The Control-Volume Finite-Difference Approximation to the Diffusion Equation

ME 448/548 Notes

Gerald Recktenwald
Portland State University
Department of Mechanical & Materials Engineering
gerry@pdx.edu
21 January 2014

Motivation

Numerical solution of the 2D Poisson equation is the next step in developing our knowledge of CFD technique.

- Introduce the Finite Volume Method
  - Naturally deal with material discontinuity
  - Can naturally enforce conservation of mass and energy
  - Core idea in many (not all) commercial CFD codes

- Extend analysis to two spatial dimensions
  - More interesting practical applications
  - More complex data structures
  - More complex procedures to solve the $Ax = b$ problem
Overview

Goals for this unit

• Introduce the Poisson equation
  ▶ A model of steady heat conduction with a source term
  ▶ Form of the pressure equation for incompressible flow
  ▶ Precursor to the generalized advection diffusion equation

• Use the finite volume method to obtain discrete equations
• Allow for non-uniform mesh, diffusion coefficient, and source term.
• Introduce a set of MATLAB codes for 2D Control Volume Finite Difference (CVFD)
• Demonstrate solutions for three model problems.
• Measure truncation error for a model problem with a simple solution
• Apply to fully-developed flow in rectangular ducts

Model Problem

The two dimensional diffusion equation in Cartesian coordinates is

$$\frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left( \Gamma \frac{\partial \phi}{\partial y} \right) + S = 0$$

(1)

where $\phi$ is the scalar field, $\Gamma$ is the diffusion coefficient, and $S$ is the source term.

Example: Heat conduction in a rectangular domain

$$\Gamma = k, \text{ thermal conductivity or } \Gamma = \frac{k}{\rho c_p} = \alpha$$

$$S = \text{ volumetric heat source}$$
2D Cartesian Finite Volume Mesh

Finite Volume Mesh: Compass Point Notation

Nodes are labelled with their position relative to the typical point “P”. Capital letters designate node locations.

N, S, E, W are the names of north, south, east and west neighbors.

\( \phi_N, \phi_S, \phi_E, \phi_W \) are the values of \( \phi \) at the north, south, east and west neighbor nodes.

Use of compass point names simplifies the algebra. For example the value of \( \phi \) at point N is \( \phi_N \) or \( \phi_{i+1,j+1} \), but \( \phi_N \) is simpler to write.

Compass point notation is an historic convention that does not extend to unstructured meshes.
Finite Volume Mesh

Detailed notation for a typical 2D control volume

Lowercase letters designate the interface between the nodes.

\( x_e \) and \( x_w \) are the \( x \) coordinates of the east and west faces of the control volume around \( P \).

\( y_n \) and \( y_s \) are the \( y \) coordinates of the north and south faces of the control volume around \( P \).

Similarly, \( \phi_n, \phi_s, \phi_e, \phi_w \) are the values of \( \phi \) at the north, south, east and west control volume interfaces.

Regardless of whether the control volumes sizes are uniform, node \( P \) is always located in the geometric center of each control volume.

Thus,

\[
\begin{align*}
    x_P - x_w &= x_e - x_P = \frac{\Delta x}{2} \\
    y_P - y_s &= y_n - y_P = \frac{\Delta y}{2}
\end{align*}
\]

for uniform or non-uniform meshes.
Finite Volume Mesh

Detailed notation for a typical 2D control volume

In contrast, the distances between the nodes is not assumed to be uniform

\[ \delta x_e = x_E - x_P \neq \delta x_w = x_P - x_W \]
\[ \delta y_n = y_N - y_P \neq \delta y_s = y_P - y_S \]

Note the use of upper and lower case subscripts: Upper case refers to nodes; lower case refers to interfaces.

Convert the Differential Equation to a Discrete Equation

Integrate over the control volume

\[
\int_{y_s}^{y_n} \int_{x_w}^{x_e} \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) \, dx \, dy
\]
\[= \int_{y_s}^{y_n} \left[ \left( \Gamma \frac{\partial \phi}{\partial x} \right)_e - \left( \Gamma \frac{\partial \phi}{\partial x} \right)_w \right] \, dy \]
\[\approx \left[ \left( \Gamma \frac{\partial \phi}{\partial x} \right)_e - \left( \Gamma \frac{\partial \phi}{\partial x} \right)_w \right] \Delta y \]
\[\approx \left[ \Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} - \Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} \right] \Delta y \]

The final step is obtained by using central difference approximations for the derivatives at the interfaces.
**Convert the Differential Equation to a Discrete Equation**

Integrate the source term

\[
\int_{xw}^{xe} \int_{ys}^{yn} S \, dy \, dx \approx S_P \Delta x \, \Delta y \quad (2)
\]

*Note:* The Control Volume Finite Difference (CVFD) method treats the source term and diffusion coefficients as piecewise constants. This is a rather crude approximation, say, compared to allowing the source term and diffusion coefficient to vary linearly within the control volume.

