COMPUTATIONAL MODELING WITH PARTIAL DIFFERENTIAL EQUATIONS (PDEs)

Chad Westphal
westphac@wabash.edu

Department of Math and Computer Science
Wabash College
### Physical Phenomena described by PDEs

<table>
<thead>
<tr>
<th>Porous Media</th>
<th>Fluid Mechanics</th>
<th>Elasticity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrostatics</td>
<td>Dynamical Systems</td>
<td>Quantum mechanics</td>
</tr>
<tr>
<td>Advection</td>
<td>Combustion</td>
<td>Atmospheric Sciences</td>
</tr>
<tr>
<td>Plasma Physics</td>
<td>MHD</td>
<td>Multiphase Flow</td>
</tr>
<tr>
<td>Coupled Systems</td>
<td>Seismic Modeling</td>
<td>Astrophysics</td>
</tr>
<tr>
<td>Finance</td>
<td>Ocean Modeling</td>
<td>Biological Modeling</td>
</tr>
</tbody>
</table>

Who else is interested in this?

- **Industry:** Engineering (Auto, Aerospace, etc.), Financial, Defense
- **National Labs / Universities:** Physicists, Chemists, Biologists, Engineers, etc.
### Computational Modeling Process

<table>
<thead>
<tr>
<th>Task</th>
<th>Those Involved</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observe and Measure Data</td>
<td>Engineers, Physicists, etc.</td>
<td>Simplicity</td>
</tr>
<tr>
<td>Math Model</td>
<td>Physicists, Mathematicians</td>
<td>Well-posedness</td>
</tr>
<tr>
<td>Discrete Model</td>
<td>Mathematicians</td>
<td>Accuracy, conservation</td>
</tr>
<tr>
<td>Numerical Solution</td>
<td>Mathematicians, Computer Scientists</td>
<td>Computational cost (storage, time, etc.)</td>
</tr>
<tr>
<td>Visualize Data</td>
<td>Computer Scientists, Engineers</td>
<td>Data Structures, architecture</td>
</tr>
<tr>
<td>Interpret Results</td>
<td>Engineers, Physicists, etc.</td>
<td>Calibration</td>
</tr>
</tbody>
</table>
## Computational Modeling Process

<table>
<thead>
<tr>
<th>Task</th>
<th>Those Involved</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observe and Measure Data</td>
<td>Engineers, Physicists, etc.</td>
<td>Simplicity</td>
</tr>
<tr>
<td>Math Model</td>
<td>Physicists, Mathematicians</td>
<td>Well-posedness</td>
</tr>
<tr>
<td>Discrete Model</td>
<td>Mathematicians</td>
<td>Accuracy, conservation</td>
</tr>
<tr>
<td>Numerical Solution</td>
<td>Mathematicians, Computer Scientists</td>
<td>Computational cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(storage, time, etc.)</td>
</tr>
<tr>
<td>Visualize Data</td>
<td>Computer Scientists, Engineers</td>
<td>Data Structures, architecture</td>
</tr>
<tr>
<td>Interpret Results</td>
<td>Engineers, Physicists, etc.</td>
<td>Calibration</td>
</tr>
</tbody>
</table>
What are PDEs?

Partial Differential Equations (PDEs) are equations involving partial derivatives of functions that depend on more than one variable.

May include: systems (many unknowns), time dependent, nonlinear or stochastic behavior, higher dimensions, etc.

Example: By conservation of energy and Fourier’s law of heat conduction we can derive

\[-\Delta u = -\partial_{xx}^2 u - \partial_{yy}^2 u = f\]

\(u = u(x, y)\) the temperature at any point in a 2-D object

\(f = f(x, y)\) internal sources of energy
We want to find $u(x, y)$, the temperature in $\Omega$, from the relation above
Sometimes it is possible to find a solution to these PDEs “by hand.” In most practical applications it becomes too difficult.

**Idea:** Replace the continuous domain with a discrete domain (grid):

\[
\begin{align*}
  &i = 0, 1, 2, \ldots, n_x \\
  &j = 0, 1, 2, \ldots, n_y \\
  &\Delta x = h_x = \frac{1}{n_x} \\
  &\Delta y = h_y = \frac{1}{n_y}
\end{align*}
\]

Now look for the solution only at the nodes: \( u(x, y) \rightarrow u_{i,j} = u(h_x i, h_y j) \)
Discretization: Replace the derivatives by finite differences (via truncated Taylor series)

\[ \partial_x u_{i,j} \approx \frac{u_{i,j} - u_{i-1,j}}{h_x} \]

\[ \partial_x u_{i+1,j} \approx \frac{u_{i+1,j} - u_{i,j}}{h_x} \]

\[ \partial_{xx} u_{i,j} \approx \frac{\partial_x u_{i+1,j} - \partial_x u_{i,j}}{h_x} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} \]
Discretization: Replace the derivatives by finite differences (via truncated Taylor series)

\[ \partial_x u_{i,j} \approx \frac{u_{i,j} - u_{i-1,j}}{h_x} \]

\[ \partial_x u_{i+1,j} \approx \frac{u_{i+1,j} - u_{i,j}}{h_x} \]

