Discretization of PDEs and Tools for the Parallel Solution of the Resulting Systems

Stan Tomov

Innovative Computing Laboratory
Computer Science Department
The University of Tennessee

Wednesday March 29, 2017
Topics

Projection in Scientific Computing
(lecture 1)

Sparse matrices, parallel implementations
(lecture 3)

PDEs, Numerical solution, Tools, etc.
(lecture 2)

Iterative Methods
(lectures 4 and 5)
Outline

- **Part I**
  Partial Differential Equations

- **Part II**
  Mesh Generation and Load Balancing

- **Part III**
  Tools for Numerical Solution of PDEs
Part I

Partial Differential Equations
Mathematical Modeling

Mathematical Model:
- a representation of the essential aspects of an existing system which presents knowledge of that system in usable form (Eykhoff, 1974)

Mathematical Modeling:
Real world \hspace{1cm} Model

Navier-Stokes equations:

\[ \begin{align*}
\nabla \cdot u &= 0 \\
\frac{\partial u}{\partial t} &= -(u \cdot \nabla)u - \frac{1}{\rho} \nabla p + \nu \nabla^2 u + f \\
B.C. & \hspace{1cm} etc.
\end{align*} \]
We are interested in models that are

- **Dynamic**
  i.e. account for changes in time

- **Heterogeneous**
  i.e. account for heterogeneous systems

Typically represented with

- **Partial Differential Equations**
How can we model for e.g. Heat Transfer?

- Heat
  * a form of energy (thermal)

- Heat Conduction
  * transfer of thermal energy from a region of higher temperature to a region of lower temperature

- Some notations

  $Q$: amount of heat
  $k$: material conductivity
  $T$: temperature
  $A$: area of cross-section
The Law of Heat Conduction

\[ \frac{\Delta Q}{\Delta t} = k \ A \ \frac{\Delta T}{\Delta x} \]

Change of heat is proportional to the gradient of the temperature and the area \( A \) of the cross-section.

- \( Q \): amount of heat
- \( k \): material conductivity
- \( T \): temperature
- \( A \): area of cross-section
Consider 1-D heat transfer in a thin wire

- so thin that $T$ is piecewise constant along the slides, i.e. $T_0(t)$, $T_1(t)$, $T_2(t)$, etc.
- ideally insulated

Let us write a balance for the temperature at $T_1$ for time $t + \Delta t$

$$T_1(t + \Delta t) = ?$$
Heat Transfer

\[ T_1(t + \triangle t) \approx T_1(t) + k\triangle t \frac{(T_2(t) - T_1(t))}{(\triangle x)^2} + k\triangle t \frac{(T_0(t) - T_1(t))}{(\triangle x)^2} \]

\[ = T_1(t) + k\triangle t \frac{T_2(t) - 2T_1(t) + T_0(t)}{(\triangle x)^2} \]

Take \( \lim_{\triangle x, \triangle t \to 0} \)

\[ \Rightarrow \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \quad (Exercise) \]
Extend to 2-D and put a source term $f$ to easily get

$$\frac{\partial T}{\partial t} = k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + f \equiv k \triangle T + f$$

Known as the Heat equation
Other Important PDEs

- **Poisson equation (elliptic)**
  \[ \Delta u = f \]

- **Heat equation (parabolic)**
  \[ \frac{\partial T}{\partial t} = k \Delta T + f \]

- **Wave equation (hyperbolic)**
  \[ \frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = \Delta u + f \]
Classification of PDEs

For a general second-order PDE in 2 variables:

\[ Au_{xx} + Bu_{xy} + Cu_{yy} + \cdots = 0 \]

**Elliptic:**
- if \( B^2 - 4AC < 0 \)
- process in equilibrium (no time dependence)
- easy to discretize but challenging to solve

**Parabolic:**
- if \( B^2 - 4AC = 0 \)
- processes evolving toward steady state

**Hyperbolic:**
- if \( B^2 - 4AC > 0 \)
- not evolving toward steady state
- difficult to discretize (support discontinuities) but easy to solve in characteristic form
How do we solve them?

