MTH 453-553 Spring 2018
Class notes

Malgorzata Peszyńska

Department of Mathematics, Oregon State University

Spring 2018

Introduction and Overview

These notes are intended to supplement rather than replace any textbook material or material covered in lectures.
Examples will be worked out in class.

I will use material from several references

- [RJL]. R.J. LeVeque, Finite Difference Methods for ODEs and PDEs, SIAM 2007
- [IK] Isaacson, Keller, Analysis of Numerical Methods
- [QV] Quarteroni, Valli, Numerical Approximation of Partial Differential Equations, Springer, Second Ed., 1997

The purpose ... (by Richard Hamming)

- "The purpose of computing is insight, not numbers"
- "The purpose of analysis is to solve problems, not create pretty theorems"

PDE types

Consider $u = u(t, x, y, \ldots)$, or $u : \Omega \in \mathbb{R}^n \to \mathbb{R}$ which solves

$$F(u, u_t, u_x, u_y, \ldots, u_{tx}, u_{ty}, u_{tt}, u_{xx}, u_{yy}, u_{xy}, \ldots) = 0 \quad (1)$$

1. The PDE (1) must be supplemented with appropriate boundary and/or initial conditions on $\partial\Omega$. We will consider BVP, IVP, and IBVP.
2. Does the solution to (1) (with the additional conditions) exist? Is it unique? How does it depend on its data? Is the problem well-posed?
3. What is the qualitative nature of solutions and their regularity?
4. In principle, higher order PDEs can be converted to systems of lower order, but this does not help much in analysis/solving except in special circumstances.
5. Numerical methods have to honor the behavior.

Suggested review/reading: Guenter/Lee text; other MTH 621- or MTH 4/582.*

Examples of PDEs of canonical types and not

Recognize . . .

Order, linear/nonlinear, what conditions (BVP, IVP, IBVP) needed. Application/name ?

$$u_t + au_x = 0; \;\; u_t + uu_x = 0; \;\; u_t + (\frac{u^2}{2})_x = 0; \;\; u_t + (u_x)^2 = 0 \tag{2}$$

$$-u_{xx} - u_{yy} = 0; \;\; -\Delta u = f; \tag{3}$$

$$u_t - u_{xx} = 0; \quad u_t - \nabla \cdot (k(x,y)\nabla u(x,u)) = f; \tag{4}$$

$$u_t - \nabla \cdot (k(u)\nabla u) = 0; \tag{5}$$

$$u_t + au_x - ku_{xx} = f(u), \; u_t + \nabla \cdot g(u) - \nabla \cdot (k(x,y,u)\nabla u) = f(u) \tag{6}$$

$$u_{tt} - u_{xx} = 0; \;\; u_{tt} + u_t = \Delta u \tag{7}$$

$$-\mu\Delta u = -\nabla p, \; \nabla \cdot u = 0, \;\; \text{or} \;\; u_t - u \cdot \nabla u - \mu \Delta u = -\nabla p, \; \nabla \cdot u = 0 \tag{8}$$

NOT (just) PDEs:

$$u_t + \int_0^t e^{-(t-s)} u_t(s) ds - u_{xx} = 0 \tag{9}$$

$$u_t + \int_\Omega k(x)u(x)dx - u_{xx} = 0 \tag{10}$$

What we have covered in other classes

MTH 452/552, Basic methods for BVP and IVP

- BVP solvers for $-u'' = f$:
  - Approximate as $-(D_h^2 U)_j = f_j$; apply BC, solve linear system
  - Error analysis for BVP solvers:
    Consistency & Stability ≡ Convergence
    LTE of $O(h^p)$ & Bounds for $\| A^{-1} \|_* \equiv \| E_h \|_\dagger = O(h^p)$.
  - Norm choice † depends on $*$ and p
- IVP solvers for $u' = f(t, u)$: single-step and LMM methods

What we have not covered

- Non-Dirichlet BC; higher order methods, non-uniform grids
- How to extend to BVP in $\Omega \subset \mathbb{R}^d, d > 1$; $-\nabla \cdot (K \nabla u) = f$
- How to treat IBVP, e.g., for $u_t - u_{xx} = f$
- How to solve large sparse linear systems (see MTH 451-551)

Outline of class

- Two-point BVP, general BC, variable coefficients

  Recall 4/552 material; expand the theory, implementation, and applications

- Laplace equation $-\Delta u = 0$ and more generally $-\nabla \cdot (K \nabla u) = f$
- Heat (diffusion) equation $u_t - \Delta u = f$
- Wave equation $u_{tt} - \Delta u = 0$
- Advection/transport equation $u_t + a u_x = 0$
- Other: in particular, ADR $u_t + a u_x - d u_{xx} = f(u)$.

Methods, math/ implementation/ applications content

We will focus on FD (finite differences). However, other methods such as FE (finite elements), and spectral methods will be mentioned.

`Code templates will be provided.`

We will study, as always, consistency, stability, convergence.

Interplay of discretization in t and x will be important.

Recall the two-point BVP (MTH 452/552)

Conditions in linear two-point constant coefficient BVP

$$u'' + mu' + nu = f(x), \quad x \in [a,b]; u(a) = u_a, u(b) = u_b \tag{11}$$

Condition $u(a) = u_a$ is known as Dirichlet condition, $u'(a) = u_b$ is known as Neumann condition, $u'(a) + Cu(a) = u_c$ is known as the Robin condition. One can also impose periodic conditions, e.g., $u(a) = u(b), u'(a) = u'(b)$. If the data equals 0, we call the BC homogeneous.

Example 2.1 (Existence and uniqueness of solutions depending on BC)

Consider $x \in [0,1]$. Find the general solution and determine if the solution exists and if it is unique.

$u'' = 0, u(0) = 0, u(1) = 0$
$u'' = 1, u(0) = 0, u(1) = 0$
$u'' = 1, u'(0) = 0, u'(1) = 0$
$u'' = 1, u(0) = 0, u'(1) = 0$
$u'' = 1, u'(0) + u(0) = 0, u'(1) = 0$
$u'' = 1, u(0) = u(1), u'(0) = u'(1)$
$u'' = u, u(0) = 0, u(1) = 0$
$u'' = u, u(0) = 0, u'(1) = 0$
$u'' = -u, u(0) = 0, u(1) = 0$

Numerical solution of two-point BVP (11)

1. Define a grid on $[a, b]$. Easiest is uniform grid $a = x_0, x_1, \dots x_M = b$. Here $h = \frac{b-a}{M}$, and $x_j = a + jh$.
2. Discretize. (Replace $u''(x)$ by $D_h^2 u(x) = \frac{u(x+h)-2u(x)+u(x-h)}{h^2}$, and other derivatives by difference quotients.)
3. Write the discrete equation to be satisfied by U^{j-1}, U^j, U^{j+1} at every interior node t_j
4. Apply BC to U^0 and U^M
5. Collect all unknowns in $U = [U^0, U^1, \dots U^{M-1}, U^M]^T$, and right-hand side in $F = [f_0, f_1, f_2, \dots f_{M-1}, f_M]^T$. The entries f_0, f_M will be given from boundary conditions.
6. Identify the coefficients in steps 2,3 as components of a matrix A
7. Solve the linear system $AU = F$

Remark 1 (MATLAB numbering)

The nodes and unknowns above are numbered from $0, 1, \dots M$. MATLAB does not allow indexing from 0, so the unknowns in your code will likely be numbered $U^1, \dots U^{M+1}$. The principles remain the same.

