AM 205: lecture 14

- Last time: Additional ODE methods, boundary value problems
- Today: Numerical solution of PDEs

ODE BVPs

A more general approach is to formulate a coupled system of equations for the BVP based on a finite difference approximation

Suppose we have a grid $x_i = a + ih$, $i = 0, 1, \ldots, n-1$, where $h = (b-a)/(n-1)$

Then our approximation to $u \in C^2[a, b]$ is represented by a vector $U \in \mathbb{R}^n$, where $U_i \approx u(x_i)$

ODE BVPs

Recall the ODE:

$$-\alpha u''(x) + \beta u'(x) + \gamma u(x) = f(x), \quad x \in [a, b]$$

Let's develop an approximation for each term in the ODE

For the reaction term γu, we have the pointwise approximation
$\gamma U_i \approx \gamma u(x_i)$

ODE BVPs

Similarly, for the derivative terms:

- Let $D_2 \in \mathbb{R}^{n \times n}$ denote diff. matrix for the second derivative
- Let $D_1 \in \mathbb{R}^{n \times n}$ denote diff. matrix for the first derivative

Then $-\alpha(D_2 U)_i \approx -\alpha u''(x_i)$ and $\beta(D_1 U)_i \approx \beta u'(x_i)$

Hence, we obtain $(AU)_i \approx -\alpha u''(x_i) + \beta u'(x_i) + \gamma u(x_i)$, where $A \in \mathbb{R}^{n \times n}$ is:

$$A \equiv -\alpha D_2 + \beta D_1 + \gamma \mathrm{I}$$

Similarly, we represent the right hand side by sampling f at the grid points, hence we introduce $F \in \mathbb{R}^n$, where $F_i = f(x_i)$

ODE BVPs

Therefore, we obtain the linear[1] system for $U \in \mathbb{R}^n$:

$$AU = F$$

Hence, we have converted a linear differential equation into a linear algebraic equation

(Similarly we can convert a nonlinear differential equation into a nonlinear algebraic system)

However, we are not finished yet, need to account for the boundary conditions!

[1] It is linear here since the ODE BVP is linear

ODE BVPs

Dirichlet boundary conditions: we need to impose $U_0 = c_1$, $U_{n-1} = c_2$

Since we fix U_0 and U_{n-1}, they are no longer variables: we should eliminate them from our linear system

However, instead of removing rows and columns from A, it is slightly simpler from the implementational point of view to:

- "zero out" first row of A, then set $A(0,0) = 1$ and $F_0 = c_1$
- "zero out" last row of A, then set $A(n-1, n-1) = 1$ and $F_{n-1} = c_2$

ODE BVPs

We can implement the above strategy for $AU = F$ in Python

Useful trick[2] for checking your code:

1. choose a solution u that satisfies the BCs
2. substitute into the ODE to get a right-hand side f
3. compute the ODE approximation with f from step 2
4. verify that you get the expected convergence rate for the approximation to u

e.g. consider $x \in [0,1]$ and set $u(x) = e^x \sin(2\pi x)$:

$$\begin{aligned} f(x) &\equiv -\alpha u''(x) + \beta u'(x) + \gamma u(x) \\ &= -\alpha e^x \left[4\pi \cos(2\pi x) + (1 - 4\pi^2) \sin(2\pi x)\right] + \\ &\quad\ \beta e^x \left[\sin(2\pi x) + 2\pi \cos(2\pi x)\right] + \gamma e^x \sin(2\pi x) \end{aligned}$$

[2] Sometimes called the "method of manufactured solutions"

ODE BVPs

Python example: ODE BVP via finite differences

Convergence results:

h	error
2.0e-2	5.07e-3
1.0e-2	1.26e-3
5.0e-3	3.17e-4
2.5e-3	7.92e-5

$O(h^2)$, as expected due to second order differentiation matrices

ODE BVPs: BCs involving derivatives

Question: How would we impose the Robin boundary condition $u'(b) + c_2 u(b) = c_3$, and preserve the $O(h^2)$ convergence rate?

