Computational Science: Computational Methods in Engineering

Polynomial Technique for Deriving Finite-Difference Approximations

Outline

• Polynomial technique
• Examples
Polynomial Technique for Deriving Finite-Difference Approximations

Concept of Using Polynomials

An $N$th-order polynomial can be fit to a set of $N + 1$ points.

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_Nx^N$$

After the curve fit, the function or any of its derivatives can be interpolated at any point from the polynomial.

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + \cdots + a_Nx^N$$

$$f'(x) = a_1 + 2a_2x + 3a_3x^2 + 4a_4x^3 + \cdots + Na_Nx^{N-1}$$

$$f''(x) = 2a_2 + 6a_3x + 12a_4x^2 + \cdots + N(N-1)a_Nx^{N-2}$$

$$f'''(x) = 6a_3 + 24a_4x + \cdots + N(N-1)(N-2)a_Nx^{N-3}$$

\vdots
Easiest Point $x$ for Evaluating $f(x)$

Recall the equations that will be used to evaluate $f(x)$ or one of its derivatives:

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_N x^N$$
$$f'(x) = a_1 + 2a_2 x + 3a_3 x^2 + \cdots + N a_N x^{N-1}$$
$$f''(x) = 2a_2 + 6a_3 x + 12a_4 x^2 + \cdots + N(N-1) a_N x^{N-2}$$

These are most easily evaluated at $x = 0$ because the above equations reduce to:

$$f(0) = a_0$$
$$f'(0) = a_1$$
$$f''(0) = 2a_2$$
$$f'''(0) = 6a_3$$
$$\vdots$$

How can this be made to happen every time?

How to Make Any Point Easy

Suppose it is desired to evaluate $f(x)$ or one of its derivatives at the general point $x = x_{fd}$.

To do this, translate the x-axis by $x_{fd}$ before fitting the polynomial.

Recall that the finite-difference coefficients depend only on the relative position of the points. An offset will not affect their values.

Now write the polynomial at each shifted point.

$$f(\tilde{x}_1) = a_0 + a_1 \tilde{x}_1 + a_2 \tilde{x}_1^2 + \cdots + a_N \tilde{x}_1^N$$
$$f(\tilde{x}_2) = a_0 + a_1 \tilde{x}_2 + a_2 \tilde{x}_2^2 + \cdots + a_N \tilde{x}_2^N$$
$$\vdots$$
$$f(\tilde{x}_{N+1}) = a_0 + a_1 \tilde{x}_{N+1} + a_2 \tilde{x}_{N+1}^2 + \cdots + a_N \tilde{x}_{N+1}^N$$

In the shifted coordinate system, the finite-difference is being evaluated at $\tilde{x} = 0$.

$$f(0) = a_0$$
$$f'(0) = a_1$$
$$f''(0) = 2a_2$$
$$f'''(0) = 6a_3$$
$$\vdots$$
Four-Step Procedure to Derive Finite-Difference Approximations

**Step 1** – Identify set of points \(x_1, x_2, ..., x_N\) from which to derive a finite-difference approximation.

**Step 2** – Shift coordinates so that \(\bar{x} = 0\) corresponds to where the function or one of its derivatives is being calculated.

\[
\bar{x}_i = x_i - x_0
\]

**Step 3** – Fit shifted points to a polynomial.

\[
f(\bar{x}) = a_0 + a_1\bar{x} + a_2\bar{x}^2 + \cdots + a_N\bar{x}^N
\]

**Step 4** – Write finite-difference approximation directly from one of the derivatives of the polynomial.

\[
f(0) = a_0
\]
\[
f'(0) = a_1
\]
\[
f''(0) = 2a_2
\]
\[
\vdots
\]

Step 1 – Choose \(x\) Coordinates

Identify the \(x\)-coordinates of the points from which a derivative is to be approximated.

Store these values of \(x\) in a column vector.

\[
[x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N+1} \end{bmatrix}
\]
Step 2 – Shift x-Axis

Shift the function across the x-axis until $\tilde{x} = 0$ corresponds to the point where the derivative is to be approximated.

$$\frac{d^a}{d\tilde{x}^a} f(x = x_{fd}) = \frac{d^a}{d\tilde{x}^a} f(\tilde{x} = 0)$$

Subtract $x_{fd}$ from the column vector $[x]$ to shift the coordinates.

$$[\tilde{x}] = [x] - x_{fd} = \begin{bmatrix} x_1 - x_{fd} \\ x_2 - x_{fd} \\ \vdots \\ x_{N+1} - x_{fd} \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_{N+1} \end{bmatrix}$$

Step 3 – Fit Points to Polynomial (1 of 3)

Use the column vector $[\tilde{x}]$ to build matrix $[\tilde{X}]$.