Numerical solution approaches:

- Finite difference method
- Finite element method
- Finite volume method
- Boundary element method
- use finite differences to approximate differential operators
- one of the simplest and extensively used method in solving PDEs
- the error, called truncation error, is due to finite approximation of the Taylor series of the differential operator
A Finite Difference Method Example

Consider the 2-D Poisson equation:

\[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \]

The idea, first in 1-D:

- Use Taylor series to approximate \( \frac{d^2 u}{dx^2}(x) \) with \( u(x), u(x+h), u(x-h) \)

\[
\begin{align*}
  u(x + h) &= u(x) + h \frac{du}{dx}(x) + \frac{h^2}{2} \frac{d^2 u}{dx^2}(x) + \frac{h^3}{3!} \frac{d^3 u}{dx^3}(x) + O(h^4) \\
  u(x - h) &= u(x) - h \frac{du}{dx}(x) + \frac{h^2}{2} \frac{d^2 u}{dx^2}(x) - \frac{h^3}{3!} \frac{d^3 u}{dx^3}(x) + O(h^4)
\end{align*}
\]

\[ \Rightarrow \frac{d^2 u}{dx^2}(x) = \frac{1}{h^2} (u(x + h) + u(x - h) - 2u(x)) + O(h^2) \]
Similarly in 2-D

- Use Taylor series to approximate $\Delta u(x, y)$ with $u(x, y), u(x + h, y), u(x - h, y), u(x, y + h), u(x, y - h)$.

\[
\begin{align*}
  u(x + h, y) &= u(x, y) + h \frac{\partial u}{\partial x}(x, y) + \frac{h^2}{2} \frac{\partial^2 u}{\partial x^2}(x, y) + \frac{h^3}{3!} \frac{\partial^3 u}{\partial x^3}(x, y) + \mathcal{O}(h^4) \\
  u(x - h, y) &= u(x, y) - h \frac{\partial u}{\partial x}(x, y) + \frac{h^2}{2} \frac{\partial^2 u}{\partial x^2}(x, y) - \frac{h^3}{3!} \frac{\partial^3 u}{\partial x^3}(x, y) + \mathcal{O}(h^4) \\
  u(x, y + h) &= u(x, y) + h \frac{\partial u}{\partial y}(x, y) + \frac{h^2}{2} \frac{\partial^2 u}{\partial y^2}(x, y) + \frac{h^3}{3!} \frac{\partial^3 u}{\partial y^3}(x, y) + \mathcal{O}(h^4) \\
  u(x, y - h) &= u(x, y) - h \frac{\partial u}{\partial y}(x, y) + \frac{h^2}{2} \frac{\partial^2 u}{\partial y^2}(x, y) - \frac{h^3}{3!} \frac{\partial^3 u}{\partial y^3}(x, y) + \mathcal{O}(h^4)
\end{align*}
\]

$\Rightarrow \Delta u(x, y) = \frac{1}{h^2} (u(x + h, y) + u(x - h, y) + u(x, y + h) + u(x, y - h) - 4u(x)) + \mathcal{O}(h^2)$
A Finite Difference Method Example

Consider the 1-D equation:

\[
\frac{d^2 u}{dx^2}(x) = f(x), \quad \text{for } x \in (0, 1)
\]

and the Dirichlet boundary condition

\[ u(0) = u(1) = 0 \]

The interval \([0, 1]\) is discretized uniformly with \(n + 2\) points

\[
\begin{array}{cccccc}
\hline
x_0 & x_1 & \cdots & x_n & x_{n+1} \\
\hline
h & h & \cdots & h & h \\
\end{array}
\]

At any point \(x_i\) we are looking for \(u_i\), an approximation of the exact solution \(u(x_i)\), using the approximation

\[-u_{i-1} + 2u_i - u_{i+1} = h^2f_i,\]

and the fact that \(u_0 = u_{n+1} = 0\),

we obtain a linear system of the form

\[ Ax = b \]

where \(b = (f_i)_{i=1,n}\) and \(x = (u_i)_{i=1,n}\) and

\[
A = \frac{1}{h^2}
\begin{pmatrix}
2 & -1 & & & \\
-1 & 2 & -1 & & \\
& -1 & 2 & -1 & \\
& & \ddots & \ddots & \\
& & & -1 & 2 & -1 \\
& & & & -1 & 2
\end{pmatrix}
\]

(slide used material from Julien Langou’s presentation)
Consider the 2-D Poisson equation:

$$\Delta u = f$$

and the Dirichlet boundary condition

$$u(x, y) = 0 \text{ for } (x, y) \in \partial \Omega$$

The interval $[0, 1] \times [0, 1]$ is discretized uniformly with $(n + 2) \times (n + 2)$ points.

\[
A = \frac{1}{h^2}
\begin{pmatrix}
B & -I & & & \\
-I & B & -1 & & \\
& -I & B & -1 & \\
& & & \ddots & \ddots \\
& & & -I & B & -I \\
& & & & -I & B
\end{pmatrix}
\]

where

\[
B =
\begin{pmatrix}
4 & -1 & & & \\
-1 & 4 & -1 & & \\
& -1 & 4 & -1 & \\
& & & \ddots & \ddots \\
& & & -1 & 4 & -1 \\
& & & & -1 & 4
\end{pmatrix}
\]