Option 1: Introduce a "ghost node" at $x_{n+1} = b + h$, this node is involved in both the B.C. and the n^{th} matrix row

Employ central difference approx. to $u'(b)$ to get approx. B.C.:

$$\frac{U_{n+1} - U_{n-1}}{2h} + c_2 U_n = c_3,$$

or equivalently

$$U_{n+1} = U_{n-1} - 2hc_2 U_n + 2hc_3$$

ODE BVPs: BCs involving derivatives

The n^{th} equation is

$$-\alpha \frac{U_{n-1} - 2U_n + U_{n+1}}{h^2} + \beta \frac{U_{n+1} - U_{n-1}}{2h} + \gamma U_n = F_n$$

We can substitute our expression for U_{n+1} into the above equation, and hence eliminate U_{n+1}:

$$\left(-\frac{2\alpha c_3}{h} + \beta c_3\right) - \frac{2\alpha}{h^2} U_{n-1} + \left(\frac{2\alpha}{h^2}(1 + hc_2) - \beta c_2 + \gamma\right) U_n = F_n$$

Set $F_n \leftarrow F_n - \left(-\frac{2\alpha c_3}{h} + \beta c_3\right)$, we get $n \times n$ system $AU = F$

Option 2: Use a one-sided difference formula for $u'(b)$ in the Robin BC, as in $\mathrm{III}.2$

Partial Differential Equations

PDEs

As discussed in III.1, it is a natural extension to consider Partial Differential Equations (PDEs)

There are three main classes of PDEs:[3]

equation type	prototypical example	equation
hyperbolic	wave equation	$u_{tt} - u_{xx} = 0$
parabolic	heat equation	$u_t - u_{xx} = f$
elliptic	Poisson equation	$u_{xx} + u_{yy} = f$

Question: Where do these names come from?

[3]Notation: $u_x \equiv \frac{\partial u}{\partial x}$, $u_{xy} \equiv \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right)$

PDEs

Answer: The names are related to conic sections

General second-order PDEs have the form

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu + g = 0$$

This "looks like" the quadratic function

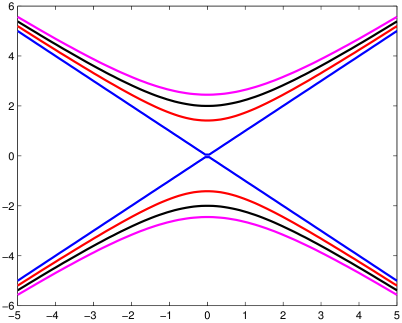
$$q(x) = ax^2 + bxy + cy^2 + dx + ey$$

PDEs: Hyperbolic

Wave equation: $u_{tt} - u_{xx} = 0$

Corresponding quadratic function is $q(x,t) = t^2 - x^2$

$q(x,t) = c$ gives a hyperbola, *e.g.* for $c = 0 : 2 : 6$, we have

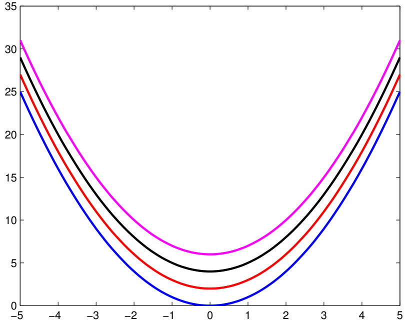


PDEs: Parabolic

Heat equation: $u_t - u_{xx} = 0$

Corresponding quadratic function is $q(x, t) = t - x^2$

$q(x, t) = c$ gives a parabola, *e.g.* for $c = 0 : 2 : 6$, we have

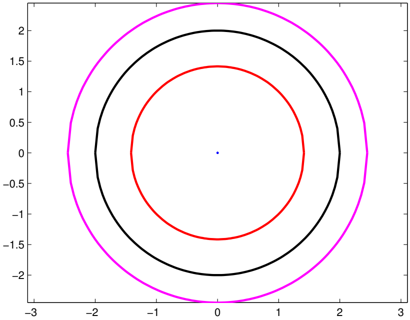


PDEs: Elliptic

Poisson equation: $u_{xx} + u_{yy} = f$

Corresponding quadratic function is $q(x, y) = x^2 + y^2$

$q(x, y) = c$ gives an ellipse, *e.g.* for $c = 0 : 2 : 6$, we have



PDEs

In general, it is not so easy to classify PDEs using conic section naming

Many problems don't strictly fit into the classification scheme (*e.g.* nonlinear, or higher order, or variable coefficient equations)

