Part 5
Chapter 19
Numerical Differentiation

PowerPoints organized by Dr. Michael R. Gustafson II, Duke University
Chapter Objectives

• Understanding the application of high-accuracy numerical differentiation formulas for equispaced data.
• Knowing how to evaluate derivatives for unequally spaced data.
• Understanding how Richardson extrapolation is applied for numerical differentiation.
• Recognizing the sensitivity of numerical differentiation to data error.
• Knowing how to evaluate derivatives in MATLAB with the `diff` and `gradient` functions.
• Knowing how to generate contour plots and vector fields with MATLAB.
Differentiation

• The mathematical definition of a derivative begins with a difference approximation:

\[ \frac{\Delta y}{\Delta x} = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} \]

and as \( \Delta x \) is allowed to approach zero, the difference becomes a derivative:

\[ \frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} \]
High-Accuracy Differentiation Formulas

• Taylor series expansion can be used to generate high-accuracy formulas for derivatives by using linear algebra to combine the expansion around several points.

• Three categories for the formula include forward finite-difference, backward finite-difference, and centered finite-difference.
Forward Finite-Difference

First Derivative

\[
f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}
\]

Error \( O(h) \)

\[
f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}
\]

Error \( O(h^2) \)

Second Derivative

\[
f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}
\]

Error \( O(h) \)

\[
f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}
\]

Error \( O(h^2) \)

Third Derivative

\[
f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}
\]

Error \( O(h) \)

\[
f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3}
\]

Error \( O(h^2) \)

Fourth Derivative

\[
f''''(x_i) = \frac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}
\]

Error \( O(h) \)

\[
f''''(x_i) = \frac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}
\]

Error \( O(h^2) \)
Backward Finite-Difference

First Derivative
\[ f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} \]
\[ f'(x_i) = \frac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h} \]

Second Derivative
\[ f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2} \]
\[ f''(x_i) = \frac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2} \]

Third Derivative
\[ f'''(x_i) = \frac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3} \]
\[ f'''(x_i) = \frac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3} \]

Fourth Derivative
\[ f^{\prime\prime\prime}(x_i) = \frac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4} \]
\[ f^{\prime\prime\prime}(x_i) = \frac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4} \]
Centered Finite-Difference

First Derivative
\[
f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}
\]
\[
f'(x_i) = -f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})
\]
\[\frac{12h}{2h}
\]

Second Derivative
\[
f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}
\]
\[
f''(x_i) = -f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})
\]
\[\frac{12h^2}{2h^2}
\]

Third Derivative
\[
f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{h^3}
\]
\[
f'''(x_i) = -f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})
\]
\[\frac{2h^3}{8h^3}
\]

Fourth Derivative
\[
f''''(x_i) = \frac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{h^4}
\]
\[
f''''(x_i) = -f(x_{i+3}) + 12f(x_{i+2}) + 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) + f(x_{i-3})
\]
\[\frac{6h^4}{6h^4}
\]
Richardson Extrapolation

As with integration, the Richardson extrapolation can be used to combine two lower-accuracy estimates of the derivative to produce a higher-accuracy estimate.

For the cases where there are two $O(h^2)$ estimates and the interval is halved ($h_2=h_1/2$), an improved $O(h^4)$ estimate may be formed using:

$$D = \frac{4}{3} D(h_2) - \frac{1}{3} D(h_1)$$

For the cases where there are two $O(h^4)$ estimates and the interval is halved ($h_2=h_1/2$), an improved $O(h^6)$ estimate may be formed using:

$$D = \frac{16}{15} D(h_2) - \frac{1}{15} D(h_1)$$

For the cases where there are two $O(h^6)$ estimates and the interval is halved ($h_2=h_1/2$), an improved $O(h^8)$ estimate may be formed using:

$$D = \frac{64}{63} D(h_2) - \frac{1}{63} D(h_1)$$
Unequally Spaced Data

• One way to calculated derivatives of unequally spaced data is to determine a polynomial fit and take its derivative at a point.

• As an example, using a second-order Lagrange polynomial to fit three points and taking its derivative yields:

\[
f'(x) = f(x_0) \frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)} + f(x_1) \frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)} + f(x_2) \frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}
\]
A shortcoming of numerical differentiation is that it tends to amplify errors in data, whereas integration tends to smooth data errors.

One approach for taking derivatives of data with errors is to fit a smooth, differentiable function to the data and take the derivative of the function.
Numerical Differentiation with MATLAB

- MATLAB has two built-in functions to help take derivatives, `diff` and `gradient`:
  - `diff(x)`
    - Returns the difference between adjacent elements in `x`
  - `diff(y)./diff(x)`
    - Returns the difference between adjacent values in `y` divided by the corresponding difference in adjacent values of `x`
Numerical Differentiation with MATLAB

- \( fx = \text{gradient}(f, h) \)
  Determines the derivative of the data in \( f \) at each of the points. The program uses forward difference for the first point, backward difference for the last point, and centered difference for the interior points. \( h \) is the spacing between points; if omitted \( h=1 \).

- The major advantage of \texttt{gradient} over \texttt{diff} is \texttt{gradient’s result is the same size as the original data}.

- Gradient can also be used to find partial derivatives for matrices:
  \[
  [fx, fy] = \text{gradient}(f, h)
  \]
Visualization

• MATLAB can generate contour plots of functions as well as vector fields. Assuming \( x \) and \( y \) represent a meshgrid of \( x \) and \( y \) values and \( z \) represents a function of \( x \) and \( y \),
  - \( \text{contour}(x, y, z) \) can be used to generate a contour plot
  - \([fx, fy] = \text{gradient}(z, h)\) can be used to generate partial derivatives and
  - \( \text{quiver}(x, y, fx, fy) \) can be used to generate vector fields