\[ \partial_{xx} u_{i,j} \approx \frac{\partial_x u_{i+1,j} - \partial_x u_{i,j}}{h_x} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} \]
**Discretization:** Replace the derivatives by finite differences (via truncated Taylor series)

\[
\partial_x u_{i,j} \approx \frac{u_{i,j} - u_{i-1,j}}{h_x}
\]

\[
\partial_x u_{i+1,j} \approx \frac{u_{i+1,j} - u_{i,j}}{h_x}
\]

\[
\partial_{xx} u_{i,j} \approx \frac{\partial_x u_{i+1,j} - \partial_x u_{i,j}}{h_x} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}
\]
Our PDE now becomes a system of linear equations

\[- \left( \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} \right) = f(h_x i, h_y j)\]

\[i = 0, 1, 2, \ldots, n_x \quad j = 0, 1, 2, \ldots, n_y\]
This is now a linear system of equations, $Ax = b$

- $A$ is a matrix of size $N \times N$, $N = (n_x + 1)(n_y + 1)$
- $x$ is an $N \times 1$ vector of unknowns (i.e., $u_{i,j}$ values)
- $b$ is an $N \times 1$ known vector (i.e., $f(h_{xi}, h_{yj})$ values)

We now just compute $x = A^{-1}b$, right?

$A$ is generally large, sparse and poorly conditioned...
\[ A = \frac{1}{h^2} \begin{pmatrix} 4 & -1 & \cdots & -1 \\ -1 & 4 & -1 & \cdots & -1 \\ -1 & 4 & -1 & \cdots & -1 \\ -1 & \cdots & -1 & 4 & -1 & \cdots & -1 \\ -1 & \cdots & -1 & 4 & -1 \\ -1 & \cdots & -1 & 4 \\ -1 & \cdots & -1 & 4 \end{pmatrix} \]
\[ A = \frac{1}{h^2} \begin{pmatrix}
4 & -1 & \cdots & -1 \\
-1 & 4 & -1 & \cdots & -1 \\
-1 & 4 & -1 & \cdots & -1 \\
-1 & \cdots & -1 & 4 & -1 & \cdots & -1 \\
-1 & \cdots & -1 & 4 & -1 \\
-1 & \cdots & -1 & 4 \\
\end{pmatrix} \]
\[ A = \frac{1}{h^2} \begin{pmatrix} 
4 & -1 & \ldots & -1 \\
-1 & 4 & -1 & \ldots & -1 \\
-1 & 4 & -1 & \ldots & -1 \\
-1 & \ldots & -1 & 4 & -1 & \ldots & -1 \\
-1 & \ldots & -1 & 4 & -1 \\
-1 & \ldots & -1 & 4 \\
-1 & \ldots & -1 & 4 \\
\end{pmatrix} \]
A is invertible, but has a large "near null space"...

the vector

\[ \xi = \begin{pmatrix}
1 \\
1 \\
1 \\
1 \\
\end{pmatrix} \]

almost produces \( A\xi = 0 \). These low energy modes are hard to resolve.
Computed Solution

insulated boundary

prescribed temperature

prescribed heat flux

$\mathbf{f}$
Example:

Steady state heat equation on unit square (2-D)

<table>
<thead>
<tr>
<th>grid points per side</th>
<th>$n$</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>unknowns (N)</td>
<td>$n^2$</td>
<td>100</td>
<td>$10^4$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>Entries in Matrix</td>
<td>$n^4$</td>
<td>10000</td>
<td>$10^8$</td>
<td>$10^{12}$</td>
</tr>
</tbody>
</table>
Example:
Steady state heat equation on unit cube (3-D)

<table>
<thead>
<tr>
<th>grid points per side</th>
<th>$n$</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>unknowns (N)</td>
<td>$n^3$</td>
<td>1000</td>
<td>$10^6$</td>
<td>$10^9$</td>
</tr>
<tr>
<td>Entries in Matrix</td>
<td>$n^6$</td>
<td>$10^6$</td>
<td>$10^{14}$</td>
<td>$10^{18}$</td>
</tr>
</tbody>
</table>
Scalability of Linear Solvers

![Graph showing scalability of linear solvers.](image)

- **Problem Size, N** vs. **Estimated Required Solution Time (s)**
- Solvers: Cholesky, Band Cholesky, Jacobi, GSSOR, CG, Optimal, MG
- Time markers: one minute, 1 day, age of your advisor (~60 years), age of the universe

**Computational Modeling with Partial Differential Equations (PDEs)** – p.15
Parallel Computing

Divide and Conquor: break up the problem and send it to several different processors to compute at the same time.
Parallel Computing

Divide and Conquor: break up the problem and send it to several different processors to compute at the same time.

BlueGene/L at Lawrence Livermore National Lab recently exceeded 135 flops (trillion floating point operations per second)
Other Considerations

- Time dependent problems
- Better models often involve nonlinear phenomena
- Other discretization techniques (FE, FV, Spectral, etc.)
- Linear solvers should be scalable in the
  - number of unknowns
  - number of processors
- Linear solvers are often very problem dependent
- Visualization of data can be extremely complicated (3-D, 4-D, ...)
- Is the solution accurate, reasonable, useful?

Each step in the modeling process is a specialized process, but should consider the adjacent steps. Interdisciplinary teams often work together on this process. There is always more to be done!!