$$[\tilde{X}] = \begin{bmatrix} [\tilde{x}]^0 & [\tilde{x}]^1 & \cdots & [\tilde{x}]^N \end{bmatrix} = \begin{bmatrix} 1 & \tilde{x}_1 & \cdots & \tilde{x}_1^N \\ 1 & \tilde{x}_2 & \cdots & \tilde{x}_2^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{x}_{N+1} & \cdots & \tilde{x}_{N+1}^N \end{bmatrix} \cdot \begin{bmatrix} x_1 - x_{fd} & \cdots & (x_1 - x_{fd})^N \\ x_2 - x_{fd} & \cdots & (x_2 - x_{fd})^N \\ \vdots & \ddots & \vdots \\ x_{N+1} - x_{fd} & \cdots & (x_{N+1} - x_{fd})^N \end{bmatrix}$$

Insert 1's instead of $[\tilde{x}]^0$. 
Step 3 – Fit Points to Polynomial (2 of 3)

Invert the matrix \([\tilde{X}]\).

\[
[\tilde{Y}] = [\tilde{X}]^{-1} = \begin{bmatrix}
\tilde{y}_{11} & \tilde{y}_{12} & \cdots & \tilde{y}_{1,N+1} \\
\tilde{y}_{21} & \tilde{y}_{22} & \cdots & \tilde{y}_{2,N+1} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{y}_{N+1,1} & \tilde{y}_{N+1,2} & \cdots & \tilde{y}_{N+1,N+1}
\end{bmatrix}
\]

Step 3 – Fit Points to Polynomial (3 of 3)

Calculate the polynomial coefficients.

\[
[a] = [\tilde{X}]^{-1} [f] = [\tilde{Y}][f] \rightarrow \begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
\vdots \\
a_N
\end{bmatrix} = \begin{bmatrix}
\tilde{y}_{11} & \tilde{y}_{12} & \cdots & \tilde{y}_{1,N+1} \\
\tilde{y}_{21} & \tilde{y}_{22} & \cdots & \tilde{y}_{2,N+1} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{y}_{N+1,1} & \tilde{y}_{N+1,2} & \cdots & \tilde{y}_{N+1,N+1}
\end{bmatrix} \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
\vdots \\
 f_{N+1}
\end{bmatrix}
\]

\[
a_0 = \tilde{y}_{11} f_1 + \tilde{y}_{12} f_2 + \tilde{y}_{13} f_3 + \cdots + \tilde{y}_{1,N+1} f_{N+1}
\]
\[
a_1 = \tilde{y}_{21} f_1 + \tilde{y}_{22} f_2 + \tilde{y}_{23} f_3 + \cdots + \tilde{y}_{2,N+1} f_{N+1}
\]
\[
a_2 = \tilde{y}_{31} f_1 + \tilde{y}_{32} f_2 + \tilde{y}_{33} f_3 + \cdots + \tilde{y}_{3,N+1} f_{N+1}
\]
\vdots
\[
a_N = \tilde{y}_{N+1,1} f_1 + \tilde{y}_{N+1,2} f_2 + \tilde{y}_{N+1,3} f_3 + \cdots + \tilde{y}_{N+1,N+1} f_{N+1}
\]

At this point, \(f_1\) to \(f_{N+1}\) will be symbolic.
Step 4 – Write Finite-Difference Approximation

Recall how the function or one of it’s derivatives is interpolated given the polynomial...

\[
\begin{align*}
    f(\tilde{x} = 0) &= a_0 \\
    \frac{d}{dx} f(\tilde{x} = 0) &= a_1 \\
    \frac{d^2}{dx^2} f(\tilde{x} = 0) &= 2a_2 \\
    \vdots \\
    f(\tilde{x} = 0) &= a_t = \tilde{y}_{1,t}f_1 + \tilde{y}_{1,2}f_2 + \tilde{y}_{1,3}f_3 + \cdots + \tilde{y}_{1,N+1}f_{N+1} \\
    \frac{d}{dx} f(\tilde{x} = 0) &= a_t = \tilde{y}_{2,t}f_1 + \tilde{y}_{2,2}f_2 + \tilde{y}_{2,3}f_3 + \cdots + \tilde{y}_{2,N+1}f_{N+1} \\
    \frac{d^2}{dx^2} f(\tilde{x} = 0) &= 2a_2 = \tilde{y}_{3,t}f_1 + \tilde{y}_{3,2}f_2 + \tilde{y}_{3,3}f_3 + \cdots + \tilde{y}_{3,N+1}f_{N+1} \\
    \vdots \\
    a_n = \tilde{y}_{N+1,t}f_1 + \tilde{y}_{N+1,2}f_2 + \tilde{y}_{N+1,3}f_3 + \cdots + \tilde{y}_{N+1,N+1}f_{N+1}
\end{align*}
\]

The rows of \( \tilde{Y} \) are essentially the finite-difference coefficients.
Example #1

Derive first-order and second-order finite-difference approximations that span across three evenly spaced points. The approximations should be evaluated at the midpoint.