Example: $-u'' = f$ on $[a, b]$ with Dirichlet BC $u(a) = u_a, u(b) = u_b$

Example 2.2 (Derive and solve the discrete system of equations)

Decide on M and h, $h = \frac{b-a}{M}$.
Equation for internal nodes and boundary conditions

$$\text{Interior nodes}: \quad \frac{1}{h^2}(2U^j - U^{j-1} - U^{j+1}) = f(t_j), \ \ \forall j = 1, \dots M-1 \tag{12a}$$

$$\text{Boundary nodes}: \quad U^0 = u_a, \ U^M = u_b. \tag{12b}$$

Set-up an equivalent system of equations $AU = F$ with

$$A = \frac{1}{h^2}\begin{bmatrix} h^2 & 0 & \dots & 0 & \\ -1 & 2 & -1\dots & & \\ \dots & & & & \\ \dots & & -1 & 2 & -1 \\ \dots & & & & h^2 \end{bmatrix}, \quad F = \begin{bmatrix} u_a \\ f(t_1) \\ \dots \\ f(t_{M-1}) \\ u_b \end{bmatrix}. \tag{13}$$

Solve linear system $AU = F$. In MATLAB, write `U=A\F`

Note the first and last rows in (13) correspond to the boundary conditions, and could be trivially eliminated. The numbering above is not MATLAB-like.

Code for BVP from slide 9

Numbering, again (recall Remark 1)

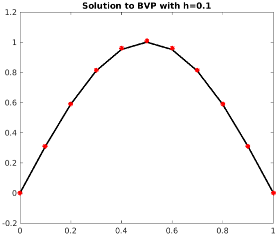
When coding, the interior nodes will be numbered $2, \dots M$, and boundary nodes will be numbered 1 (left boundary) and $M+1$ (right boundary). In particular, when we apply the Dirichlet BC, we write $U(1) = u_a; U(M+1) = u_b$, for the first and last entries of U.

```
dx = (b-a)/(M); x = (a:dx:b)'; %% use uniform grid, x is numbered from 1...M+1
%% set up the matrix and the right hand side vector
f = rhsfun(x);
A = sparse(M+1,M+1);
for j=2:M
    %% set up j'th internal row of the matrix
    A(j,j) = 2; A(j,j-1) = -1; A(j,j+1) = -1;
end
A = A / (dx*dx);
%% apply the Dirichlet b.c. to the matrix and right hand side
A (1,1) = 1; f (1) = ua;
A (M+1,M+1) = 1; f (M+1) = ub;
%% solve the linear system;
U = A \ f;
```

This code can be vectorized for faster performance

Solution to BVP from slide 9

Start assuming an exact solution $u(t) = \sin(\pi t)$.
Calculate (manufacture) $f(t) = \pi^2 \sin(\pi t)$.
Implement the code and plot ...



h	$E(h,\infty)$	$E(h,2)$
0.1	0.008265	0.058
0.05	0.00205871	0.0015
0.01	0.0000822508	0.0000582

Test convergence in L^∞ and L^2 to see that $E(h,*) = O(h^2)$ for each norm; see table.
(See below for a precise definition of norms, and convergence analysis).

Analysis of convergence of FD for BVP

Define $A_h \in \mathbb{R}^{(M-1)\times(M-1)}$ *to be the interior part of the matrix* A *from* (13).

Remark 2 (Error analysis)

Writing the error equation we find $A_h E = \tau$ *where* τ *is the vector of truncation errors* τ_j *(truncation errors on the boundary are 0). Proving convergence*

$$\| E \|_* \to 0, \quad \text{as } h \to 0 \tag{14}$$

requires proving consistency and stability.

Here $*$ *stands for the functional space in which we want to measure the error.*

- Consistency: with D_h^2, the truncation error $\tau_j \approx O(h^2)$ if $u \in C^4$
- Stability: to keep $\| E \|_* = O(h^2)$, we require $\| A_h^{-1} \|_* \le C$.

Choosing norm

We generally like to use $*$ to be L^∞ space, but it is easier to get the bound $\| A_h^{-1} \|_{L^2}$.

Vector, matrix, function & grid function norms.

Remark 3 (Vector norms in l_p spaces)

$$\mathbb{R}^d \ni x, \qquad \|x\|_p := \left(\sum_j |x_j|^p\right)^{\frac{1}{p}}, \; 1 \leq p < \infty. \; \|x\|_\infty = \max_j |x_j|\,. \tag{15}$$

Remark 4 (Norms for functions and grid functions)

For functions $f : [a, b] \to \mathbb{R}$ we extend the l_p vector norms to the norms in Lebesgue spaces L^p of functions for which the integral below is well defined and finite

$$\|f\|_p = \left(\int_a^b |f(x)|^p \, dx\right)^{\frac{1}{p}}; \quad 1 \leq p < \infty. \quad \|f\|_\infty = esssup_{x\in[a,b]} |f(x)|\,. \tag{16}$$

For grid functions $f : \Delta \to \mathbb{R}$ (sampled on a grid $\Delta = \{x_1, x_2, \ldots x_M\}$) with $1 \leq p < \infty$, we approximate these integrals with Riemann sums

$$\|f\|_{\Delta,p} = \left(\sum_j h\,|f(x_j))|^p\right)^{\frac{1}{p}} = h^{\frac{1}{p}} \, \|[f(x_1), \ldots f(x_j) \ldots f(x_M)]^T\|_p \tag{17}$$

The text uses the same symbol $\|\cdot\|_p$ for the norms of vectors in $\mathbb{R}^d$, functions on $\mathbb{R}$, or grid functions. We will use $\|f\|_{\Delta,p}$ for grid functions.

Stability and convergence in the 2-norm

Remark 5 (Recall some matrix norms)

$$\mathbb{R}^{d\times d} \ni A, \qquad \|A\|_1 = \text{“max abs col sum”} = \max_j \sum_i |a_{ij}| \tag{18}$$
$$\mathbb{R}^{d\times d} \ni A, \qquad \|A\|_\infty = \text{“max abs row sum”} = \max_i \sum_j |a_{ij}| \tag{19}$$

The norm $\|A\|_2 = \sqrt{\rho(A^T A)}$, *where the spectral radius* $\rho(B)$ *is the largest to magnitude eigenvalue of matrix* B.
For a symmetric real matrix $\|A\|_2 = \rho(A)$.

