Lecture 1

MTH572/MTH472
Numerical Methods for PDEs

Alfa Heryudono

UMass Dartmouth

Main references (quotes):
Trefethen: NumPDE, ATAP, Spectral Methods in MATLAB
Fornberg: PS Guide
Leveque: NumPDE
Driscoll: Learning MATLAB

Three great classes of partial differential equations:

- Elliptic: time-independent

  Ex: $-u_{xx} = f$ (Poisson equation), $u_{xx} = 0$ (Laplace equation). Application in electrostatics, static heat distribution, static distribution of tensions.

- Parabolic: time-dependent and diffusive

  Ex: $u_t = u_{xx}$, one-dimensional heat equation that describes diffusion of a quantity such as heat or salinity.

- Hyperbolic: time-dependent and wavelike; finite speed of propagation

  Ex: $u_t = u_x$, one dimensional first order wave equation that describes advection of a quantity $u(x,t)$ at a constant velocity -1. This equation is usually called advection (transport) equation.

Some PDEs sometimes are not that easy to fall into that trichotomy.

Schrödinger equation $u_t = i u_{xx}$

Looks like a parabolic equation at first glance. However, the equation is not diffusive but dispersive. Instead of decaying as time goes on, solutions tend to break up into oscillatory wave packets. (see quantum mechanics)

MTH362: study "pure" finite-difference models for

- linear,
- constant-coefficient equations on an infinite one-dimensional domain.

What you learned in that course is fundamental to an understanding of the more complicated problems.

Complications from practical problems

- Multiple space dimensions,
- System of equations,
- Geometry of domains,
- Boundary conditions,
- Variable coefficients,
- Nonlinearity.

NumPDE (collocation & Galerkin) roadmap:
Elvis era, Beatles era, Disco era.

1. 1950s: Finite-difference and/or volume methods.
2. 1960s: Finite-element methods.
3. 1970s:
   - Spectral and pseudospectral methods.
   - Spectral element methods.
   - Radial basis function methods.
   - Discontinuous Galerkin.

Popular approaches for numerically solving PDEs

- Classical way. Set-up regular grid in space and time. Compute approximate solutions on the grid by marching forward in time.
- Method of lines (MOL). Discretize the problem with respect to space, thereby generating a system of ODEs in t. Solve the system with popular ODE solvers. MOL offers flexibility in choosing method of choice for spatial discretization.

For this course (MTH572/472), we will be concentrating mostly on finite-difference methods for spatial discretizations.

Note: Not all finite difference methods can be analyzed with method of lines, but many can. However, this is becoming a point of view that is increasingly important for difficult problems.

Spatial discretization operators

Five ways of looking at finite finite-difference formulas

1. Discrete approximation to derivatives (Taylor expansions in MTH112).
2. Derivatives of polynomial interpolants (Lagrange interpolating polynomials MTH361).
3. Convolution filters.
4. Toeplitz matrices.
5. Fourier multipliers.

Note: Some of those "ways" can be done naturally in the case of infinite one-dimensional domains. Additionally, if the function to be differentiated is smooth and periodic, the derivation can be simplified even further.

Continuous - Discrete Analogy

	Continuous	Discrete
Domain	$[a, b]$	$a = x_1 < x_2 < \cdots < x_n = b$
Function	$u(x)$	1-D vector $\underline{u} \approx \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$
Diff operator	$\dfrac{\partial}{\partial x}$	"differentiation" matrix $D = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix}$
Diff operation	$\dfrac{\partial u}{\partial x}$	matrix-vector product $\vec{u}' \approx \begin{bmatrix} u_1' \\ \vdots \\ u_n' \end{bmatrix} = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$

Note: Subindex 0 is heavily used in analysis (typically when you deal with polynomials). However, in MATLAB, index starts from 1. So, make sure to check the size of your problem.

Differentiation Matrices

Goal: To compute entries (d_{ij}) of differentiation matrix D. To make life even simpler, the interval $[a, b]$ is discretized with n equal-spacing of size h. Hence, $x_0 = a$, $x_1 = a + h$, $x_2 = a + 2h, \ldots, x_j = a + jh, \ldots, x_n = a + nh = b$, where $h = (b - a)/n$. Note that the total number of points is $n + 1$.

Taylor expansion around $x = x_j$

$$u(x) \approx u(x_j) + \frac{u'(x_j)}{1!}(x - x_j) + \frac{u''(x_j)}{2!}(x - x_j)^2 + \frac{u'''(x_j)}{3!}(x - x_j)^3 + \ldots$$

Evaluating the expansion at the neighboring point x_{j+1} gives

$$u(x_{j+1}) \approx u(x_j) + \frac{u'(x_j)}{1!}h + \frac{u''(x_j)}{2!}h^2 + \frac{u'''(x_j)}{3!}h^3 + \ldots$$

where $u'(x_j)$ can be obtained (assuming that $u(x)$ is smooth enough) as

$$u_j' \approx \frac{1}{h}u_{j+1} - \frac{1}{h}u_j + \mathcal{O}(h),$$

where u_j stands for $u(x_j)$.

Similarly, using the left neighboring point x_{j-1} gives

$$u'_j \approx -\frac{1}{h}u_{j-1} + \frac{1}{h}u_j + \mathcal{O}(h),$$

- The constants $1/h, -1/h$ in front of function values are called finite difference "weights".
- The stencil-size of this finite difference is 2 since the value of derivative u'_j can be computed using function values at 2 points x_j or x_{j+1} (or x_j or x_{j-1}).
- the order of accuracy is one (the power of h in $\mathcal{O}(h)$).

Assemble everything together in a matrix style

$$\vec{u}' \approx \begin{bmatrix} u'_0 \\ u'_1 \\ u_2 \\ \vdots \\ u'_{n-1} \\ u'_n \end{bmatrix} = D\underline{u} = \frac{1}{h}\begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & -1 & 1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}$$

Things to do in class

1. With $n = 10$, use the command `spdiags` in MATLAB to create the differentiation matrix on the previous slide.
2. Test it to find the derivative of $u(x) = e^{-x^2}$ on the interval $[-1, 1]$. Compute $\|\cdot\|_\infty = |\underline{u} - \underline{u}_{exact}|_\infty$ to measure the error of numerical derivative with respect the exact derivative.
3. Using different values of $n = 10, 100, 1000, \cdots$, redo problem 2 and plot the $\|\cdot\|_\infty$ vs h in log scale. Do you observe 1^{st}-order convergence (i.e. $\mathcal{O}(h)$ or $\mathcal{O}(n^{-1})$) ?
4. What will happen if you use the matrix on the periodic function $u(x) = \sin(x)$ on the interval $[0, 2\pi]$ or $u(x) = \sin(\pi x)$ on the interval $[-1, 1]$? How do you modify the matrix to deal with periodicity ?

Tips: Equally-spaced points (nodes) can be generated with the commands `linspace` and `logspace` (for log scale).