Computational Science:
Computational Methods in Engineering

Matrix Operators

Outline

• One-Dimensional Matrix Operators
• Incorporating Boundary Conditions
One-Dimensional Matrix Operators

Functions Vs. Operations (1 of 2)

\[ a(x) \frac{\partial^2}{\partial x^2} f(x) + \gamma b(x) \frac{\partial}{\partial x} f(x) + c(x) f(x) = g(x) \]

**Operations**
Everything else in a differential equation is something that operates on a function.

- \( a(x), b(x), c(x) = \) point-by-point multiplication on \( f(x) \)
- \( \frac{\partial}{\partial x}, \frac{\partial^2}{\partial x^2} = \) calculates derivatives of \( f(x) \)
- \( \gamma = \) scales entire \( f(x) \)

**Functions**
Functions only appear in a differential equation as the unknown or as the excitation.

- \( f(x) = \) unknown
- \( g(x) = \) excitation
Functions Vs. Operations (2 of 2)

\[
\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} D \end{bmatrix}^2 \begin{bmatrix} f \end{bmatrix} + \gamma \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} D \end{bmatrix} \begin{bmatrix} f \end{bmatrix} + [C][f] = [g]
\]

Operations
Operations are always stored in square matrices. Any linear operation can be put into matrix form.

\[
[L] = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1M} \\ l_{21} & l_{22} & \cdots & l_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ l_{M1} & l_{M2} & \cdots & l_{MM} \end{bmatrix}
\]

Functions
Functions are stored as column vectors.

\[
[f] = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_M \end{bmatrix}, \quad [g] = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_M \end{bmatrix}
\]

Point-by-Point Multiplication (1 of 2)
Since the functions are stored in column-vector form, how are point-by-point multiplications performed using a square matrix?

\[
b(x)f(x) \rightarrow [B][f]
\]
Point-by-Point Multiplication (2 of 2)

Since the functions are stored in column-vector form, how are point-by-point multiplications performed using a square matrix?

\[ b(x)f(x) \rightarrow [B][f] \]

First-Order Partial Derivative (1 of 2)

How can a square matrix be constructed so that when it premultiplies a vector it calculates a vector containing the first-order partial derivative?

\[ \frac{\partial}{\partial x} f(x) \rightarrow [D_x][f] \]
First-Order Partial Derivative (2 of 2)

How can a square matrix be constructed so that when it premultiplies a vector it calculates a vector containing the first-order partial derivative?

\[ \frac{\partial}{\partial x} f(x) \rightarrow [D_x][f] \]

Second-Order Partial Derivative (1 of 2)

How can a square matrix be constructed so that when it premultiplies a vector it calculates a vector containing the second-order partial derivative?

\[ \frac{\partial^2}{\partial x^2} f(x) \rightarrow [D_x^2][f] \]
Second-Order Partial Derivative (2 of 2)

How can a square matrix be constructed so that when it premultiplies a vector it calculates a vector containing the second-order partial derivative?

\[ \frac{\partial^2}{\partial x^2} f(x) \rightarrow \begin{bmatrix} D_x^2 \end{bmatrix}[f] \]

Why Are Separate Derivative Matrices Needed for 1st and 2nd-Order Derivative Matrices?

It is known that,

\[ \frac{\partial^2}{\partial x^2} = \frac{\partial}{\partial x} \frac{\partial}{\partial x} \]

So why can’t \( D_x^{(2)} \) simply be calculated as \( D_x^{(2)} = D_x^{(1)} D_x^{(1)} \)?

