Quadrature weights

\[ \text{d}V \rightarrow \text{Vandermonde matrix with } \text{basis evaluated at nodes} \]

\[ \text{w-weights} \]

integrals of basis functions
Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo
Error, Accuracy and Convergence
Floating Point
Modeling the World with Arrays
The World in a Vector
What can Matrices Do?
Graphs
Sparsity
Norms and Errors
The ‘Undo’ Button for Linear Operations: LU
Repeating Linear Operations:
Eigenvalues and Steady States
Eigenvalues: Applications
Approximate Undo: SVD and Least Squares
SVD: Applications
Solving Funny-Shaped Linear Systems
Data Fitting
Norms and Condition Numbers
Low-Rank Approximation
Interpolation
Iteration and Convergence
Solving One Equation
Solving Many Equations
Finding the Best: Optimization in 1D
Optimization in $n$ Dimensions
Solving Nonlinear Equations

What is the goal here?

given \( f \)

\[ f(x) = 0 \]

\[ f(x) = g \]

\[ f(x) = f(x) - g \]

Solve \( \tilde{f}(x) = 0 \)
Bisection Method

Assume continuous function $f$ has a zero on the interval $[a, b]$ and

$$\text{sign}(f(a)) = -\text{sign}(f(b)).$$

Perform binary search: check sign of $f((a + b)/2)$ and define new search interval so that ends have opposite sign.

**Demo:** Bisection Method

What’s the rate of convergence? What’s the constant?
Newton’s Method

Derive Newton’s method.

\[ f(x) = 0 \]

\[ f_k(x) = 0 \]

\[ 0 = f(x_k) + f'(x_k)(x-x_k) \]
0 = f(x_k) + f'(x_k) x - f'(x_k) x_k

x = \frac{-f(x_k)}{f'(x_k)} + x_k

x = x_k - \frac{f(x_k)}{f'(x_k)}

\uparrow

x_{k+1}
Demo: Newton’s method
Demo: Convergence of Newton’s Method

What are some drawbacks of Newton?

may not converge
stick at local min.
or move far away from last guess
need derivative
Secant Method

What would Newton without the use of the derivative look like?

\[ f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \]

And \( x_k \approx x_{k-1} \)
\[ e_k = \text{error at the } k \text{th step} \]

Second:

\[
\lim_{k \to \infty} \frac{e_{k+1}}{e_k e_{k-1}} = c
\]

Side Note:

\[
\lim_{k \to \infty} \frac{e_{k+1}}{e_k} < c
\]
Secant Method Drawbacks

What are some **drawbacks** of Secant?

- 2 starting guesses
- convergence problem Newton

**Demo:** Secant Method

**In-class activity:** Secant Method

+ no derivative
Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo
Error, Accuracy and Convergence
Floating Point
Modeling the World with Arrays
  The World in a Vector
  What can Matrices Do?
  Graphs
  Sparsity
Norms and Errors
The ‘Undo’ Button for Linear Operations: LU
Repeating Linear Operations:
  Eigenvalues and Steady States
  Eigenvalues: Applications
Approximate Undo: SVD and Least Squares
SVD: Applications
  Solving Funny-Shaped Linear Systems
  Data Fitting
  Norms and Condition Numbers
  Low-Rank Approximation
Interpolation
Iteration and Convergence
Solving One Equation
  Solving Many Equations
Finding the Best: Optimization in 1D
  Optimization in \( n \) Dimensions
Solving Nonlinear Equations

What is the goal here?

\[ f(x) = 0 \]

\[ f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \]

\[ f(x) = f(x) - \bar{f} \]
Newton’s method

What does Newton’s method look like in $n$ dimensions?

\[ J_f(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} \]

\[ x_{k+1} = x_k - J_f(x_k)^{-1} f(x_k) \]