In MTH 452-552 we proved stability of FD solution to the two-point Dirichlet BVP...

We have proved that $\|A_h^{-1}\|_p \leq C$ for $p = 2, \infty$.
From this it follows for $E = A_h^{-1}\tau$ that

$$\|E\|_{\Delta,2} = \sqrt{h}\,\|E\|_2 \leq \sqrt{h}\,\|A_h^{-1}\|_2\,\|\tau\|_2 = \sqrt{h}CO(h^2)\sqrt{M} = O(h^2) \tag{20}$$
$$\|E\|_{\Delta,\infty} = \|E\|_\infty \leq \|A_h^{-1}\|_\infty\,\|\tau\|_\infty \leq CO(h^2) = O(h^2) \tag{21}$$

Two-point BVP; Neumann boundary conditions

Consider $-u'' = f, \ \ x \in (a,b); u(a) = u_a, u'(b) = g_b$.

To discretize, we modify (12b) in (12)

$$\text{Interior}: \quad \frac{1}{h^2}(2U^j - U^{j-1} - U^{j+1}) = f(t_j), \ \ \forall j = 1, \ldots M-1 \tag{22a}$$

$$\text{Boundary}: \quad U^0 = u_a, \ (D_h^* U)_M = g_b. \tag{22b}$$

We have to choose the difference operator $(D_h^* U)_M$.

- One-sided $(D_h^* U)_M = \frac{U^M - U^{M-1}}{h} = g_b$, which is $O(h)$ accurate
- Two-sided $(D_h^* U)_M = \frac{U^{M+1} - U^{M-1}}{2h} = g_b$, which is $O(h^2)$, but introduces an additional unknown U^{M+1} and requires an additional equation (e.g., we can write (22a) for $j = M$).
- One-sided second order approximation (recall from MTH 4/552).

After the choice is made, write the system (22) as $AU = F$.

When choosing D_h^*, note: if one can ensure that the system $AU = F$ has a symmetric matrix A, (i) proofs of stability in L^2 are much easier, and (ii) iterative linear solvers for sparse systems "like" spd A.

Variable coefficients

Consider $-(ku')' = f, x \in (a,b), u(a) = u_a, -(ku')(b) = q_b$.

Do it right

1. It is a BAD idea to apply the product rule to this problem.
2. Instead, we first discretize $(D_h U)_{j\pm1/2}$ at each point $x_{j\pm1/2}$, calculate $k(x_{j\pm1/2})(D_h U)_{j\pm1/2}$, and apply D_h again.

The analogue of (22a) is now

$$-\frac{1}{h}\left(k(x_{j+1/2})\frac{U^{j+1}-U^j}{h} - k(x_{j-1/2})\frac{U^j-U^{j-1}}{h}\right) = f(x_j) \qquad (23)$$

Next we discretize the flux condition $-(ku')(b) = q_b$ with, say, one-sided approximation.
Finally we set up the system $AU = F$, and solve.

Ex.: Attempt LTE analysis assuming $k(x)$ is smooth. Also, write the linear system to be solved

Solving BVP in $\mathbb{R}^d$, $d > 1$

We first consider $d = 2$ and the Laplace operator
$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. (in $d = 3$ we add $\frac{\partial^2}{\partial z^2}$).

Consider an open bounded domain $\Omega \subset \mathbb{R}^d$, with the boundary $\partial\Omega$ which is (at least) piecewise smooth. Our model problem is the Poisson's equation with homogeneous Dirichlet BC.

$$-\Delta u = f, \; x \in \Omega, \tag{24a}$$

$$u|_{\partial\Omega} = 0. \tag{24b}$$

The E/U theory for (24) is more complicated than for (11).

From now on we will always assume that the unique solution exists and is smooth enough for the approximation to make sense.

Remark 6

More generally than (24) one can consider other elliptic PDEs *with variable coefficients and lower order terms.*

FD for (24)

It is fairly natural to generalize the discretization of $-\frac{d^2}{dx^2}$ in (11) to the discretization of $-\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2}$ in (24).

- We need a grid in x and y directions (with step sizes h_x and h_y, respectively). Points are numbered with i, j. We approximate $u(x_i, y_j) \approx U^{i,j}$.
- We can easily write the analogues of (22a) for (24a)

$$\frac{1}{h_x^2}(2U^{i,j} - U^{i-1,j} - U^{i+1,j}) + \frac{1}{h_y^2}(2U^{i,j} - U^{i,j-1} - U^{i,j+1}) = f(x_i, y_j), \quad (x_i, y_j) \in \Omega \tag{25a}$$

- We can set-up boundary conditions for (24b)

$$U^{i,j} = 0, \ (x_i, y_j) \in \partial\Omega \tag{25b}$$

Challenge: How to implement (25) as $AU = F$, and solve it.

"PDEs are easy, boundary conditions are difficult" (RES)

When Ω = unit square = $(0, 1) \times (0, 1)$

The solution to "How to implement (25) as $AU = F$" is particularly simple. Below we show one (of many) way(s).

Define grid: uniform with $h_x = \frac{1}{M_x}, h_y = \frac{1}{M_y}$.

The grid points $x_i = (i-1)h_x$, $i = 1, \dots M_x + 1$, $y_j = (j-1)h_y$, $j = 1, \dots M_y + 1$.
The interior points (x_i, y_j) have $2 \le i \le M_x$, and $2 \le j \le M_y$.
The boundary points have numbers as follows ...
start with $i = 1$ ($x_i = 0$, West=left)
or $i = M_x + 1$ ($x_i = 1$, East=right),
or $j = 1$ ($y_j = 0$, South=bottom),
or $j = M_y + 1$ ($y_j = 1$, North=top).

Lexicographic numbering of unknowns, and coding

We need to order (i, j), assigning a "global" index to each. It should be easy to go both ways $(i, j) \leftrightarrow index$, and to recognize which nodes are on the boundary.
For example, $index(i, j) = (j-1)(M_x + 1) + i$ assigns order row-wise.