It turns out this does not make efficient use of the grid. For a 5-point, 1D grid, this would be

\[
D_x^{(1)} D_x^{(1)} = \frac{1}{(2\Delta x)^2} \begin{bmatrix}
-1 & 0 & 1 & 0 & 0 \\
0 & -2 & 0 & 1 & 0 \\
1 & 0 & -2 & 0 & 1 \\
0 & 1 & 0 & -2 & 0 \\
0 & 0 & 1 & 0 & -1
\end{bmatrix}
\]

\[
D_x^{(2)} = \frac{1}{\Delta x^2} \begin{bmatrix}
-2 & 1 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 \\
0 & 0 & 1 & -2 & 0 \\
0 & 0 & 0 & 1 & -2
\end{bmatrix}
\]

This is not as accurate because it calculates the derivative with poorer grid resolution than is available. This derive matrix makes optimal use of the available grid resolution.
USE SPARSE MATRICES!!!!!!!

WARNING !!

The derivative operators will be **EXTREMELY** large matrices.

For a small grid that is just $100 \times 200$ points:

- Total Number of Points: 20,000
- Size of Derivate Operators: $20,000 \times 20,000$
- Total Elements in Matrices: 400,000,000
- Memory to Store One Full Matrix: 6 Gb
- Memory to Store One Sparse Matrix: 1 Mb

**NEVER AT ANY POINT** should you use **FULL MATRICES** in the finite-difference method. Not even for intermediate steps. **NEVER!!**

---

Placing Diagonals into Sparse Matrices in MATLAB

```matlab
M = 6;
Z = sparse(M,M);
d = ones(M,1);
A = spdiags(d,0,Z);

A =
[ 1 0 0 0 0 0 ]
[ 0 1 0 0 0 0 ]
[ 0 0 1 0 0 0 ]
[ 0 0 0 1 0 0 ]
[ 0 0 0 0 1 0 ]
[ 0 0 0 0 0 1 ]

A =
[ 0 1 0 0 0 0 ]
[ -1 0 1 0 0 0 ]
[ 0 -1 0 1 0 0 ]
[ 0 0 -1 0 1 0 ]
[ 0 0 0 -1 0 1 ]
[ 0 0 0 0 -1 0 ]
```

---

---
Incorporating Boundary Conditions

Dirichlet Boundary Conditions
(1 of 2)

The simplest boundary condition is to assume all function values outside of the grid are zero.

\[
\frac{d^2 f_1}{dx^2} \approx \frac{f_{i-1} - 2f_i + f_{i+1}}{(\Delta x)^2}
\]

\[
\frac{d^2 f_2}{dx^2} \approx \frac{f_1 - 2f_2 + f_3}{(\Delta x)^2}
\]

\[
\frac{d^2 f_3}{dx^2} \approx \frac{f_{i-1} - 2f_i + f_{i+1}}{(\Delta x)^2}
\]

\[
\frac{d^2 f_4}{dx^2} \approx \frac{0 - 2f_4 + f_5}{(\Delta x)^2}
\]

\[
\frac{d^2 f_5}{dx^2} \approx \frac{f_4 - 2f_5 + 0}{(\Delta x)^2}
\]

\[
\frac{d^2 f_6}{dx^2} \approx \frac{f_5 - 2f_6 + f_7}{(\Delta x)^2}
\]
Dirichlet Boundary Conditions
(2 of 2)

If the problem is periodic (i.e., keeps repeating), then the value outside of the grid is the same as the value at the opposite side of the grid.
Periodic Boundary Conditions
(2 of 2)

\[
\begin{align*}
\frac{d^2 f_i}{dx^2} & \equiv \frac{f_{i-2} + f_{i+2}}{(\Delta x)^2} \\
\frac{d^2 f_2}{dx^2} & \equiv \frac{f_{2-2} + f_{2+2}}{(\Delta x)^2} \\
\frac{d^2 f_3}{dx^2} & \equiv \frac{f_{3-2} + f_{3+2}}{(\Delta x)^2} \\
\frac{d^2 f_6}{dx^2} & \equiv \frac{f_{6-2} + f_{6+2}}{(\Delta x)^2} \\
\frac{d^2 f_7}{dx^2} & \equiv \frac{f_{7-2} + f_{7+2}}{(\Delta x)^2}
\end{align*}
\]

\[
\begin{bmatrix}
-2 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & -2
\end{bmatrix}
\]

\[
[D^2]
\]

