Office hours: W 11-12 LCB 116, R 1-2pm WEB 4666

Math 6620: Analysis of Numerical Methods, II
Background and Review: PDEs

Akil Narayan[1]

[1] Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

THE UNIVERSITY OF UTAH

SCI www.sci.utah.edu

"Prerequisites"

Several topics are background for this course:

- (Numerical) linear algebra
- Calculus
- "Basic" knowledge of ordinary/partial differential equations
- Some programming experience

From the previous course, 6610, you'll be expected to have familiarity with:

- linear algebraic factorizations
- (polynomial) truncation error and numerical approximation (e.g., of derivatives)
- quadrature rules
- computational considerations for solving linear and nonlinear systems

We'll spend some time briefly reviewing (small) portions of these.

Notation

When considering PDEs, we'll generally work over d-dimensional physical space, $d = 1, 2, 3$, with variables x, y, z. We'll use t as the time variable for time-dependent problems.

Generally we'll refer to state (unknown) functions as u, e.g.,

$$u(x), \quad u(x, y), \quad u(t, x), \quad u(t, x, y, z)$$

Notation

When considering PDEs, we'll generally work over d-dimensional physical space, $d = 1, 2, 3$, with variables x, y, z. We'll use t as the time variable for time-dependent problems.

Generally we'll refer to state (unknown) functions as u, e.g.,

$$u(x), \quad u(x, y), \quad u(t, x), \quad u(t, x, y, z)$$

(Partial) Derivatives are abbreviated with subscripts,

$$\frac{\partial u}{\partial t} = u_t, \qquad \frac{\partial^3 u}{\partial x^3} = u_{xxx}.$$

Ordinary or partial differential equations (ODEs/PDEs) are mathematical laws governing an unknown u that model some phenomenon.

$$u_t = u_{xx}, \qquad \text{(Heat diffusion in 1D)}$$
$$u_{tt} = u_{xx}, \qquad \text{(Wave motion in 1D)}$$
$$u_{xx} + u_{yy} = 0, \qquad \text{(Steady-state temperature in 2D)}$$

Differential equations prescribe incomplete knowledge of u without initial/boundary conditions.

Ordinary or partial differential equations (ODEs/PDEs) are mathematical laws governing an unknown u that model some phenomenon.

$$u_t = u_{xx}, \qquad \text{(Heat diffusion in 1D)}$$

$$u_{tt} = u_{xx}, \qquad \text{(Wave motion in 1D)}$$

$$u_{xx} + u_{yy} = 0, \qquad \text{(Steady-state temperature in 2D)}$$

Differential equations prescribe incomplete knowledge of u without initial/boundary conditions.

Differential equations can have *input* parameters, e.g., a scalar coefficient or a function.

$$u_{xx} = f(x), \qquad u_t = \kappa u_{xx},$$

At a high level, one can view the task of solving a differential equation as a map from inputs (e.g., f, κ) to outputs (the solution u).

We can view the task of solving PDEs as a function from inputs to outputs:

$$\text{Inputs, e.g., } f,\ \kappa \xrightarrow{\text{Solution map}} \text{Solution } u, \text{ the ``output''}$$

It is reasonable that we should really only try to solve a PDE if we know that the above procedure is **well-posed**.

The strict definition of a well-posed PDE can depend on the context, and it's frequently easier to define non-well-posed ("ill-posed") problems.

Well-posedness

We can view the task of solving PDEs as a function from inputs to outputs:

$$\text{Inputs, e.g., } f,\ \kappa \xrightarrow{\text{Solution map}} \text{Solution } u, \text{ the ``output''}$$

It is reasonable that we should really only try to solve a PDE if we know that the above procedure is **well-posed**.

The strict definition of a well-posed PDE can depend on the context, and it's frequently easier to define non-well-posed ("ill-posed") problems.

For example, non-existence:

A PDE is ill-posed if for a given input, there is no solution u. E.g.,

$$\begin{aligned} u_t &= u_x, & x &\in (0, 2\pi),\ t > 0 \\ u(x,0) &= \sin x, & x &\in [0, 2\pi], \\ u(0,t) &= 0, & t &> 0 \\ u(2\pi, t) &= 0, & t &> 0. \end{aligned}$$

We can view the task of solving PDEs as a function from inputs to outputs:

$$\text{Inputs, e.g., } f,\ \kappa \xrightarrow{\text{Solution map}} \text{Solution } u, \text{ the "output"}$$

It is reasonable that we should really only try to solve a PDE if we know that the above procedure is **well-posed**.