This brings us to the second difficulty when coding: element $(1, 1)$ *is in upper left corner of a matrix, while we usually associate* $(1, 1)$ *on a grid with the lower left corner. Also, i runs in rows, and j runs in columns. Keep this in mind when interpreting! MATLAB also has functions* `meshgrid,flipud` *which can help.*

Code for numbering and indexing

```
   ax=0;bx=1;ay=0;by=1;Mx=2;My=3;hx=(bx-ax)/Mx;hy=(by-ay)/My;
   x=hx*(0:1:Mx);y=hy*(0:1:My);
   [xx,yy]=meshgrid(x,y); %% ''grid'' has Mx columns and My rows
   %%
   zz = zeros(size(xx'));
   for i=1:Mx+1,for j=1:My+1, myin(i,j) = myind(i,j,Mx,My); end end
   %% show the numbering provided by index
   myin,
   %% show the numbering as intuition suggests
   flipud(myin'),
   %%
   surfc(xx,yy,myin');title('Index of grid points');pause;
   %% show the interpretation of grid
   surfc(xx,yy,xx/hx+yy/hy);title('Affine function of grid-points');
   %% draw just the mesh
   mesh(xx,yy,myin');
%%%%%%%%%%%%%%%%%%%%%%%
function ind=myind(i,j,Mx,My)
    ind=(j-1)*(Mx+1) + i;
end
```

There are other and better ways to do it but we will stick to this format!

Code for solving (24) ... continue previous

```
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Nall = (Mx+1)*(My+1);F = sparse(Nall,1); A = sparse(Nall,Nall);
    U=0*F; Uexact=0*F; interior_nodes = [];
    for i=2:Mx,for j=2:My
            me = myin(i,j); interior_nodes = [interior_nodes,me];
            F(me)= rhsfun(x(i),y(j)); Uexact(me) = exactfun(x(i),y(j));
        end
    end
%%%
    for i=2:Mx, for j=2:My, me = myin(i,j);
            mel=myin(i-1,j);mer=myin(i+1,j);meb=myin(i,j-1);met=myin(i,j+1);
            %%
            A(me,me)=2/hx^2+2/hy^2;
            %% neighbors might be on the boundary.
            A(me,mel)=-1/hx^2;if ismember(mel,interior_nodes),A(mel,me)=-1/hx^2;end
                ....
            A(me,met)=-1/hy^2;if ismember(met,interior_nodes),A(met,me)=-1/hy^2;end
        end
    end
    %% boundary conditions imposed explicitly
    i=1; for j=1:My+1, me = myin(i,j); A(me,me)=1; end
    ...
    j=My+1; for i=1:Mx+1, me = myin(i,j); A(me,me)=1; end
```

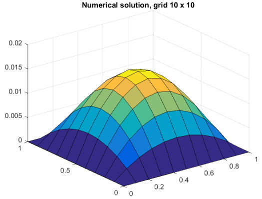
There are more elegant ways to do it but we shortened the code for the exposition's sake. Some code is redundant and could be simplified.

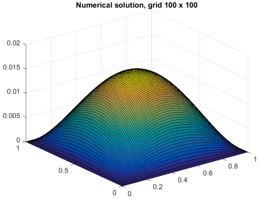
Code for solving (24) ... continue ...

