Interpolation
Differentiation
Integration
“Quadrature”
Convergence
\[ f(x) = a_0 \phi_0(x) + \ldots + a_n \phi_n(x) \]

\[ f'(x) \approx f'_n(x) = a_0 \phi'_0(x) + \ldots + a_n \phi'_n(x) \]

\[ f'(x) = [\phi'_0(x), \ldots, \phi'_n(x)] \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} \]
Given that \( f(x) = \sqrt{x^2 + 1} \), find \( f'(x) \).

\[
\begin{align*}
f'(x) &= \frac{d}{dx} \left( x^2 + 1 \right)^{1/2} \\
&= \frac{1}{2} \left( x^2 + 1 \right)^{-1/2} \cdot 2x \\
&= \frac{x}{\sqrt{x^2 + 1}}
\end{align*}
\]
How could you find coefficients of the derivative in the original basis $\varphi_i$?

Give a matrix that finds the second derivative.

$V'V'V^{-1}$ → Differentiation matrix

$y' = V'V^{-1}y$

Approximation of first order
degree \( n-1 \) polynomial at \( n \)-points

\[ V' \quad V' \quad V' \quad y \]

coefficients of \( f' \) in diff. basis

coeffs of \( f' \) in orig. basis

value of \( f' \) at each point \( x, \ldots, x_n \)
Demo: Taking derivatives with Vandermonde matrices
Finite Difference Formulas

It is possible to use the process above to find ‘canned’ formulas for taking derivatives. Suppose we use three points equispaced points \((x - h, x, x + h)\) for interpolation (i.e. a degree-2 polynomial).

- What is the resulting differentiation matrix?
- What does it tell us for the middle point?

\[
V = \begin{bmatrix}
1 & x-h & (x-h)^2 \\
1 & x & x^2 \\
1 & x+h & (x+h)^2 \\
\end{bmatrix} \quad V' = \begin{bmatrix}
0 & 1 & 2(x-h) \\
0 & 2x & 2 \\
0 & 2(x+h) & 2 \\
\end{bmatrix}
\]

\[
D = V^{-1}V' = \begin{bmatrix}
\frac{1}{2h} & 0 & \frac{1}{2h} \\
0 & \frac{1}{h} & 0 \\
\frac{1}{2h} & 0 & \frac{1}{2h} \\
\end{bmatrix}
\]
$$D \cdot V = V'$$

$$
\begin{bmatrix}
\frac{-1}{2h} & 0 & \frac{1}{2h} \\
\frac{1}{2h} & 0 & \frac{-1}{2h}
\end{bmatrix}
\begin{bmatrix}
1 \\
1 & x & x^2 \\
1 & x & (x+h)^2
\end{bmatrix}
\begin{bmatrix}
0 \\
2x \\
\frac{2xh + 2xh}{2h} + \frac{(x-h)^2}{2h} + \frac{(x+h)^2}{2h}
\end{bmatrix}
$$
\[ D \cdot \begin{bmatrix} f(x+h) \\ f(x) \\ f(x-h) \end{bmatrix} = \begin{bmatrix} \frac{f(x+h) - f(x-h)}{2h} \\ \vdots \\ O(h^2) \end{bmatrix} \]

\[ D_{15} \cdot \begin{bmatrix} \frac{1}{15} & \frac{1}{15} \end{bmatrix} = \begin{bmatrix} \frac{1}{15} & \frac{1}{15} \end{bmatrix} \]
Can we use a similar process to compute (approximate) integrals of a function \( f \)?

\[
\hat{f}(x) = a_0, a_1(x) + \ldots + a_n(x)
\]

\[
\int_a^b f(x) \, dx = \int_a^b f(x) \, dx = a_0 \int_a^b 1 \, dx + \ldots
\]

\[
W = \begin{bmatrix}
\int_a^b g_1(x) \\
\int_a^b g_2(x) \\
\vdots \\
\int_a^b g_n(x)
\end{bmatrix}
\]

\[
W^T \cdot a \approx \text{vector of weights}
\]

\[
V_{a \approx y}
\]
Example: Building a Quadrature Rule

**Demo:** Computing the Weights in Simpson’s Rule

Suppose we know

\[ f(x_0) = 2 \quad f(x_1) = 0 \quad f(x_2) = 3 \]

\[
x_0 = 1 \quad \frac{x_1}{2} \quad x_2 = 1
\]

How can we find an approximate integral? \[ \int_0^1 f(x) \, dx \]

\[
\omega = \begin{bmatrix}
\int_0^1 f(x) \, dx \\
\int_1^2 f(x) \, dx \\
\int_2^3 f(x) \, dx
\end{bmatrix} = \begin{bmatrix}
\frac{1}{2} \\
\frac{1}{3}
\end{bmatrix}
\]
\[ V = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_0 & 0 & 0 \\ x_1 & x_2 & x_3 \end{bmatrix} \]

\[ a = V^{-1} \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ \cdot \\ \cdot \end{bmatrix} \]

\[ \text{integral} = w^T \cdot a = w^T V^T y \]

\[ \text{weight}(x) \]
Facts about Quadrature

What does Simpson’s rule look like on $[0, 1/2]$?

What does Simpson’s rule look like on $[5, 6]$?

How accurate is Simpson’s rule with $n$ points and functions?
Accuracy with $n$ points/factors

Interpolation error $O(h^n)$

Integration error $O(h^{n+2})$

Differential error $O(h^n)$
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
What is linear convergence? quadratic convergence?
About Convergence Rates

**Demo:** Rates of Convergence

Characterize linear, quadratic convergence in terms of the ‘number of accurate digits’.
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?
Bisection Method

Assume there is a zero on the interval \([a, b]\) and that \(f\) is continuous, perform binary search.

**Demo:** Bisection Method

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

Derive Newton’s method.
**Demo:** Newton’s method

**Demo:** Convergence of Newton’s Method

What are some **drawbacks** of Newton?
Secant Method

What would Newton without the use of the derivative look like?
Demo: Secant Method
In-class activity: Nonlinear equations in 1D
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
What is the goal here?
Newton’s method

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

Set up Newton’s method to find a root of

\[
f(x, y) = \begin{pmatrix} x + 2y - 2 \\ x^2 + 4y^2 - 4 \end{pmatrix}.
\]

Demo: Newton’s method in \( n \) dimensions
Secant in \( n \) dimensions?

What would the secant method look like in \( n \) dimensions?
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
State the problem.
| What are some potential problems in optimization? |
Optimization: What is a solution?

How can we tell that we have a (at least local) minimum? (Remember calculus!)
Newton’s Method

Let’s steal the idea from Newton’s method for equation solving: Build a simple version of f and minimize that.
Demo: Newton’s method in 1D
In-class activity: Optimization Methods
Golden Section Search

Would like a method like bisection, but for optimization. In general: No invariant that can be preserved.
Need *extra assumption*.
**Demo:** Golden Section Search Proportions
Optimization in $n$ dimensions: What is a solution?

How can we tell that we have a (at least local) minimum? (Remember calculus!)
Steepest Descent

Given a scalar function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at a point $x$, which way is down?
Demo: Steepest Descent
Newton’s method ($nD$)

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

**Demo:** Nelder-Mead Method
What if the $f$ to be minimized is actually a 2-norm?

$$f(x) = \| r(x) \|_2, \quad r(x) = y - f(x)$$

**Demo:** Gauss-Newton