However, piecewise constant profiles of \( S \) and \( \Gamma \) allow the method to be conservative, i.e., conserving mass or energy, automatically. The conservative nature of the CVFD method is one of its primary strengths.

Putting pieces back into the model equation gives the discrete system of equations

\[
-a_S \phi_S - a_W \phi_W + a_P \phi_P - a_E \phi_E - a_N \phi_N = b \quad (3)
\]

where

\[
a_E = \frac{\Gamma_e}{\Delta x \, \delta x_e}, \quad a_W = \frac{\Gamma_w}{\Delta x \, \delta x_w}, \quad a_N = \frac{\Gamma_n}{\Delta y \, \delta y_n}, \quad a_S = \frac{\Gamma_s}{\Delta y \, \delta y_s}
\]

\[
a_P = a_E + a_W + a_N + a_S \quad (4)
\]

\[
b = S_P \quad (5)
\]

This is *not* a tridiagonal system of equations.
Non-uniform $\Gamma$

Continuity of fluxes at the interface requires

$$\Gamma_P \frac{\partial \phi}{\partial x} \bigg|_{x_e^-} = \Gamma_E \frac{\partial \phi}{\partial x} \bigg|_{x_e^+} = \Gamma_e \frac{\partial \phi}{\partial x} \bigg|_{x_e}$$

Use central difference approximations

$$\Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} = \Gamma_P \frac{\phi_e - \phi_P}{\delta x_e^-} \tag{6}$$

$$\Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} = \Gamma_E \frac{\phi_E - \phi_e}{\delta x_e^+} \tag{7}$$

Equations 6 and 7 can be rearranged as

$$\phi_e - \phi_P = \frac{\delta x_e^-}{\Gamma_P} \frac{\Gamma_e}{\delta x_e} (\phi_E - \phi_P) \tag{8}$$

$$\phi_E - \phi_e = \frac{\delta x_e^+}{\Gamma_E} \frac{\Gamma_e}{\delta x_e} (\phi_E - \phi_P) \tag{9}$$

Add Equation 8 and Equation 9

$$\phi_E - \phi_P = \frac{\Gamma_e}{\delta x_e} (\phi_E - \phi_P) \left[\frac{\delta x_e^-}{\Gamma_P} + \frac{\delta x_e^+}{\Gamma_E}\right].$$

Cancel the factor of $(\phi_E - \phi_P)$ and solve for $\Gamma_e/\delta x_e$ to get

$$\frac{\Gamma_e}{\delta x_e} = \left[\frac{\delta x_e^-}{\Gamma_P} + \frac{\delta x_e^+}{\Gamma_E}\right]^{-1} = \frac{\Gamma_E \Gamma_P}{\delta x_e \Gamma_E + \delta x_e^+ \Gamma_P}.$$
**Non-uniform** $\Gamma$

Thus, the diffusion coefficient at the interface that results in flux continuity is

$$\Gamma_e = \frac{\Gamma_E \Gamma_P}{\beta \Gamma_E + (1 - \beta) \Gamma_P}$$  \hspace{1cm} (10)

where

$$\beta \equiv \frac{\delta x_e-}{\delta x_e} = \frac{x_e - x_P}{x_E - x_P}$$  \hspace{1cm} (11)

An analogous derivation gives formulas for $\Gamma_w$, $\Gamma_n$, and $\Gamma_s$.

---

**Solving the System of Equations**

Regardless of uniform or variable $\Gamma$, the discrete equation has a five-point stencil, and the discrete equation for any interior node can be written.

$$-a_S\phi_S - a_W\phi_W + a_P\phi_P - a_E\phi_E - a_N\phi_N = b$$  \hspace{1cm} (12)

To set up the matrix for this system of equations, we need to re-number the unknowns.

**Important**: $i$ and $j$ subscripts for the mesh are *not* the same as the row and column indices in the system $Ax = b$. 

---
Solving the System of Equations

Order the nodes:

\[ n = i + (j - 1)nx \]

\( n \) is the node number (row number) in \( Ax = b \). \( i \) and \( j \) are the mesh indices corresponding to \( x_i \) and \( y_j \).