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
...
   U = A\F;
   %% unpack the solution
   for i=1:Mx+1,for j=1:My+1,
    me=myin(i,j);
  Uplot(i,j)=U(me);Uexactplot(i,j)=Uexact(me);
   end; end
  %% plot and check convergence
surf(xx,yy,Uplot');title('Numerical solution');
fprintf('Error =%g\n',norm(Uexact-U,inf));
```

$h = h_x = h_y$	$E(h,\infty)$	$E(h,2)$
0.1	0.00826542	0.00413271
0.05	0.00205871	0.00102935
0.01	8.22508e-05	4.11254e-05

The result for a problem with $u(x,y) = sin(\pi x)\sin(\pi y)$ *is plotted below.*





Addressing Challenge of Complexity

We mentioned before the challenge how to solve $AU = F$.
In MATLAB, we simply wrote `U=A\F`.
This is OK for class, but in general it is an incredible challenge.

Assume $Mx = My = Mz = M, hx = hy = hz = h$

- The matrix $A \in \mathbb{R}^{N\times N}$ is spd and sparse (banded) but $N = O(M^d)$. The band size is $b = O(N^{d-1})$, and factorization (such as LU) will require at least $O(Nb)$ operations.
- The condition number of A increases as $O(h^{-2})$, thus accuracy may be an issue, and iterative solvers such as CG will require many iterations.
- The best solvers for (25) in $d = 2, 3$ are from the *multigrid family* of linear solvers or Fast Poisson solvers (e.g., FFT), which scale as low as $O(N)$. However, these do not always apply well to variable coefficients or grids.
- Will all that is said above, however, in a textbook code for Poisson's problem, the bulk of the time is spent on setting up rather than solving $AU = F$.

Proving convergence of FD for an elliptic equation

We write the error equation

$$AE = \tau. \tag{26}$$

We need to prove (i) consistency, and (ii) stability.

- To prove consistency, we calculate LTE. (For (24) and 5-point stencil D_h^2 we obtain $\tau = O(h_x^2 + h_y^2)$ provided $u \in C^4$)
- To prove stability, we need a bound $\| A^{-1} \|_?$.
  - For a simple region Ω, and Dirichlet BC one can find the eigenvalues and eigenfunctions of $-\Delta$ easily. From this, one can infer the eigenvalues of A similarly as it was done for a two-point bvp. A bound for $\| A^{-1} \|_2$ follows.
  - For more general elliptic problems and BC, general domains Ω, and norms other than "2" this proof may not be trivial.

Addressing Challenge of (non-)Smoothness

Remark 7

FD solution to Poisson's equation (24) using the approach described above (with 5-point stencil in $d = 2$) can be shown to be $O(h^2)$ accurate ($h = max(h_x, h_y, \ldots)$), provided the solution and data are smooth enough. The proof combines consistency analysis and stability.

- When the data (f, coefficients, or Ω) are rough, FD may not be the best numerical approximation method, since likely $u \notin C^4(\Omega)$.
  Instead, one could resort to the weak (variational) formulations of (24), and to other methods such as FE (Finite Elements).
- In contrast, if the solutions are very smooth, one could be interested in a method of higher order than the 5-point stencil FD (only $O(h^2)$ accurate). Possibilities include a larger stencil for FD, or the use of spectral methods. However, these require solving a more dense linear system that that for FD.
- FE and spectral methods approximate the functions $u(x)$ rather than their values $u(x_i)$ at the grid points. Their error is measured in norms such as $L^p(\Omega), C^k(\Omega)$.

Solving general elliptic BVP with FD

Consider

$$-\nabla(K(x,y)\nabla u) + \ldots \text{ lower order terms} = f \tag{27}$$

Remark 8

For this pbm to be elliptic, we need the matrix coefficient $K(x,y) \in \mathbb{R}^{2\times 2}$ to be uniformly positive definite. (Uniformly means the eigenvalues of $K(x,y)$ for all $(x,y) \in \Omega$ are bounded below).

FD approximation to (27)

- If $K : \Omega \to \mathbb{R}$, or is a diagonal matrix, then 5-pt stencil suffices.
- If mixed derivatives are present, 9-pt stencil is needed.
- If higher-order approximation is desired, one can use, e.g., Richardson extrapolation

Ex: find higher order approximation to $-u''$, and extend to $-\Delta u$. What is the band size of the linear system when solving on an $M_x \times M_y$ grid? What should happen near boundaries?

Type of PDE (in $\mathbb{R}^2$)

Canonical types

- elliptic: $u_{\xi\xi} + u_{\eta\eta} = \ldots$,
- hyperbolic: $u_{\xi\xi} - u_{\eta\eta} = \ldots$, or $u_{\xi\eta} = \ldots$
- parabolic: $u_{\xi} - u_{\eta\eta} = \ldots$.

Here $\ldots$ means lower order terms.

Type of a general PDE

If PDE has constant coefficients in the leading terms, we attempt to find a change of variable that will turn it to one of these canonical types. Then we associate with the PDE the type of that canonical type.
If the coefficients are variable, or depend on u, we (try to) do the same, but the PDE may have a different type in different regions of (x, y) or may change type depending on u.

Non-stationary problems

Previously, we considered equilibrium problems such as $-u'' = f$ or $-\Delta u = f$.

These are the stationary analogues of the problems below

- Parabolic (Dissipative): Heat/diffusion equation or in $\Omega \subset \mathbb{R}^d$

$$u_t - \Delta u = f \tag{28}$$

  - Needs BC on $\partial\Omega$ and IC $u(x, 0) = u_0(x)$
  - Finite dimensional analogue: $U' + AU = F, U(0) = U_0$ where A is positive definite

- Hyperbolic (Conservative): Wave equation

$$u_{tt} - \Delta u = 0 \tag{29}$$

  - Needs BC and IC $u(x, 0) = u_0(x)$, $u_t(x, 0) = u_1(x)$
  - Finite dimensional analogue: $U'' + AU = F, U(0) = U_0, U'(0) = U_1$ where A is positive definite

FD for parabolic problems

Consider heat equation $u_t - u_{xx} = f(x,t)$ in $\mathbb{R}$ supplemented with initial and homogeneous Dirichlet boundary conditions

First, we discretize in space on a spatial grid $\Delta = \{x_0, \dots x_M\}$, with $h = \frac{b-a}{M}$, approximating $U_j(t) \approx u(x_j, t)$.

Note change of notation from superscripts U^j to subscripts U_j