\[
\begin{bmatrix}
  -h \\
  0 \\
  h
\end{bmatrix}
= \begin{bmatrix}
  \tilde{x}_0 \\
  \tilde{x}_1 \\
  \tilde{x}_2
\end{bmatrix}
= \begin{bmatrix}
  0 & -h & h^2 \\
  1 & 0 & 0 \\
  1 & h & h^2
\end{bmatrix}
\begin{bmatrix}
  \tilde{y}_0 \\
  \tilde{y}_1 \\
  \tilde{y}_2
\end{bmatrix}
= \begin{bmatrix}
  0 & 1 & 0 \\
  -1/2h & 0 & 1/2h \\
  1/2h^2 & -1/2h & 1/2h^2
\end{bmatrix}
\]

\[
a_0 = \begin{bmatrix}
  0 & 1 & 0 \end{bmatrix}
\]

\[
a_1 = \begin{bmatrix}
  -1/2h & 0 & 1/2h \end{bmatrix}
\]

\[
a_2 = \begin{bmatrix}
  1/2h^2 & -1/2h & 1/2h^2 \end{bmatrix}
\]

\[
f(x_i) = a_0 f_i
\]

\[
\frac{df(x_i)}{dx} = a_1 = \frac{f_i - f_{i+1}}{2h}
\]

\[
\frac{d^2f(x_i)}{dx^2} = 2a_2 = \frac{f_i - 2f_{i+1} + f_{i+2}}{h^2}
\]

Example #2

Derive first-order and second-order finite-difference approximations that span across three evenly spaced points. The approximations should be evaluated at the first point.

\[
\begin{bmatrix}
  0 \\
  h \\
  2h
\end{bmatrix}
= \begin{bmatrix}
  \tilde{x}_0 \\
  \tilde{x}_1 \\
  \tilde{x}_2
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 \\
  1 & h & h^2 \\
  1 & 2h & (2h)^2
\end{bmatrix}
\begin{bmatrix}
  \tilde{y}_0 \\
  \tilde{y}_1 \\
  \tilde{y}_2
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 \\
  -3/(2h) & 2h & -1/2h \\
  1/2h^2 & -1/2h & 1/(2h)^2
\end{bmatrix}
\]

\[
a_0 = \begin{bmatrix}
  1 & 0 & 0 \end{bmatrix}
\]

\[
a_1 = \begin{bmatrix}
  -3/(2h) & 2h & -1/2h \end{bmatrix}
\]

\[
a_2 = \begin{bmatrix}
  1/2h^2 & -1/2h & 1/(2h)^2 \end{bmatrix}
\]

\[
f(x_i) = a_0 f_i
\]

\[
\frac{df(x_i)}{dx} = a_1 = \frac{-1.5f_i + 2f_{i+1} - 0.5f_{i+2}}{h}
\]

\[
\frac{d^2f(x_i)}{dx^2} = 2a_2 = \frac{f_i - 2f_{i+1} + f_{i+2}}{h^2}
\]
Example #3 – Higher Order Accuracy (1 of 2)

Let's evaluate some derivatives at the midpoint of four discrete points.

$$[x] = \begin{bmatrix} -3h/2 \\ -h/2 \\ h/2 \\ 3h/2 \end{bmatrix}$$

$$[\hat{x}] = [x]^T [\hat{x}]^T [\hat{x}]^T [\hat{x}]^T = \begin{bmatrix} 1 & -3h/2 & -h/2 & 3h/2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 2h & -h & h \end{bmatrix}$$

$$[y] = [\hat{x}]^{-1}$$

Example #3 – Higher Order Accuracy (2 of 2)

The coefficients are then

$$[a_0] = \begin{bmatrix} 1/16 & 9/16 & 9/16 & 1/16 \\ 1/24h & 8h/24h & 8h/24h & 1/24h \\ 1/4h^2 & 1/4h^2 & 1/4h^2 & 1/4h^2 \\ 1/6h^3 & 1/6h^3 & 1/6h^3 & 1/6h^3 \end{bmatrix}$$

$$[f_1] = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$a_0 = \frac{1}{16} f_0 + \frac{9}{16} f_1 + \frac{9}{16} f_2 + \frac{1}{16} f_3$$

$$a_1 = \frac{1}{24h} f_0 - \frac{9}{8h} f_1 + \frac{9}{8h} f_2 - \frac{1}{24h} f_3$$

$$a_2 = \frac{1}{4h^2} f_0 - \frac{1}{4h^2} f_1 - \frac{1}{4h^2} f_2 + \frac{1}{4h^2} f_3$$

$$a_3 = \frac{1}{6h^3} f_0 + \frac{1}{2h^3} f_1 - \frac{1}{2h^3} f_2 + \frac{1}{6h^3} f_3$$

$$f'(x_{1/2}) = a_0 = \frac{f_0 - 9f_1 + 9f_2 - f_3}{16}$$

$$\frac{df}{dx}(x_{1/2}) = a_1 = \frac{f_0 - 27f_1 + 27f_2 - f_3}{24h}$$

$$\frac{d^2 f}{dx^2}(x_{1/2}) = 2a_3 = \frac{f_0 - f_2 - f_1 + f_3}{2(\Delta x)^2}$$