Nevertheless, the names hyperbolic, parabolic, elliptic are the standard ways of describing PDEs, based on the criteria:

- Hyperbolic: time-dependent, conservative physical process, no steady state
- Parabolic: time-dependent, dissipative physical process, evolves towards steady state
- Elliptic: describes systems at equilibrium/steady-state

Hyperbolic PDEs

Hyperbolic PDEs

We introduced the wave equation $u_{tt} - u_{xx} = 0$ above

Note that the system of first order PDEs

$$\begin{aligned} u_t + v_x &= 0 \\ v_t + u_x &= 0 \end{aligned}$$

is equivalent to the wave equation, since

$$u_{tt} = (u_t)_t = (-v_x)_t = -(v_t)_x = -(-u_x)_x = u_{xx}$$

(This assumes that u, v are smooth enough for us to switch the order of the partial derivatives)

Hyperbolic PDEs

Hence we shall focus on the so-called linear advection equation

$$u_t + cu_x = 0$$

with initial condition $u(x, 0) = u_0(x)$, and $c \in \mathbb{R}$

This equation is representative of hyperbolic PDEs in general

It's a first order PDE, hence doesn't fit our conic section description, but it is:

- time-dependent
- conservative
- not evolving toward steady state

$\Longrightarrow$ hyperbolic!

Hyperbolic PDEs

We can see that $u(x,t) = u_0(x - ct)$ satisfies the PDE

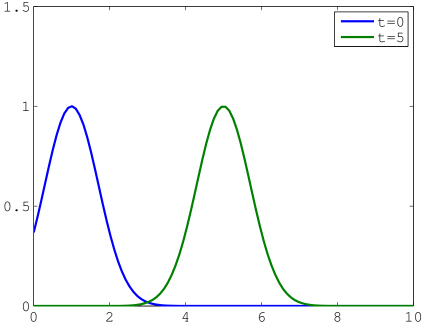
Let $z(x,t) \equiv x - ct$, then from the chain rule we have

$$\begin{aligned}
\frac{\partial}{\partial t} u_0(x - ct) + c \frac{\partial}{\partial x} u_0(x - ct) &= \frac{\partial}{\partial t} u_0(z(x,t)) + c \frac{\partial}{\partial x} u_0(z(x,t)) \\
&= u_0'(z) \frac{\partial z}{\partial t} + c u_0'(z) \frac{\partial z}{\partial x} \\
&= -c u_0'(z) + c u_0'(z) \\
&= 0
\end{aligned}$$

Hyperbolic PDEs

This tells us that the solution transports (or advects) the initial condition with "speed" c

e.g. with $c = 1$ and an initial condition $u_0(x) = e^{-(1-x)^2}$ we have:



Hyperbolic PDEs

We can understand the behavior of hyperbolic PDEs in more detail by considering characteristics

Characteristics are paths in the xt-plane — denoted by $(X(t), t)$ — on which the solution is constant

For $u_t + cu_x = 0$ we have $X(t) = X_0 + ct$,[4] since

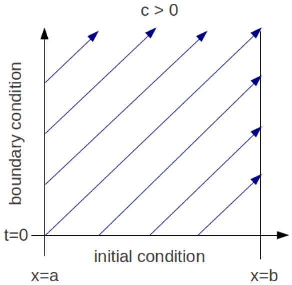
$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}u(X(t),t) &= u_t(X(t),t) + u_x(X(t),t)\frac{\mathrm{d}X(t)}{\mathrm{d}t} \\ &= u_t(X(t),t) + cu_x(X(t),t) \\ &= 0 \end{aligned}$$

[4]Each different choice of X_0 gives a distinct characteristic curve

Hyperbolic PDEs

Hence $u(X(t), t) = u(X(0), 0) = u_0(X_0)$, i.e. the initial condition is transported along characteristics

Characteristics have important implications for the direction of flow of information, and for boundary conditions