The strict definition of a well-posed PDE can depend on the context, and it's frequently easier to define non-well-posed ("ill-posed") problems.

For example, non-uniqueness:

A PDE is ill-posed if for a given input, there are multiple solutions u. E.g.,

$$\begin{aligned} u'' &= \sin x, & x \in (0, 2\pi) \\ u'(0) &= 0, \\ u'(2\pi) &= 0. \end{aligned}$$

$u(x) = A\sin x + B\cos x$, $A = -1$, $B = 0$

$u(x) = -\sin x + x + 34$

Well-posedness

We can view the task of solving PDEs as a function from inputs to outputs:

$$\text{Inputs, e.g., } f,\ \kappa \xrightarrow{\text{Solution map}} \text{Solution } u, \text{ the “output”}$$

It is reasonable that we should really only try to solve a PDE if we know that the above procedure is **well-posed**.

The strict definition of a well-posed PDE can depend on the context, and it's frequently easier to define non-well-posed ("ill-posed") problems.

For example, ill-behaved properties:

A PDE is ill-posed if it depends on input parameters in "ill-behaved" ways. E.g.,

$$\begin{aligned} u_t &= -u_{xx}, & x &\in (0, 2\pi),\ t > 0 \\ u(x,0) &= f(x), & x &\in [0, 2\pi], \\ u(0,t) &= 0, & t &> 0 \\ u(2\pi,t) &= 0. & t &> 0 \end{aligned}$$

$f(x) = \frac{1}{n}\sin(nx), \ n \in \mathbb{N}$

$\lim_{n\uparrow\infty} \|f\| = 0$, but $\lim_{n\uparrow\infty} \|u(\cdot, \varepsilon)\| = \infty$

The solution u at arbitrarily small time $t > 0$ behaves uncontrollably with respect to infinitesimal perturbations of f.

Numerical methods: overall goals

"Most" PDEs *cannot* be analytically solved ☹

Our main strategy for recourse is to *approximate* the solution with a numerically computed one.

Numerical methods: overall goals

"Most" PDEs *cannot* be analytically solved ☹

Our main strategy for recourse is to *approximate* the solution with a numerically computed one.

We will *always* assume that a given ODE/PDE is well-posed.

If it's not, why bother to compute an approximate solution to a non-existent/non-unique/ill-behaved exact solution? (Although, mathematical/numerical methods for ill-posed problems are of significant intereset....)

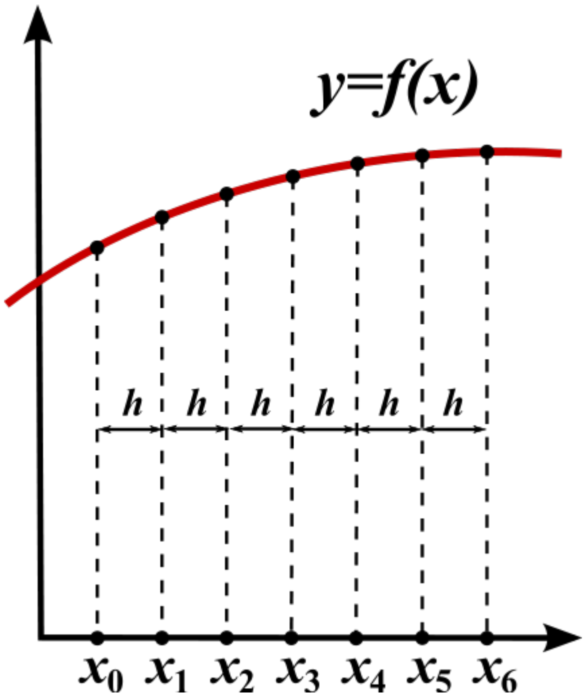
For numerical methods, we typically want the following things:

- Stability: The method does not "blow up" given reasonable inputs
- Accuracy: The solution computed by the method is "close" to the exact solution.
- Efficiency: The method does not take too much computational effort to compute a solution, and/or the memory and operation complexity required to compute a solution can be estimated.
- Simple: The method can be implemented and deployed with relative ease.