The interior equations for $j = 1, \dots M$-1 are found by using $(D_h^{2,x} U(t))_j \approx -u_{xx}(x_j, t)$, with the notation as in (22a).

$$U_j'(t) + \frac{1}{h^2}(2U_j(t) - U_{j-1}(t) - U_{j+1}(t)) = f_j(t), \quad j = 1, \dots M-1 \tag{30a}$$

$$U_0(t) = 0, \quad U_M(t) = 0. \tag{30b}$$

After we eliminate $U_0(t), U_M(t)$ using (30b) from (30a), we can write the resulting discrete system of equations in vector form for $U(t) = [U_1, \dots U_{M-1}]$

$$U' + AU = F. \tag{31}$$

The problem (31) is called semi-discrete.

Discretize in time

Discretization in time on a time grid is applied to (31). For example, we can choose uniform time-stepping with step Δt, and some one-step method with $(D^t_{\Delta t}U)(^n) \approx U'(t_n)$ to yield

$$\frac{1}{\Delta t}(U^n - U^{n-1}) + AU^{n*} = F. \tag{32}$$

Here $n^* = n-1$, or $n* = n$, or $n* = n - \frac{1}{2}$, indicate whether we use an explicit (FE) or implicit method such as BE or (CN) Crank-Nicholson (trapezoidal method).

Example 5.1 (Fully discrete equations (BE))

At every time step $n = 1, 2, \ldots$ we solve

$$\frac{1}{\Delta t}(U^n_j - U^{n-1}) + \frac{1}{h^2}(2U^n_j - U^n_{j-1} - U^n_{j+1}) = f^n_j, \quad j = 1, \ldots M-1 \tag{33a}$$

$$U^n_0 = 0, \quad U^n_M = 0. \tag{33b}$$

One can arrive at (33) by first discretizing in time, and next in space.

`Rewrite an analogue of (33) with trapezoidal or FE time-stepping.`

Choose variant of time-stepping

The choice of time-stepping determines the stability, ease of implementation, and computational complexity.

See summary below and details in the next slides.

- FE is only conditionally stable, $O(\Delta t + h^2)$ accurate, and is inexpensive

$$U^n = U^{n-1} + \Delta t F^{n-1} - \Delta t A U^{n-1} = (I - \Delta t A)U^{n-1} + \Delta t F^{n-1} \quad (34)$$

- BE is always stable and $O(\tau + h^2)$ accurate, but requires a linear solver

$$(I + \Delta t A)U^n = \Delta t F^n + U^n \quad (35)$$

- Trapezoidal (CN) method is always stable and $O(\Delta t^2 + h^2)$ accurate, but requires a solve

$$(I + \frac{\Delta t}{2} A)U^n = \frac{\Delta t}{2}(F^n + F^{n-1}) + (I - \frac{\Delta t}{2} A)U^{n-1} \quad (36)$$

Stability analysis for one-step methods for IVP

For a system of ODEs such as $u' = f(t, u)$, we found that all of FE, BE, and trapezoidal schemes are zero-stable, and this sufficed for convergence, regardless of the Lipschitz constant L of f. However, the error bound increases with time T as $O(T \exp(LT))$.

Example 5.2 (Zero-stability but no convergence for (31))

Here $L = \| A \|$ increases with $O(h^{-2})$ as $h \to 0$, and this does not give convergence unless additional conditions on Δt hold. (The problem is increasingly stiff as $h \to 0$).

On the other hand, for a linear problem (test equation $u' = \lambda u$), recall the region of absolute stability for some one-step method. If $z = \lambda \Delta t$ is in this region, the growth factor $R(z)$ is bounded by 1; and the error bound grows linearly with T regardless of L.

Example 5.3 (Recall $R(z)$ for FE, BE, CN)

These are, respectively, $R(z) = 1 + z$, $R(z) = (1 - z)^{-1}$, $R(z) = \frac{1+\frac{z}{2}}{1-\frac{z}{2}}$.

For a system $u' = Cu$, we would require that z is in the absolute stability region for all eigenvalues $\lambda(C)$.

Stability of time-stepping for the heat equation

In the slides below we consider time-stepping for (31) written as

$$U' = -AU + F. \tag{37}$$

For Dirichlet bc, A is spd, and $\| A \|_2 = \rho(A) = \frac{4}{h^2}$.

Example 5.4 (Growth factor $R(z)$ for FE, BE, CN applied to (37))

These are $R(z) = 1 - \Delta t \lambda(A)$, $R(z) = (1 + \Delta t \lambda(A))^{-1}$, $R(z) = \frac{1 - \frac{\Delta t \lambda(A)}{2}}{1 + \frac{\Delta t \lambda(A)}{2}}$.

It is also helpful to see each of (34), (35), (36) written as

$$U^n = BU^{n-1} + b^n. \tag{38}$$

In the error equation we work with $\| B \|$ to see how the error bound grows. The analysis of $\| B \|_2$ is equivalent to that of $R(z)$ as in Ex. 5.3.

`Ex.: work out the details of a stability constraint for` $u_t - k\Delta u = f$, `with homogeneous Dirichlet or homogeneous Neumann b.c.`

Stability for the heat equation in $\|\cdot\|_2$, cd

Example 5.5 (Conditional stability of FE (34))

Here we need to ensure that the eigenvalues of $B = I - \Delta t A$ given by $1 - \Delta t \lambda(A)$ are bounded by 1. Choosing the largest eigenvalue of A we find asymptotically that these are given by $1 - \Delta t \frac{4}{h^2}$.
A bit of algebra gives that

$\frac{\Delta t}{h^2} \leq \frac{1}{2}$ is the condition for stability of FE for the heat equation.

Example 5.6 (Unconditional stability for BE (35))

We need to show that the eigenvalues of $B = (I + \Delta t A)^{-1}$, given by $\frac{1}{1+\Delta t \lambda(A)}$, are bounded by 1. Since A is positive definite, we find the stability to hold unconditionally for BE.

Example 5.7 (Unconditional stability for CN (36))

Here $B = (I + \frac{\Delta t}{2} A)^{-1}(1 - \frac{\Delta t}{2} A)$. We find that its eigenvalues are given by $\frac{1 - \frac{\Delta t}{2}\lambda(A)}{1 + \frac{\Delta t}{2}\lambda(A)}$. Since A is positive definite, we find that they are bounded by 1 unconditionally for CN.

Compare these results to the analysis of $R(z)$

Convergence theory

Now we write the error equation for FE $E^n = BE^{n-1} + \Delta t \tau^n$.

A similar equation can be derived for BE, CN.

A few steps of inequalities give that $\| E^n \|_2 \to 0$ as $h, \Delta t \to 0$, provided the stability holds.

Theorem 5.8 (Lax-Richtmyer stability and Lax Equivalence Theorem)

Consider the fully discrete method written as (37). The method is strongly stable if $\| B \|_2 \leq 1$, and stable if $\| B^n \|_2 \leq C_T$, where C_T is allowed to grow in time with T. A consistent method converges iff it is stable.

Remark 9 (Summary of convergence for FE, BE, CN)

Based on the analysis above, we find that the results summarized on slide 31 hold.

Implementation: FE for the heat equation

Combine the FE-code (4/552) with code from slide 22.

