.3.1 PINN

The neural network can also be used to directly solve the PDE problem without the aid from an established high-accuracy solver. For example, the physics informed neural networks (PINNs [Raissi, Perdikaris, Karniadakis, 2017]) propose to solve Eqn. A.2.1 by

$$\min_{\theta} \|\mathcal{L}(u_{nn}(x; \theta); a) - f\|_{\Omega}^2 + \|\mathcal{B}(u_{nn}(x; \theta); a) - g\|_{\partial\Omega}^2.$$

The second term here demonstrates how to match boundary conditions. We shall point out this formulation shares a lot of similarities with the collocation method in the FEM literature. We can gain flexibility in geometry and form of the differential operator in collocation-based methods, but there is no theory on error bound in general.

A.3.2 Deep Ritz

Loosely speaking, the Deep Ritz method resembles the Galerkin method if we identify PINNs as the collocation method. The Deep Ritz method applies to problems that can be written as a weak minimization form, for example

$$-\nabla \cdot (a \nabla u) = f.$$

In [Yu, 2018], the authors propose to minimize the Ritz energy functional

$$\min_{\theta} \frac{1}{2} \int_{\Omega} a \nabla u \cdot \nabla u - \int_{\Omega} f u.$$

The advantage of these neural solvers is that it is relatively easier to set-up a computation procedure based on the well-established autograd library infrastructure. However, we shall point

out that although PINNs and Deep Ritz method has promising numerical performances, there is no general framework for an a priori error estimate. Besides, training the network can also be an issue as we deal with a non-convex optimization problem even for linear PDEs. Furthermore, these problems may fail for problems with special structures, say shock speed in NLCL, so one might need to take these factors into account when designing the formulation of neural solvers.

Bibliography

- [Siegel & Xu, 2020] Siegel, Jonathan W., and Jinchao Xu. "Approximation rates for neural networks with general activation functions." *Neural Networks* 128 (2020): 313-321.
- [Kovachki, Nikola, et al, 2021] Kovachki, Nikola, et al. "Neural operator: Learning maps between function spaces." *arXiv preprint arXiv:2108.08481* (2021).
- [Lu, Jin, Karniadakis, 2019] Em Karniadakis. "Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators." *arXiv preprint arXiv:1910.03193* (2019).
- [Raissi, Perdikaris, Karniadakis, 2017] Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations." *arXiv preprint arXiv:1711.10561* (2017).
- [Yu, 2018] Yu, Bing. "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems." *Communications in Mathematics and Statistics* 6.1 (2018): 1-12.