Algorithms of Scientific Computing

Fast Poisson Solvers

Michael Bader
Technical University of Munich

Summer 2017
Part I

Excursion: Discrete Models for Heat Transfer and the Poisson Equation

Modelling of Heat Transfer

- objective: compute the temperature distribution of some object
- under certain prerequisites:
  - temperature $T$ at object boundaries given
  - heat sources
  - material parameters $k$, 
- observation from physical experiments: $q \approx k \cdot \delta T$
  heat flow proportional to temperature differences

Michael Bader | Algorithms of Scientific Computing | Fast Poisson Solvers | Summer 2017
A Finite Volume Model

- object: a rectangular metal plate (again)
- model as a collection of small connected rectangular cells
- examine the heat flow across the cell edges
Heat Flow Across the Cell Boundaries

- consider temperature $T_{ij}$ in each cell
- Heat flow across a given edge is proportional to
  - temperature difference $(T_{ij} - T_{i-1,j})$ between adjacent cells
  - length $h$ of the edge
- e.g.: heat flow across the left edge:

$$q_{ij}^{(\text{left})} = k_x (T_{ij} - T_{i-1,j}) h_y$$

$k_x$ depends on material
- heat flow across all edges determines change of heat energy:

$$q_{ij} = k_x (T_{ij} - T_{i-1,j}) h_y + k_x (T_{ij} - T_{i+1,j}) h_y + k_y (T_{ij} - T_{i,j-1}) h_x + k_y (T_{ij} - T_{i,j+1}) h_x$$
Temperature change due to heat flow

- model assumption: conservation of energy, i.e.,
  
in equilibrium, total heat flow equal to 0 for each cell
- or: consider additional source term $F_{ij}$ due to
  - external heating
  - radiation
- $F_{ij} = f_{ij} h_x h_y$ ($f_{ij}$ heat flow per area)
- equilibrium with source term requires $q_{ij} + F_{ij} = 0$:

$$f_{ij} h_x h_y = -k_x h_y (2T_{ij} - T_{i-1,j} - T_{i+1,j})$$
$$-k_y h_x (2T_{ij} - T_{i,j-1} - T_{i,j+1})$$
Finite Volume Model

- divide by $h_x h_y$:

$$f_{ij} = -\frac{k_x}{h_x} (2T_{ij} - T_{i-1,j} - T_{i+1,j})$$

$$-\frac{k_y}{h_y} (2T_{ij} - T_{i,j-1} - T_{i,j+1})$$

- again, system of linear equations
- how to treat boundaries?
  - prescribe temperature in a cell
    (e.g. boundary layer of cells)
  - prescribe heat flow across an edge;
    for example insulation at left edge:

$$q_{ij}^{(\text{left})} = 0$$
From Discrete to Continuous

• system of equations derived from the discrete model:

\[
f_{ij} = -\frac{k_x}{h_x} \left( 2T_{ij} - T_{i-1,j} - T_{i+1,j} \right) - \frac{k_y}{h_y} \left( 2T_{ij} - T_{i,j-1} - T_{i,j+1} \right)
\]

• assumption: heat flow across edges is proportional to temperature difference

\[
q_{ij}^{(left)} = k_x \left( T_{ij} - T_{i-1,j} \right) h_y
\]

• in reality: heat flow proportional to temperature gradient

\[
q_{ij}^{(left)} \approx kh_y \frac{T_{ij} - T_{i-1,j}}{h_x}
\]
From Discrete to Continuous (2)

- replace $k_x$ by $k/h_x$, $k_y$ by $k/h_y$, and get:

\[
\begin{align*}
    f_{ij} &= -\frac{k}{h_x^2} \left(2T_{ij} - T_{i-1,j} - T_{i+1,j}\right) \\
    &\quad - \frac{k}{h_y^2} \left(2T_{ij} - T_{i,j-1} - T_{i,j+1}\right)
\end{align*}
\]

- consider arbitrary small cells: $h_x, h_y \to 0$:

\[
    f_{ij} = -k \left(\frac{\partial^2 T}{\partial x^2}\right)_{ij} - k \left(\frac{\partial^2 T}{\partial y^2}\right)_{ij}
\]

- leads to partial differential equation (PDE):

\[
    -k \left(\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2}\right) = f(x, y)
\]
Part II

Fast Poisson Solvers and the Sine Transform

- situation: solve a system of linear equations

\[-u_{i-1,j} - u_{i+1,j} + 4u_{ij} - u_{i,j-1} - u_{i,j+1} = f_{ij} \quad \forall i, j\]

- or, simpler, for a 1D problem:

\[-u_{n-1} + 2u_n - u_{n+1} = f_n \quad \text{for } n = 1, \ldots, N - 1\]

with \( u_0 = u_N = 0 \)

- consider very fine meshes, e.g. with 1000 \( \times \) 1000 unknowns (in 2D)

- solution can be computed fast, \( O(N) \), in 1D (tri-diagonal system), but hard to solve efficiently in 2D (and even harder in 3D)
Applying the Sine Transform