```
%% set up spatial grid x, matrix A, and initial guess
x = ... ; A .... ; u0 = uexact(x,0); uFEprev = u0; steperr=0;
%% set up the discretization
T=1;tsteps=(0:dt:T)';nsteps=length(tsteps);
```

```
%% time stepping loop
for n=2:nsteps
%% calculate rhside; change boundary conditions if needed
   f = rhsfun(x,tsteps(n-1)); ...
%% calculate the numerical solution, new time step
   uFE = uFEprev + dt*f -dt*A*uFEprev;
%% plot, calculate error
   plot(x,uexactsol,'-',x,uFE,'*'); steperr= max(steperr,...)...
   uFEprev = uFE;
end
%% report on the error
...
```

Ex: implement BE: structure quite different

Another approach to stability in $\|\cdot\|_2$ via Fourier analysis

For linear PDEs with constant coefficients, Fourier analysis (and the associated von-Neumann Ansatz) are quite useful. We pose the PDE (28) on $\mathbb{R}$, or impose periodic b.c., and assume homogeneity $f = 0$.

- Von-Neumann Ansatz: assume $U_j^{n-1} = e^{ijh\xi}$ for FD discretization of (28).
- Plug this to the discrete equation for each j.
- Calculate the amplification factor $\rho(\xi)$ so that $U_j^n = \rho(\xi)U_j^{n-1}$.
- Find conditions on Δt so that $|\rho(\xi)| \leq 1$.

Example 5.9 (FE conditional stability via von-Neumann)

We find $\rho(\xi) = 1 - 2\frac{2\tau}{h^2}(1 - cos(\xi h))$, and after some algebra we find the same stability condition as in Ex. 5.5.

`Ex: Carry out the calculations of` $\rho(\xi)$ `for BE and CN methods`

`Ex: For multi-step methods such as DuFort-Frankel, consider the amplification matrix` $G(\xi)$`, and check its norm`

Why von-Neumann works; discrete Fourier transform

Continuous Fourier transform $L^2(\mathbb{R}) \ni S \to \hat{S} \in L^2(\mathbb{R})$

... transforms functions $S(x)$ to functions $\hat{S}(\xi) := \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} s(x)e^{-i\xi x}dx$.
Via Parseval's identify $\| S \|_2 = \| \hat{S} \|_2$.

Analysis of FD: use discrete Fourier transform

We are interested in grid functions on h-grid,
$l_2(\mathbb{R}) \ni S = (S_j)_j \to \hat{S} = (\hat{S}(\xi))_\xi \in L^2(-\pi/h, \pi/h)$ defined as
$\hat{S}(\xi) = \frac{h}{\sqrt{2\pi}} \sum_j S_j e^{-ijh\xi}$, and $S_j = \frac{1}{\sqrt{2\pi}} \int_{-\pi/h}^{\pi/h} \hat{S}(\xi)e^{ijh\xi}d\xi$.
The analogue of Parseval's identity holds

$$\| S \|_{\Delta,2} = \left(h \sum_j | S_j |^2 \right)^{\frac{1}{2}} = \| \hat{S} \|_2 = \left(\int_{-\pi/h}^{\pi/h} | \hat{S}(\xi) |^2 \right)^{\frac{1}{2}} \tag{39}$$

Example 5.10 (Apply Fourier analysis to analyze the growth $U^{n-1} \to U^n$)

Consider U^n defined pointwise in (33a). Apply discrete F-transform to (33a) $U^n \to \hat{U}^n$ and study the growth factor $\rho(\xi)$ so that $\hat{U}^n(\xi) = \rho(\xi)\hat{U}^{n-1}(\xi)$. Thanks to the Parseval's identity we get $\| U^n \|_{\Delta,2} = | \rho(\xi) | \| U^{n-1} \|_{\Delta,2}$. It turns out this is equivalent to the von-Neumann Ansatz!

Practical way to use von-Neumann Ansatz

Sometimes instead of writing $U_j^n = e^{ijh\xi}$ as on slide 37, and deriving $\rho(\xi)$ to have $U_j^n = \rho(\xi)U_j^{n-1}$, it is convenient to write

$$U_j^n = \rho_n e^{-ijh\xi}, \;\; U_j^{n-1} = \rho_{n-1} e^{-ijh\xi} \tag{40}$$

where ρ_n, ρ_{n-1} are the amplitudes of the grid value U_j^n, U_j^{n-1} respectively. Then we find the amplification factor $\rho(\xi) = \frac{\rho_n}{\rho_{n-1}}$.

This technique is useful when dealing with multi-step schemes. For multi-step schemes, we derive an amplification matrix $G(\xi)$ which replaces $\rho(\xi)$. For stability then we want $\|G(\xi)\|_2 \leq 1$.

Example 5.11 (FE, BE, and CN with von Neumann Ansatz for the heat equation)

...carry out the calculations .... compare to the analysis via MOL and $\mathcal{R}^{ABS}$.

`Ex.:` Derive the amplification matrix $G(\xi)$ for the midpoint method applied to the heat equation. To deal with the steps U^n, U^{n-1}, U^{n-2}, rename $U^{n-2} = V^{n-1}$, and rewrite your equations as a system involving U^n, U^{n-1}, and V^{n-1} in one equation, and $V^n = U^{n-1}$ in the second equation. Attempt to derive conditions on Δt so that $\|G(\xi)\|_2 \leq 1$.

FD for advection (convection) equation

We start with the first order homogeneous hyperbolic equation, with constant coefficients

$$u_t + a u_x = 0. \tag{41}$$

It must be supplemented with some auxiliary (non-characteristic) data in order to be well posed.

Example 6.1 (Advection pbm with initial data $u(x,0) = u_0(x), x \in \mathbb{R}$)

If you supplement (41) with IC, and if $u_0(\cdot)$ is smooth enough, then $u(x,t) = u_0(x - at)$ is the unique solution for $x \in \mathbb{R}, t \in \mathbb{R}$. The lines $x - at = const$ are the characteristics along which the solution remains constant.

Example 6.2 (Advection on a bounded domain, inflow condition)

When $x \in (0, L)$, we require boundary conditions for (41), such as on the inflow boundary. If $a > 0$, the inflow boundary is at $x = 0$. If $a < 0$, it is at $x = L$.

$$u_t + a u_x = 0, \;\; x \in (0, L), t > 0; \;\; u(x,u) = u_0(x); \;\; u(0,t) = g_0(t). \tag{42}$$

To have a smooth solution, compatibility $u_0(0) = g_0(0)$ must be imposed in (42)

Additional BC, and FD discretization

Example 6.3 (Advection with periodic BC)

Periodic boundary condition can be imposed as well

$$u_t + au_x = 0, \;\; u(x,0) = u_0(x), u(0,t) = u(L,t); t > 0, \;\; x \in (0,L), t > 0. \qquad (43)$$

These are less common in practical application. However, analysis of FD schemes for (43) avoids a lot of technical non-essential calculations that are required for (42).

FD for (41): a plethora of schemes

The experience you gained so far suggests many different ways to discretize (41), leading to different (consistency) order of LTE, and corresponding to different stencils. It is clear however that stability is a crucial factor determining whether a scheme is convergent. We will assess it via MOL and von-Neumann analysis.

First, we discretize in space, and consider a uniform grid over $(0,L)$ discretized with $x_j = jh$, $j = 0, 1, \ldots M$, and $h = \frac{L}{M}$.
We also set $\nu = \frac{a\Delta t}{h}$ (the “Courant number” of CFL number). The value of ν is usually constrained to ensure stability.

Simplest code for advection equation (43)

Use first order conditionally stable scheme (analysis later)

```
%% set up initial data
x=(0:dx:1)'; uprev = uinit(x); unew = 0*uprev;t = 0; n= 0;
%%
plot(x,uprev);title(sprintf('Initial condition at t=%g',t));pause;
while t < 1
%% advance to new time step
    t = t + dt; n = n+1;
    unew(2:end) = uprev(2:end) - nu*(uprev(2:end)-uprev(1:end-1));
    unew(1) = uprev(end);
    %% plot, compare with true solution ...?
    plot(x,unew);title(sprintf('Numerical solution at t=%g',t));pause;
    %% calculate error
    uprev = unew;
end
end
```

This code can be made even more compact for faster performance. For example, you can use `circshift`*. This code can also be rewritten without vectorization with a loop, so it is more clear but slower. Compare!*

MOL analysis for advection equation

We approximate the solution vector $U(t) = (U_j(t))_{j=0}^{M}$ using the PDE and the boundary conditions. Either way, the rank of matrix A is M. A semidiscrete scheme for (42) or (43) reads

$$U'(t) + AU = 0. \tag{44}$$

- The matrix A depends on the particular difference quotient used for u_x, and on the boundary conditions. The decision how to approximate $u_t|_{(x_j,t_n)}$ gives rise to a variety of schemes, with different consistency and different stability properties.
- Next, we discretize (44) in time for $0 = t_0 < t_1, \ldots t_n = n\Delta t, \ldots$.
- When choosing a time discretization scheme, we must ensure that the scheme is Lax-Richtmyer stable. This will depend both on the time discretization scheme as well as on A.
- In the end, the scheme we implement is fully discrete in space and time and finds $U_j^n \approx u(x_j, t_n)$.

One-sided scheme for au_x at (x_j, t_n)

We have either $au_x|_{(x_j,t_n)} \approx a\frac{U^n_{j+1}-U^n_j}{h}$ (downwind)
or $au_x|_{(x_j,t_n)} \approx a\frac{U^n_j-U^n_{j-1}}{h}$ (upwind).
If $a > 0$, which do you think makes more sense?
Fully discrete schemes:

- $u_t|_{(x_j,t_n)} \approx \frac{U^{n+1}_j-U^n_j}{\Delta t}$ is $O(\Delta t + h)$ accurate, but leads to FE scheme with $\rho(\xi) = \dots$
- $u_t|_{(x_j,t_n)} \approx \frac{U^{n+1}_j-U^{n-1}_j}{2\Delta t}$ would be $O(\Delta t^2 + h)$ accurate. But, calculate $G(\xi) =$ ... (amplification matrix)
- $u_t|_{(x_j,t_n)} \approx \frac{U^n_j-U^{n-1}_j}{\Delta t}$ is $O(\Delta t + h)$ accurate, and is an implicit scheme with $\rho(\xi) = \dots$ (see below)

`Verify the LTE order as stated and check the stability calculations via MOL or von-Neumann`

Central scheme for au_x at (x_j, t_n)

With the central scheme, we have $au_x|_{(x_j,t_n)} \approx a\frac{U^n_{j+1}-U^n_{j-1}}{2h}$. The decision how to approximate $u_t|_{(x_j,t_n)}$ gives rise to a variety of schemes, with different consistency and different stability properties.

- $u_t|_{(x_j,t_n)} \approx \frac{U_j^{n+1}-U_j^n}{\Delta t}$ is $O(\Delta t + h^2)$ accurate, but leads to FE scheme with $\rho(\xi) = \ldots$ (see below)
- $u_t|_{(x_j,t_n)} \approx \frac{U_j^{n+1}-U_j^{n-1}}{2\Delta t}$ is second order accurate, and is the midpoint scheme (in time). The method is called "leapfrog". To analyze via von-Neumann we need to use the two-step approach outlined on slide 39. Instead, we can consider checking the region $\mathcal{R}^{ABS}$ for the midpoint method for the eigenvalues.
- $u_t|_{(x_j,t_n)} \approx \frac{U_j^{n+1}-\frac{1}{2}(U^n_{j-1}+U^n_{j+1})}{\Delta t}$ is consistent (see below), and is known as Lax-Friedrichs scheme. We calculate $\rho(\xi) = \cos(\xi h) - \nu i \sin(\xi h)$.