With natural ordering the neighbors in the compass point notation have these indices:

- \( np = i + (j-1)nx \)
- \( ne = np + 1 \)
- \( nw = np - 1 \)
- \( nn = np + nx \)
- \( ns = np - nx \)

With natural ordering the neighbors in the compass point notation have these indices:

- \( np = i + (j-1)nx \)
- \( ne = np + 1 \)
- \( nw = np - 1 \)
- \( nn = np + nx \)
- \( ns = np - nx \)

This leads to a vector of unknowns

\[
\begin{pmatrix}
\phi_{1,1} \\
\phi_{2,1} \\
\vdots \\
\phi_{nx,1} \\
\phi_{1,2} \\
\phi_{2,2} \\
\vdots \\
\phi_{i,j} \\
\vdots \\
\phi_{nx,ny}
\end{pmatrix}
\leftrightarrow
\begin{pmatrix}
\phi_1 \\
\phi_2 \\
\vdots \\
\phi_{nx} \\
\phi_{nx+1} \\
\phi_{nx+2} \\
\vdots \\
\phi_n \\
\vdots \\
\phi_N
\end{pmatrix}
\] (13)
Solving the System of Equations

The coefficient matrix has 5 non-zero diagonals

\[ A = \begin{pmatrix}
-\alpha_S & -\alpha_w & \alpha_p & -\alpha_e & -\alpha_n \\
-\alpha_w & -\alpha_w & -\alpha_e & -\alpha_e & -\alpha_n \\
\alpha_p & -\alpha_e & -\alpha_e & -\alpha_e & -\alpha_n \\
-\alpha_e & -\alpha_e & -\alpha_e & -\alpha_e & -\alpha_n \\
-\alpha_n & -\alpha_n & -\alpha_n & -\alpha_n & -\alpha_n
\end{pmatrix} \]

Algorithm for obtaining the numerical solution

1. Define physical parameters: \( L_x, L_y, \Gamma(x, y) \) and boundary conditions
2. Define the mesh: \( n_x \) and \( n_y \) if uniform
3. Compute the coefficient matrix
4. Solve the system of equations \( A\phi = b \), where \( \phi \) is the vector of unknowns.
5. Post-process to visualize the solution

The Poisson equation is steady. Each step is performed only once.
Structured Mesh

The CVFD MATLAB codes use structured meshes. In size.

- The cells in the domain are topologically equivalent to a rectangular array
- The cells need not be uniform in size.
- Each cell not on a boundary touches four other cells
- Each row has the same number of cells
- Each column has the same number of cells

![Uniform Mesh and Block-Uniform Mesh Diagram]

\[
\begin{align*}
&\Delta x, \Delta y \\
&L_x, n_x \\
&L_y, n_y \\
&L_y_3, n_y_3 \\
&L_y_2, n_y_2 \\
&L_y_1, n_y_1
\end{align*}
\]

Boundary Conditions (part 1)

<table>
<thead>
<tr>
<th>Boundary type</th>
<th>Boundary Condition</th>
<th>Post-processing in \texttt{fvpost}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specified $T$</td>
<td>Compute $\dot{q}$ from discrete approximation to Fourier’s law.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{q} = k \frac{T_b - T_i}{x_b - x_i}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where $T_i$ and $T_b$ are interior and boundary temperatures, respectively.</td>
</tr>
<tr>
<td>2</td>
<td>Specified $\dot{q}$</td>
<td>Compute $T_b$ from discrete approximation to Fourier’s law.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_b = T_i + \frac{\dot{q} x_b - x_i}{k}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where $T_i$ and $T_b$ are interior and boundary temperatures, respectively.</td>
</tr>
</tbody>
</table>
Boundary Conditions (part 2)

<table>
<thead>
<tr>
<th>Boundary type</th>
<th>Boundary Condition</th>
<th>Post-processing in</th>
<th>fvpost</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Convection</td>
<td>From specified $h$ and $T_\infty$, compute boundary temperature and heat flux through the cell face on the boundary. Continuity of heat flux requires $-k \frac{T_b - T_i}{x_b - x_i} = h(T_b - T_{amb})$ which can be solved for $T_b$ to give $T_b = hT_{amb} + \frac{(k/\delta x_e)T_i}{h + (k/\delta x_e)}$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Symmetry</td>
<td>$q'' = 0$. Set boundary $T_b$ equal to adjacent interior $T_i$.</td>
<td></td>
</tr>
</tbody>
</table>

MATLAB codes for obtaining the numerical solution

A set of general purpose codes has been written to facilitate experimentation with the CVFD method.