**Idea:** apply discrete sine transform on $u_n$ and $f_n$

\[
\begin{align*}
    u_n &= 2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N}, \\
    f_n &= 2 \sum_{k=1}^{N-1} F_k \sin \frac{\pi nk}{N}
\end{align*}
\]  

(1)

into the system of equations

\[-u_{n-1} + 2u_n - u_{n+1} = f_n \quad \text{for } n = 1, \ldots, N - 1\]

**Why should that help?**

- corresponding continuous problem is $-u''(x) = f(x)$
- is solved by $u(x) = \sin(x)$, if $f(x) = \sin(x)$ (with $u(x) = 0$ at both boundaries)
- sine modes are **eigenvectors** of the system matrix, and **eigenmodes** of the continuous solution
Applying the Sine Transform (2)

We insert the transformations

\[ u_n = 2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} \quad \text{and} \quad f_n = 2 \sum_{k=1}^{N-1} F_k \sin \frac{\pi nk}{N} \]

into the system of linear equations

\[ -u_{n+1} + 2u_n - u_{n-1} = f_n \quad \text{for} \quad n = 1, \ldots, N - 1 \]
\[ u_0 = u_N = 0, \]

and get, for \( n = 1, \ldots, N - 1 \):

\[
-2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi (n+1)k}{N} + 4 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} - 2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi (n-1)k}{N} = 2 \sum_{k=1}^{N-1} F_k \sin \frac{\pi nk}{N}
\]
Applying the Sine Transform (3)

Use theorems of addition

\[ \sin(A + B) = \sin(A) \cos(B) + \cos(A) \sin(B) \quad \text{and} \]
\[ \sin(A - B) = \sin(A) \cos(B) - \cos(A) \sin(B) \]

applied to:

\[ \sin \left( \frac{\pi(n + 1)k}{N} \right) = \sin \left( \frac{\pi nk}{N} + \frac{\pi k}{N} \right) \quad \text{and} \]
\[ \sin \left( \frac{\pi(n - 1)k}{N} \right) = \sin \left( \frac{\pi nk}{N} - \frac{\pi k}{N} \right) \]

We have the situation

\[ \sin(A + B) + \sin(A - B) = 2 \sin(A) \cos(B) \]

or, particularly:

\[ \sin \left( \frac{\pi(n + 1)k}{N} \right) + \sin \left( \frac{\pi(n - 1)k}{N} \right) = 2 \sin \left( \frac{\pi nk}{N} \right) \cos \left( \frac{\pi k}{N} \right) \]
Applying the Sine Transform (4)

Use theorems of addition in the left-hand side:

\[-2 \sum_{k=1}^{N-1} \left( U_k \sin \frac{\pi(n+1)k}{N} + U_k \sin \frac{\pi(n-1)k}{N} \right) + 4 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} \]

\[-4 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} \cos \frac{\pi k}{N} + 4 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} \]

and obtain simplified system of equations:

\[-4 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} \cos \frac{\pi k}{N} + 4 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} = 2 \sum_{k=1}^{N-1} F_k \sin \frac{\pi nk}{N} \]

\[\Leftrightarrow 2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N} \left( 1 - \cos \frac{\pi k}{N} \right) = \sum_{k=1}^{N-1} F_k \sin \frac{\pi nk}{N} \]
Solution of the System of Equations

We transformed the system of equations into

\[
2 \sum_{k=1}^{N-1} U_k \sin \left( \frac{\pi nk}{N} \right) \left( 1 - \cos \left( \frac{\pi k}{N} \right) \right) = \sum_{k=1}^{N-1} F_k \sin \left( \frac{\pi nk}{N} \right)
\]

All equations are satisfied, if

\[
2U_k \left( 1 - \cos \left( \frac{\pi k}{N} \right) \right) = F_k \quad \text{for all } k = 1, \ldots, N - 1
\]

This is true, if we set

\[
U_k = \frac{F_k}{2 - 2 \cos \left( \frac{\pi k}{N} \right)} \quad \text{for all } k = 1, \ldots, N - 1.
\]
Fast Poisson Solver – Algorithm

1. Compute the coefficients $F_k$ by a **Fast Sine Transform**:

$$F_k = \frac{1}{N} \sum_{n=1}^{N-1} f_n \sin \frac{\pi nk}{N}$$

2. Compute all coefficients $U_k$ from the $F_k$ as

$$U_k = \frac{F_k}{2 - 2 \cos \frac{\pi k}{N}} \text{ for all } k = 1, \ldots, N - 1.$$ 

3. Compute the $u_n$ from the $U_k$ by means of an Inverse **Fast Sine Transform**:

$$u_n = 2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi nk}{N},$$
Fast Poisson Solver – Algorithm (2)

Computational Costs:

- the two Fast Sine Transforms require $O(N \log N)$ operations
- step 2 needs only $O(N)$ operations

$\Rightarrow$ total computational effort is $O(N \log N)$

- thus: slower than solving the tridiagonal system of equations directly, which has effort $O(N)$
- however: pays off in 2D and higher-dimensional settings! (due to similar complexity)

When can the Algorithm be applied:

- boundary conditions need to be $u_0 = u_N = 0$ (otherwise: different transform required)
- requires rectangular/cuboid domain and Cartesian mesh
- requires uniform material parameters $k$