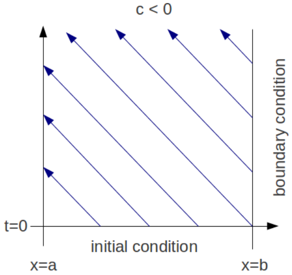


Must impose BC at $x = a$, cannot impose BC at $x = b$

Hyperbolic PDEs

Hence $u(X(t), t) = u(0, X(0)) = u_0(X_0)$, i.e. the initial condition is transported along characteristics

Characteristics have important implications for the direction of flow of information, and for boundary conditions



Must impose BC at $x = b$, cannot impose BC at $x = a$

Hyperbolic PDEs: More Complicated Characteristics

More generally, if we have a non-zero right-hand side in the PDE, then the situation is a bit more complicated on each characteristic

Consider $u_t + cu_x = f(t, x, u(t, x))$, and $X(t) = X_0 + ct$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}u(X(t), t) &= u_t(X(t), t) + u_x(X(t), t)\frac{\mathrm{d}X(t)}{\mathrm{d}t} \\
&= u_t(X(t), t) + cu_x(X(t), t) \\
&= f(t, X(t), u(X(t), t))
\end{aligned}$$

In this case, the solution is no longer constant on $(X(t), t)$, but we have reduced a PDE to a set of ODEs, so that:

$$u(X(t), t) = u_0(X_0) + \int_0^t f(t, X(t), u(X(t), t)\mathrm{d}t$$

Hyperbolic PDEs: More Complicated Characteristics

We can also find characteristics for variable coefficient advection

Exercise: Verify that the characteristic curve for $u_t + c(t, x)u_x = 0$ is given by

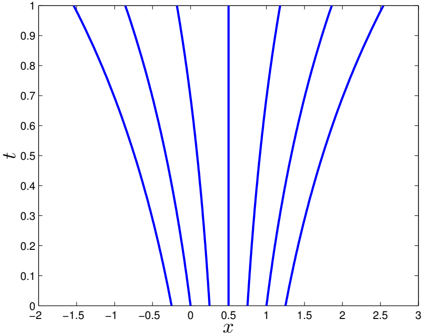
$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = c(X(t), t)$$

In this case, we have to solve an ODE to obtain the curve $(X(t), t)$ in the xt-plane

Hyperbolic PDEs: More Complicated Characteristics

e.g. for $c(t, x) = x - 1/2$, we get $X(t) = 1/2 + (X_0 - 1/2)e^t$

In this case, the characteristics "bend away" from $x = 1/2$



Characteristics also apply to nonlinear hyperbolic PDEs (*e.g.* Burger's equation), but this is outside the scope of AM205

Hyperbolic PDEs: Numerical Approximation

We now consider how to solve $u_t + cu_x = 0$ equation using a finite difference method

Question: Why finite differences? Why not just use characteristics?

Answer: Characteristics actually are a viable option for computational methods, and are used in practice

However, characteristic methods can become very complicated in 2D or 3D, or for nonlinear problems

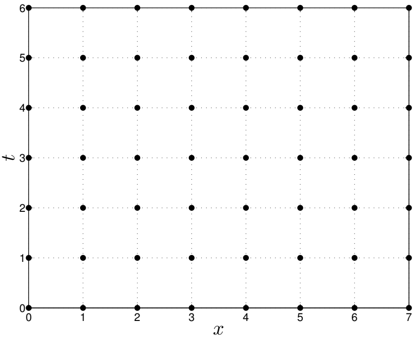
Finite differences are a much more practical choice in most circumstances

Hyperbolic PDEs: Numerical Approximation

Advection equation is an Initial Boundary Value Problem (IBVP)

We impose an initial condition, and a boundary condition (only one BC since first order PDE)

A finite difference approximation leads to a grid in the xt-plane



Hyperbolic PDEs: Numerical Approximation

The first step in developing a finite difference approximation for the advection equation is to consider the CFL condition[5]

The CFL condition is a necessary condition for the convergence of a finite difference approximation of a hyperbolic problem

Suppose we discretize $u_t + cu_x = 0$ in space and time using the explicit (in time) scheme

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

Here $U_j^n \approx u(t_n, x_j)$, where $t_n = n\Delta t$, $x_j = j\Delta x$

[5]Courant-Friedrichs-Lewy condition, published in 1928

Hyperbolic PDEs: Numerical Approximation

This can be rewritten as

$$\begin{aligned} U_j^{n+1} &= U_j^n - \frac{c\Delta t}{\Delta x}(U_j^n - U_{j-1}^n) \\ &= (1-\nu)U_j^n + \nu U_{j-1}^n \end{aligned}$$