(slide used material from Julien Langou’s presentation)
Finite Element Method

- Remember the slides from the previous lecture
  

Main pluses/minuses of FEM vs FDM

- FEM can handle complex geometries
- FDM is easy to implement
Consider the 1-D Dirichlet problem:

\[ u''(x) = f(x), \quad \text{for } x \in (0,1) \]

and the Dirichlet boundary condition

\[ u(0) = u(1) = 0 \]

**Weak or Variational formulation:**

- Multiply (1) by smooth \( v \) and integrate over \((0,1)\)

\[
\int_0^1 f(x)v(x)dx = \int_0^1 u''(x)v(x)dx
\]

- Integrate by parts the above RHS

\[
\int_0^1 u''(x)v(x)dx = u'(x)v(x)|_0^1 - \int_0^1 u'(x)v'(x)dx
\]

\[ = -\int_0^1 u'(x)v'(x)dx \equiv -a(u,v) \]

- Variational formulation: Find \( u \in H^1_0(0,1) \) such that

\[
\int_0^1 f(x)v(x)dx = -a(u,v) \text{ for } \forall v \in H^1_0(0,1)
\]
A Finite Element Method Example

Discretization (Galerkin FE problem):
- Replace $H^1_0(0, 1)$ with finite dimensional subspace $V$

Shown is a 4 dimensional space $V$ (basis in blue) and a linear combination (in red)

$$v_k(x) = \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}} & \text{if } x \in [x_{k-1}, x_k], \\ \frac{x-x_k}{x_{k+1}-x_k} & \text{if } x \in [x_k, x_{k+1}], \\ 0 & \text{otherwise,} \end{cases}$$

What is the matrix form of the problem (Exercise)

$x_0=0$ $x_1$ $x_2$ $x_3$ $x_4$ $x_5=1$
Part II

Mesh Generation and Load Balancing

Part III

Tools for Numerical Solution of PDEs
Parallel PDE Computations

Challenges:

- Software Complexity
- Data Distribution and Access
- Portability, Algorithms, and Data Redistribution

Read more in Chapter 21
Software for PDEs

There is software; to mention a few packages:

- **Overture**
  OO framework for PDEs in complex moving geometry

- **PARASOL**
  Parallel, sparse matrix solvers; in Fortran 90

- **SAMRAI**
  OO framework for parallel AMR applications

- **Hypre**
  Large sparse linear solvers and preconditioners

- **PETSc**
  Tools for numerical solution of PDEs

- **FFTW**
  parallel FFT routines

- **Diffpack**
  OO framework for solving PDEs

- **Doug**
  FEM for elliptic PDEs

- **POOMA**
  OO framework for HP applications

- **UG**
  PDEs on unstructured grids using multigrid

**PETSc**: Portable, Extensible Toolkit for Scientific computation

- for large-scale sparse systems
- facilitate extensibility
- provides interface to external packages, e.g. BlockSolve95, ESSL, Matlab, ParMeTis, PVODE, and SPAI.
- programmed in C, usable from Fortran and C++
- uses MPI for all parallel communication
  - in a distributed-memory model
  - user do communication on level higher than MPI
- Computation and communication kernels: MPI, MPI-IO, BLAS, LAPACK
PETSc’s Main Numerical Components

**Nonlinear Solvers**
- Newton-based Methods
- Line Search
- Trust Region
- Other

**Time Steppers**
- Euler
- Backward Euler
- Pseudo Time Stepping
- Other

**Krylov Subspace Methods**
- GMRES
- CG
- CGS
- Bi-CG-STAB
- TFQMR
- Richardson
- Chebychev
- Other

**Preconditioners**
- Additive Schwarz
- Block Jacobi
- Jacobi
- ILU
- ICC
- LU (Sequential only)
- Others

**Matrices**
- Compressed Sparse Row (AIJ)
- Blocked Compressed Sparse Row (BAIJ)
- Block Diagonal (BDIAG)
- Dense
- Other

**Vectors**

**Index Sets**
- Indices
- Block Indices
- Stride
- Other

more info at: [http://acts.nersc.gov/petsc/](http://acts.nersc.gov/petsc/)
Learning Goals

A brief overview of Numerical PDEs and related issues

- Mathematical modeling
- PDEs for describing changes in physical processes
- More specific discretization examples
  - Finite Differences (natural)
  - FEM
    - reinforce the idea and application of Petrov-Galerkin conditions
- Issues related to mesh generation and load balancing and importance in HPC
  - Adaptive methods
- Software