```
Calculate the LTE for the Lax-Friedrichs scheme.  Also, go over
stability calculations for the other schemes listed here and the next,
including their time-implicit versions
```

Methods which correct instabilities: LW and BW

As you (likely) saw, an explicit in time scheme combined with central difference in space is unstable, period. Instability could be corrected by adding "just enough" (numerical) diffusion to correct it.

Example 6.4 (Lax-Wendroff; requires $|\nu| \leq 1$.)

By choosing just the right amount of diffusion $D = \frac{a^2 \Delta t}{2}$, the LW scheme also becomes higher order accurate.

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_{j-1}^n}{2h} + \frac{a^2 \Delta t}{2h^2}(2U_j^n - U_{j-1}^n - U_{j+1}^n) = 0 \tag{45}$$

Example 6.5 (Beam-Warming; requires $|\nu| \leq 2$ and $sign(\nu) = sign(a)$.)

Another possibility is to correct the one-sided upwind scheme with a higher order one-sided scheme in space, and correct (the instability) similarly as LW.

`Carry out the details please to convince yourself of the claims made on LW and BW.`

Modified equation: reveals issues

Modified equation is the PDE satisfied (approximately) by a function $v(x,t)$ which satisfies exactly the finite difference scheme. (Unlike in the LTE analysis where we look for the error of when the exact solution to the PDE satisfies approximately the scheme).

Modified equations are typically either

- diffusion type $v_t + av_x = Dv_{xx}$, or
- dispersive $v_t + av_x = \mu v_{xxx}$.

Example 6.6 (The upwind and Lax-Friedrichs schemes are "diffusive")

... with $D^{upwind} = \frac{ah}{2}(1-\nu)$, and $D^{LF} = \frac{h^2}{2\Delta t}(1-\nu^2)$.

Example 6.7 (Lax-Wendroff and Beam-Warming are "dispersive")

.. with $\mu = O(\frac{h^2}{2}a(\nu^2-1))$, and $\mu = O(\frac{h^2}{6}a(2-3\nu+\nu^2))$.

`Be sure to know the details of the calculations of the modified equation`

Activity on schemes for $u_t + a u_x = 0,\ a > 0$

Connect the Method with its Stencil, LTE, Stability & Properties

Note: the cells in the table below are NOT aligned.

(1) FTBS (upwind)	**A**	**(a)**	**I.** never stable
(2) BTBS	**B**	**(b)** $O(\Delta t + h)$	**II.** stable if $0 \le \nu \le 1$, diffusive
(3) CTBS	**C**	**(c)** $O(\Delta t^2 + h^2)$	**III.** stable if ..., dispersive
(4) CTCS (leapfrog)	**D**	**(d)**	**IV.**
(5) FTCS	**E**	**(e)** $O(\Delta t^2 + h)$	**V.** stable if $\lvert\nu\rvert \le 1$, oscillations
....	**F**	**(f)** $O(\Delta t + h)$	**VI.**
(6) LF	**G**	**(g)**	**VII.** stable if $0 \le \nu \le 1$, very diffusive
(7) LW	**H**	**(h)** $O(\Delta t + h^2)$	**VIII.** unstable unless ...
(8) BW			

FD for mixed equations: ADR

Advection-diffusion-reaction problem

$$u_t + au_x - Du_{xx} + ru = f(x), \;\; x \in (0,1), \; t > 0. \tag{46}$$

Here we assume $D > 0$ (for well-posedness). Also, wlog, $a \geq 0$. The reaction term coefficient $r \geq 0$ for decay/absorption, and $r \leq 0$ for growth.

The problem requires an I.C.

$$u(x,0) = u_{init}(x), \; x \in (0,1), \tag{47}$$

and some boundary conditions. Assume homogeneous Dirichlet condition at $x = 0$.

$$u(0,t) = 0, \;\; t > 0. \tag{48}$$

On the outflow end $x = 1$

- Dirichlet condition $u(1,t) = 0$ may lead to a boundary layer at $x = 1$.
- Outflow condition $Du_x(1,t) = 0$ allows the solution to “smoothly” cross the boundary $x = 1$.

Finite difference scheme(s) for ADR

It is straightforward to apply the FD to (46), by using the discretization methods that worked well for the individual pieces, while observing the consistency order and stability of the schemes as they apply to the subproblems.

- Fully implicit schemes are unconditionally stable but ...
- Fully explicit schemes are (perhaps) conditionally stable if ...
- Implicit-explicit schemes can combine advantages of the individual schemes

Fractional step methods (operator splitting)

...allow to solve subproblems independently. A common (first order accurate) approach for (46) is to solve them in steps A→R→D, using U^n as data for step A which delivers U^A, to be used for R step, which produces U^R, to be used as data for D step. The last solves for U^{n+1}. Other splitting algorithms are possible.

Recap of what we've done

- We focused on: scalar linear PDEs with constant coefficients, of canonical type, with smooth solutions
  - hyperbolic PDEs are conservative and parabolic are dissipative
  - elliptic PDEs are stationary solutions to the transient pbms
  - "PDEs are easy, boundary conditions are hard"

Current challenges in PDEs involve solving nonlinear coupled systems of mixed type, with nonsmooth (very weak) solutions

- We focused on the following numerical (class of) methods: FD. FD require (high) smoothness of true solutions for optimal convergence. FD approximate the values $u(x_j)$ rather than the function $u(x)$

Other methods may be better suited to nonlinear non-smooth problems with heterogeneous data

In NA/scientific computing, we deal with the compromise...

... between accuracy and efficiency. High accuracy requires a lot of computational time.