<table>
<thead>
<tr>
<th>Algorithm Tasks</th>
<th>Core Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define the mesh</td>
<td>fvUniformMesh or fvUniBlockMesh</td>
</tr>
<tr>
<td>Define boundary conditions</td>
<td></td>
</tr>
<tr>
<td>Compute finite-volume coefficients for interior cells</td>
<td>fvcoef</td>
</tr>
<tr>
<td>Adjust coefficients for boundary conditions</td>
<td>fvbc</td>
</tr>
<tr>
<td>Solve system of equations</td>
<td></td>
</tr>
<tr>
<td>Assemble coefficient matrix</td>
<td>fvAmatrix</td>
</tr>
<tr>
<td>Solve</td>
<td></td>
</tr>
<tr>
<td>Compute boundary values and/or fluxes</td>
<td>fvpost</td>
</tr>
<tr>
<td>Plot results</td>
<td></td>
</tr>
</tbody>
</table>
Model Problem 1

Choose a source term that may be physically unrealistic, but one that gives an exact solution that is easy to evaluate

\[ S = \left[ \left( \frac{\pi}{L_x} \right)^2 + \left( \frac{2\pi}{L_y} \right)^2 \right] \sin \left( \frac{\pi x}{L_x} \right) \sin \left( \frac{2\pi y}{L_y} \right) \]

The exact solution is

\[ \phi = \sin \left( \frac{\pi x}{L_x} \right) \sin \left( \frac{2\pi y}{L_y} \right) \]

Main code to solve this problem is in demoModel1.m
Solutions to Model Problem 1 Show Correct Truncation Error

The local truncation error at each node is
\[ e_i \sim O(\Delta x^2) \].
Since the exact solution is known we can compute
\[
\frac{\|e\|_2}{N} = \frac{\sqrt{\sum e_i^2}}{N} \sim \frac{\sqrt{Ne^2}}{N} = \frac{\bar{e}}{\sqrt{N}},
\]
where \( N = n_x n_y \) is the total number of interior nodes in the domain, and \( \bar{e} \) is the average truncation error per node.

Since \( e_i \sim O(\Delta x^2) \), \( N \sim n_x^2 \), and \( \Delta x = L_x/(n_x + 1) \), we can estimate
\[
\frac{\|e\|_2}{N} \sim \frac{\bar{e}}{\sqrt{N}} = O(\Delta x^2) = O\left(\frac{L_x^2}{n_x(n_x + 1)^2}\right) = O\left(\frac{1}{n_x}\right)^3 = O(\Delta x^3).
\]

Model Problem 2

Uniform source term: \( S = 1 \).
Analytical solution is an infinite series
Code in demoModel2.m
Model Problem 3

Heat conduction in a rectangle consisting of two material regions.
- Inner rectangle with high conductivity
- Outer rectangle with low conductivity
- Inner region has uniform heat source

The discontinuity and difference in material properties can be used to stress the solution algorithm.

\[ \alpha = \frac{\Gamma_2}{\Gamma_1} \]

The analytical solution does not exist. Code in demoModel3.m
Model Problem 4: Fully Developed Flow in a Rectangular Duct

For simple fully-developed flow the governing equation for the axial velocity \( w \) is

\[
\mu \left[ \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right] - \frac{dp}{dz} = 0
\]

This corresponds to the generic model equation with

\[
\phi = w, \quad \Gamma = \mu \text{ (constant)}, \quad S = -\frac{dp}{dz}.
\]

The symmetry in the problem allows alternative ways of defining the numerical model

For the full duct simulation depicted on the left hand side, the boundary conditions are no slip conditions on all four walls.

\[
w(x, 0) = w(x, L_y) = w(0, y) = w(L_x, y) = 0. \quad \text{ (full duct)}
\]
Model Problem 4: Fully Developed Flow in a Rectangular Duct

The symmetry in the problem allows alternative ways of defining the numerical model

For the quarter duct simulation depicted on the right hand side, the boundary conditions are no slip conditions on the solid walls \((x = L_x \text{ and } y = L_y)\)

\[
w(L_y, y) = w(x, L_x) = 0 \quad \text{(quarter duct)}
\]

and symmetry conditions on the other two planes

\[
\left. \frac{\partial u}{\partial x} \right|_{x=0} = \left. \frac{\partial u}{\partial y} \right|_{y=0} = 0.
\]