Summaries, necessarily simplistic

Finite difference methods:

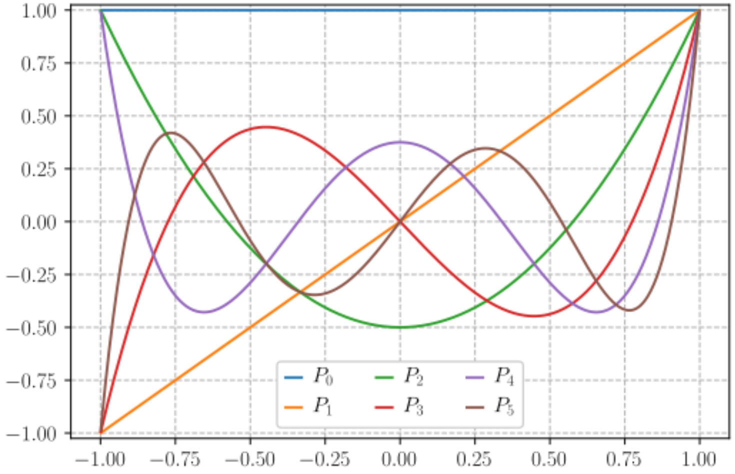
+ Easy, simple, transparent
– Relatively inflexible order of accuracy
– Difficult for complex geometries



Summaries, necessarily simplistic

Fourier/spectral methods:

+ Conceptually simple
+ "Infinite order" accuracy
- Very difficult for complex geometries
- Can suffer instability

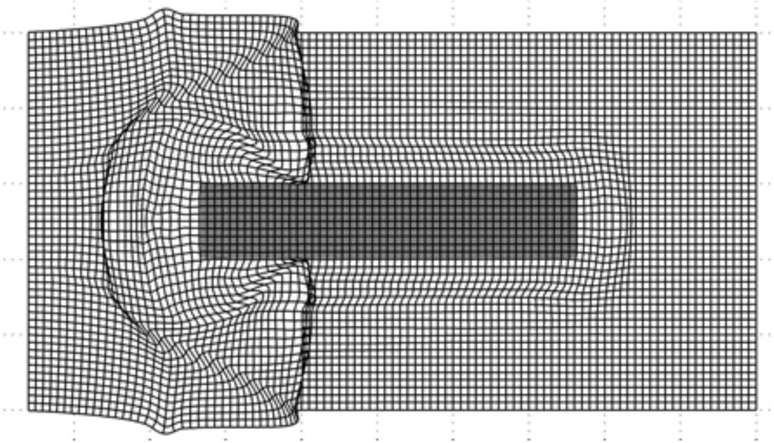


[Geek3 / CC-BY-SA-3.0]

Summaries, necessarily simplistic

Finite volume methods:

+ Solid mathematical theory
+ Can model non-smooth solutions
– Low order accuracy



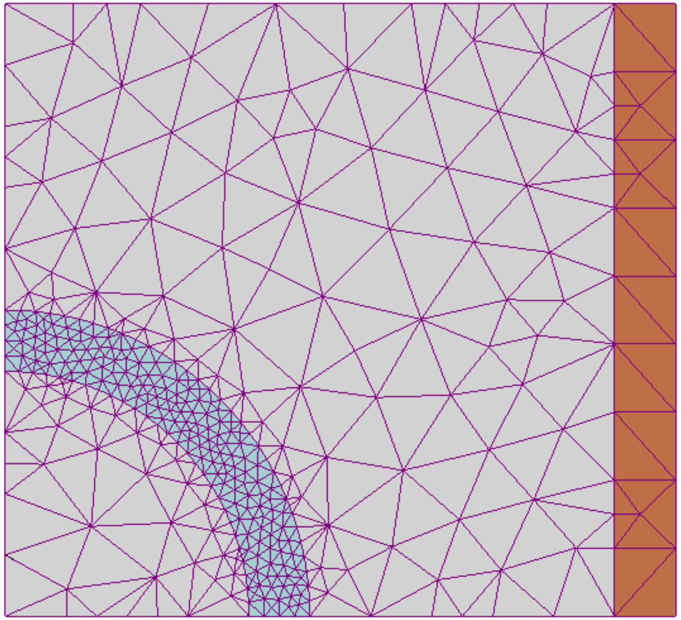
Randall J. LeVeque (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press. ISBN: 978-1-139-43418-8

(We won't cover these in this class.)

Summaries, necessarily simplistic

Finite element methods:

+ Solid mathematical theory
+ High-order and geometric flexibility
− Can involve technical mathematics
− Can be complicated to implement



[Zureks / CC-BY-SA-3.0]

(We won't cover these in this class.)

References I

- LeVeque, Randall J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press. ISBN: 978-1-139-43418-8.
- — (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM. ISBN: 978-0-89871-783-9.