CS 450: Numerical Anlaysis

Lecture 27

Chapter 11 Partial Differential Equations

Numerical Methods for Partial Differential Equations

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

April 25, 2018

Partial Differential Equations

- Partial differential equations (PDEs) are equations describe physical laws and other continuous phenomena:
  - *They contain partial derivatives in multiple variables.*
  - *Examples include: electromagnetism, fluid flow, quantum mechanics, and general relativity.*
- A simple PDE is the advection equation, which describes basic phenomena in fluid flow:

$$u_t = -a(t, x)u_x$$

where $u_t = \partial u/\partial t$ *and* $u_x = \partial u/\partial x$*. Generally, we impose an initial condition with respect to* t*, i.e.,* $u(0, x) = u_0(x)$*. When* $a(t, x) = c$ *this is the Cauchy problem with solution*

$$u(t, x) = u_0(x - ct).$$

Properties of PDEs

- A *characteristic* of a PDE is a level curve in the solution:

  For the Cauchy form of the advection equation, a characteristic $x(t)$ *satisfies* $u(t, x(t)) = $ *const. One of their uses is identifying where boundary conditions must be satisfied, e.g. for the Cauchy equation, it tells us whether boundary conditions are needed on the left or the right (in terms of* x*) of the domain. More generally, characteristic curves describe curves in the solution field* $u(t, x)$ *that correspond to ODEs e.g. for the advection equation,*

  $$\frac{\partial x(t)}{\partial t} = a(t, x(t)) \text{ with initial condition } x(0) = x_0.$$

- The order of a PDE is the highest-order of any partial derivative appearing in the PDE:

  The advection equation is a first order ODE. ODEs containing e.g. u_{xy} *or* u_{tt} *are second order.*

Types of PDEs

- Some of the most important PDEs are second order:
  - *Heat equation (diffusion),* $u_t = u_{xx}$
  - *Wave equation (oscillation),* $u_{tt} = u_{xx}$
  - *Laplace equation (steady-state),* $u_{xx} + u_{yy} = 0$

  Any PDE of the form

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

  behaves (has the character) of one of the above equations.
- The *discriminant* determines the canonical form of second-order PDEs:

  The discriminant is $r = b^2 - 4ac$ *and linear PDEs are classified as*

  $$\begin{cases} r > 0 & : \text{hyperbolic, wave-equation-like} \\ r = 0 & : \text{parabolic, heat-equation-like} \\ r < 0 & : \text{elliptic, Laplace-equation-like} \end{cases}$$

  When coefficients are varying, a PDE may exhibit different behavior in different parts of the domain.

Method of Lines

- Semidiscrete methods obtain an approximation to the PDE by solving a system of ODEs, e.g. consider heat equation

$$u_t = cu_{xx} \text{ on } 0 \leq x \leq 1, \quad u(0,x) = f(x), u(t,0) = u(t,1) = 0$$

We discretize over x and use finite differences to approximate

$$u_{xx} \approx \frac{u(t, x_{i+1}) - 2u(t, x_i) + u(t, x_{i-1})}{(\Delta x)^2}.$$

This approximation yields a system of ODEs

$$y_i'(t) = \frac{c}{(\Delta x)^2}\Big(y_{i+1}(t) - 2y_i(t) + y_{i-1}(t)\Big)$$

which we can write as $\boldsymbol{y}'(t) = \boldsymbol{A}\boldsymbol{y}(t)$, where $\boldsymbol{A}$ is tridiagonal.

- This *method of lines* often yields a stiff ODE:

  The eigenvalues of $\boldsymbol{A}$ range from near 0 to $-4c/(\Delta x)^2$, which means the ODE is stiff for small Δx. Intuitively, strong concentrations of heat dissipate very rapidly, while overall heat dissipates slowly.

Semidiscrete Collocation

- Instead of finite-differences, we can express $u(t, x)$ in a spatial basis:

$$u(t, x) \approx v(t, x, \boldsymbol{\alpha}(t)) = \sum_{j=1}^{n} \alpha_j \phi_j(x)$$

Collocation methods then ensure the approximation is exact on $x_1, \ldots, x_n$*, yielding matrix equations.*

- For the heat equation $u_t = cu_{xx}$, we obtain an ODE:

$$\sum_{j=1}^{n} \frac{\partial \alpha_j}{\partial t}(t) \underbrace{\phi_j(x_i)}_{m_{ij}} = c \sum_{j=1}^{n} \alpha_j(t) \underbrace{\frac{\partial^2 \phi_j}{\partial x^2}(x_i)}_{n_{ij}}$$

written in matrix form,

$$\boldsymbol{\alpha}'(t) = c\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{\alpha}(t)$$

Fully Discrete Methods

- Generally, both time and space dimensions are discretized, for example using finite differences:

  Lets again consider the heat equation $u_t = cu_{xx}$ *and discretize so that* $u_i^{(k)} \approx u(t_k, x_i)$,

$$\frac{u_i^{(k+1)} - u_i^{(k)}}{\Delta t} = c\frac{u_{i+1}^{(k)} - 2u_i^{(k)} + u_{i-1}^{(k)}}{(\Delta x)^2}$$

  which yields an iterative scheme defined by 3-pt stencil

$$u_i^{(k+1)} = u_i^{(k)} + c\Delta t\frac{u_{i+1}^{(k)} - 2u_i^{(k)} + u_{i-1}^{(k)}}{(\Delta x)^2}.$$

  The same scheme can be derived by applying Euler's method to the ODE given by the method of lines.