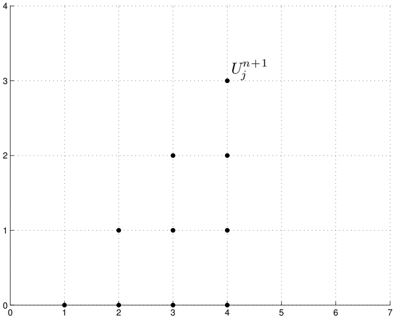
where

$$\nu \equiv \frac{c\Delta t}{\Delta x}$$

We can see that U_j^{n+1} depends only on U_j^n and U_{j-1}^n

Hyperbolic PDEs: Numerical Approximation

Definition: Domain of dependence of U_j^{n+1} is the set of values that U_j^{n+1} depends on



Hyperbolic PDEs: Numerical Approximation

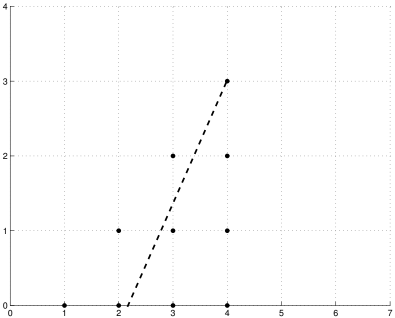
The domain of dependence of the exact solution $u(t_{n+1}, x_j)$ is determined by the characteristic curve passing through (t_{n+1}, x_j)

CFL Condition:

> For a convergent scheme, the domain of dependence of the PDE must lie within the domain of dependence of the numerical method

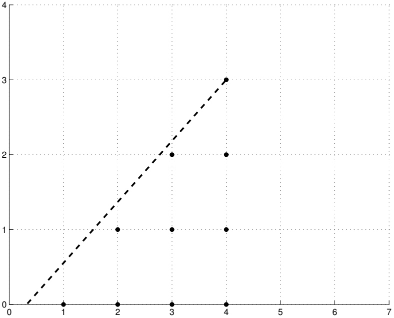
Hyperbolic PDEs: Numerical Approximation

Suppose the dashed line indicates characteristic passing through (t_{n+1}, x_j), then the scheme below satisfies the CFL condition



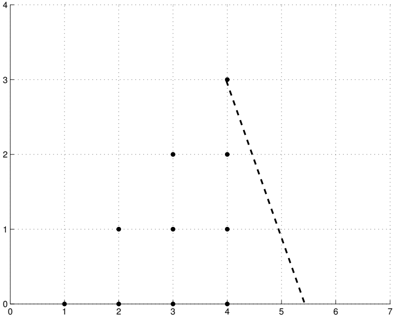
Hyperbolic PDEs: Numerical Approximation

The scheme below does not satisfy the CFL condition



Hyperbolic PDEs: Numerical Approximation

The scheme below does not satisfy the CFL condition (here $c < 0$)



Hyperbolic PDEs: Numerical Approximation

Question: What goes wrong if the CFL condition is violated?

Hyperbolic PDEs: Numerical Approximation

Answer: The exact solution $u(x, t)$ depends on initial value $u_0(x_0)$, which is outside the numerical method's domain of dependence

Therefore, the numerical approx. to $u(x, t)$ is "insensitive" to the value $u_0(x_0)$, which means that the method cannot be convergent

Hyperbolic PDEs: Numerical Approximation

Note that CFL is only a necessary condition for convergence

Its great value is its simplicity: CFL allows us to easily reject F.D. schemes for hyperbolic problems with very little investigation

For example, for $u_t + cu_x = 0$, the scheme

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + c\frac{U_j^n - U_{j-1}^n}{\Delta x} = 0 \qquad (*)$$

cannot be convergent if $c < 0$

Question: What small change to $(*)$ would give a better method when $c < 0$?

Hyperbolic PDEs: Numerical Approximation

If $c > 0$, then we require $\nu \equiv \frac{c\Delta t}{\Delta x} \leq 1$ in $(*)$ for CFL to be satisfied