Neuman Boundary Conditions
(1 of 3)

The Neuman boundary condition allows functions to continue linearly off of the grid as if to infinity.

\[
\begin{align*}
\frac{df_i}{dx} & \equiv \frac{f_{i+1} - f_{i-1}}{2\Delta x} \\
\frac{d^2 f_i}{dx^2} & \equiv \frac{f_{i+1} - 2f_i + f_{i-1}}{(\Delta x)^2}
\end{align*}
\]

\[
\begin{align*}
\frac{df_1}{dx} & \equiv \frac{f_2 - f_1}{\Delta x} \\
\frac{d^2 f_1}{dx^2} & \equiv 0 \\
\frac{df_7}{dx} & \equiv \frac{f_7 - f_6}{\Delta x} \\
\frac{d^2 f_7}{dx^2} & \equiv 0
\end{align*}
\]
Neuman Boundary Conditions  
(2 of 3)

$$\begin{align*}
\frac{df_i}{dx} &\approx \frac{2f_i - 2f_{i-1}}{2\Delta x} \\
\frac{df_i}{dx} &\approx \frac{f_i - f_{i-1}}{\Delta x} \\
\frac{df_i}{dx} &\approx \frac{f_i - f_{i-1}}{2\Delta x} \\
\frac{df_i}{dx} &\approx \frac{f_i - f_{i-1}}{\Delta x} \\
\frac{df_i}{dx} &\approx \frac{f_i - f_{i-1}}{2\Delta x} \\
\frac{df_i}{dx} &\approx \frac{f_i - 2f_i + f_{i+1}}{2\Delta x} \\
\frac{df_i}{dx} &\approx \frac{f_i - 2f_i + f_{i+1}}{\Delta x^2}
\end{align*}$$

$$\Rightarrow \frac{1}{2\Delta x} \begin{bmatrix}
-2 & 2 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -2 & 2 & 0
\end{bmatrix} \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7
\end{bmatrix}$$

Neuman Boundary Conditions  
(3 of 3)

$$\begin{align*}
\frac{d^2f_i}{dx^2} &\approx 0 \\
\frac{d^2f_i}{dx^2} &\approx \frac{f_i - 2f_i + f_{i+1}}{\Delta x^2} \\
\frac{d^2f_i}{dx^2} &\approx \frac{f_i - 2f_i + f_{i+1}}{\Delta x^2} \\
\frac{d^2f_i}{dx^2} &\approx \frac{f_i - 2f_i + f_{i+1}}{\Delta x^2} \\
\frac{d^2f_i}{dx^2} &\approx \frac{f_i - 2f_i + f_{i+1}}{\Delta x^2} \\
\frac{d^2f_i}{dx^2} &\approx 0
\end{align*}$$

$$\Rightarrow \frac{1}{(\Delta x)^2} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & -2 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7
\end{bmatrix}$$
High-Order Boundary Conditions (1 of 2)

Here we estimate the derivative at the boundaries using special finite-difference equations derived specifically for these points.

\[
\frac{d^2 f_i}{dx^2} \approx \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}
\]

\[
\frac{d^2 f_1}{dx^2} \approx \frac{2f_1 - 5f_2 + 4f_3 - f_4}{h^2}
\]

\[
\frac{d^2 f_7}{dx^2} \approx -\frac{f_4 + 4f_5 - 5f_6 + 2f_7}{h^2}
\]

High-Order Boundary Conditions (2 of 2)

\[
\begin{align*}
\begin{bmatrix}
2 & -5 & 4 & -1 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
1 & (\Delta x)^2 \\
\end{bmatrix}
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7 \\
\end{bmatrix}
\end{align*}
\]

\[
\begin{bmatrix}
0 & 0 & 0 & 1 & -2 & 1 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & -2 & 1 \\
0 & 0 & 0 & -1 & 4 & -5 & 2 \\
\end{bmatrix}
\]