Implicit Fully Discrete Methods

- When using Euler's method for the heat equation, to stay in stability region, require

$$\Delta t = O\Big((\Delta x)^2\Big)$$

Step-size restriction on stability can be circumvented by use of implicit time-stepper, such as backward Euler,

$$u_i^{(k+1)} = u_i^{(k)} + c\Delta t \frac{u_{i+1}^{(k+1)} - 2u_i^{(k+1)} + u_{i-1}^{(k+1)}}{(\Delta x)^2}.$$

This scheme requires for a tridiagonal matrix system to be solved at each time-step, but obtains unconditional stability, albeit only first-order accuracy. Using the trapezoid method to solve the ODE we obtain the second-order Crank-Nicolson method,

$$u_i^{(k+1)} = u_i^{(k)} + c\Delta t \frac{u_{i+1}^{(k+1)} - 2u_i^{(k+1)} + u_{i-1}^{(k+1)} + u_{i+1}^{(k)} - 2u_i^{(k)} + u_{i-1}^{(k)}}{2(\Delta x)^2}.$$

Convergence and Stability

- *Lax Equivalence Theorem*: consistency + stability = convergence

  Consistency means that the local truncation error goes to zero, and is easy to verify by Taylor expansions. Stability implies that the approximate solution at any time t *must remain bounded. Together these conditions are necessary and sufficient for convergence.*

- Stability can be ascertained by spectral or Fourier analysis:

  In the method of lines, we saw that the eigenvalues of the resulting ODE define the stability region. Fourier analysis decomposes the solution into a sum of harmonic functions and considers the behavior of their amplitude.

CFL Condition

- The domain of dependence of a PDE for a given point (t, x) is the portion of the problem domain influencing this point through the PDE:

  Generally determined by characteristics of PDE. For a stencil method, depends on the set of mesh-points influencing the mesh point at (t, x).

- The Courant, Friedrichs, and Lewy (CFL) condition states that a *necessary* condition for an explicit finite-differencing scheme to be stable for a hyperbolic PDE is that the domain of the dependence of the PDE be contained in the domain of dependence of the scheme:

  Intuitively, we can then achieve stability by choosing a sufficiently large grid spacing h*, or including more mesh points in our stencil.*

Time-Independent PDEs

- We now turn our focus to time-independent PDEs as exemplified by the *Helmholtz equation*:

$$u_{xx} + u_{yy} + \lambda u = f(x, y)$$

  - $\lambda = 0$ *yields the Poisson equation*
  - $\lambda = 0$ *and* $f = 0$ *yields the Laplace equation*
  - *Boundary conditions (e.g. Dirichlet or Neumann or mixed) on surface of domain*

- We discretize as before, but no longer perform time stepping:

  For example given a domain $[0, 1]^2$*, we can tile it using* $n \times n$ *mesh points, and setup finite-difference equations or boundary-condition equations at each point.*

Finite-Differencing for Poisson

- Consider the Poisson equation with equispaced mesh-points on $[0, 1]$:
  If $\boldsymbol{u}$ is a vector containing the mesh-points, we have that

  $$\mathcal{D}_x\boldsymbol{u} + \mathcal{D}_y\boldsymbol{u} = \boldsymbol{b} \text{ where } b_i = f(u_i)$$

  where $\mathcal{D}_x$ and $\mathcal{D}_y$ are finite-difference operators along x and y dimensions, respectively. Given a differencing matrix $\boldsymbol{D}$ (e.g. tridiagonal with $1, -2, 1$), we obtain the matrix equation,

  $$(\boldsymbol{I} \otimes \boldsymbol{D} + \boldsymbol{D} \otimes \boldsymbol{I})\boldsymbol{u} = \boldsymbol{b}$$

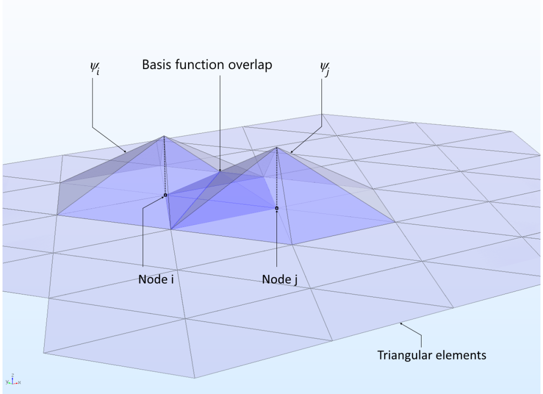
  where the Kronecker product is defined as

  $$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots \\ a_{21}\boldsymbol{B} & \ddots & \\ \vdots & & \end{bmatrix},$$

  and the elements of $\boldsymbol{b}$ contain the mesh elements in column-major (or row-major in this example) order.

Multidimensional Finite Elements

- There are many ways to define localized basis functions, for example in the 2D FEM method[1]:



We partition the domain into triangles (elements) and define linear basis functions that are 1 at the intersection of three or more elements (nodes).

[1] Source: Comsol Multiphysics Cyclopedia `https://www.comsol.com/multiphysics/finite-element-method`