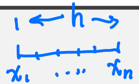
Differentiation Matrices

How can numerical differentiation be cast as a matrix-vector operation?



$$f(x) \approx p_n(x) = \sum \alpha_i \varphi_i(x)$$

$$\vec{\alpha} = (\alpha_1 \cdots \alpha_n) = \underline{V^{-1} f(\vec{x})}, \quad \vec{x} = (x_1 \cdots x_n)$$

$$f'(x) \approx p_n'(x) = \sum \alpha_i \varphi_i'(x)$$

$$V' = \begin{bmatrix} \varphi_1'(x_1) & \cdots & \varphi_n'(x_1) \\ \vdots & & \vdots \\ \varphi_1'(x_n) & \cdots & \varphi_n'(x_n) \end{bmatrix} \Rightarrow f'(\vec{x}) \approx V' V^{-1} f(\vec{x})$$

$$D = V' V^{-1}$$ differentiation matrix

Demo: Taking Derivatives with Vandermonde Matrices [cleared] (Build D)

Finite Difference Formulas from Diff. Matrices

How do the rows of a differentiation matrix relate to FD formulas?

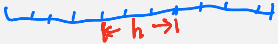
$$D = (d_{ij})_{i,j=1}^{n} \Rightarrow f'(x_i) \approx \sum_j d_{ij} f(x_j)$$

e.g. $n=3$.

1st : $f'(x_1) \approx d_{11} f(x_1) + d_{12} f(x_2) + d_{13} f(x_3)$

2nd : $f'(x_2) \approx d_{21} f(x_1) + d_{22} f(x_2) + d_{23} f(x_3)$

Assume a large equispaced grid and 3 nodes w/same spacing. How to use?



Finite Differences: via Taylor

forward difference.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

first order.

$$f(x+h) = f(x) + f'(x)h + f''(x) \cdot \frac{h^2}{2} + \cdots$$

$$\Rightarrow \quad \frac{f(x+h) - f(x)}{h} = f'(x) + f''(x) \cdot \frac{h}{2} + \cdots$$

More Finite Difference Rules

Similarly:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

(Centered differences)

Can also take higher order derivatives:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

Can find these by trying to match Taylor terms.
Alternative: Use linear algebra with interpolate-then-differentiate to find FD formulas.

Demo: Finite Differences vs Noise [cleared]
Demo: Floating point vs Finite Differences [cleared]

Richardson Extrapolation

Deriving high-order methods is hard work. Can I just do multiple low-order approximations (with different h and get a high-order one out?

Suppose we have $F = \tilde{F}(h) + O(h^p)$ and $\tilde{F}(h_1)$ *and* $\tilde{F}(h_2)$.

$$F = \tilde{F}(h) + a h^p + O(h^q) \quad (q = p+1)$$

typically but not always

$$F = \alpha \tilde{F}(h_1) + \beta \tilde{F}(h_2) + O(h^q)$$

Need:

- $\alpha a h_1^p + \beta a h_2^p = 0$
- $\alpha + \beta = 1 \rightsquigarrow \beta = 1 - \alpha$

$$\alpha (h_1^p - h_2^p) + 1 h_2^p = 0 \Leftrightarrow \alpha = \frac{-h_2^p}{h_1^p - h_2^p}$$

Richardson Extrapolation: Observations,

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} + O(h^1)$$

What are α and β for a first-order (e.g. finite-difference) method if we choose $h_2 = h_1/2$?

$p=1$

$$\alpha = \frac{-h_2^p}{h_1^p - h_2^p} = \frac{-\frac{1}{2}}{1-\frac{1}{2}} = -1 \qquad \beta = 1 - \alpha = 2$$

Demo: Richardson with Finite Differences [cleared]

Romberg Integration

Can this be used to get *even higher order* accuracy?

h

$h/2$

$h/4$

$h/8$

exponential cost increase

1st order

2nd order

3rd order

"nested Richardson for quadrature"

=

Romberg integration

In-Class Activity: Differentiation and Quadrature

In-class activity: Differentiation and Quadrature

Outline

What can we solve already?

- Linear Systems: yes
- Nonlinear systems: yes
- Systems with derivatives: no

Some Applications

IVPs	BVPs
▶ Population dynamics $y_1' = y_1(\alpha_1 - \beta_1 y_2)$ (prey) $y_2' = y_2(-\alpha_2 + \beta_2 y_1)$ (predator) ▶ chemical reactions ▶ equations of motion	▶ bridge load ▶ pollutant concentration (steady state) ▶ temperature (steady state) ▶ waves (time-harmonic)

Initial Value Problems: Problem Statement

Want: Function $\boldsymbol{y} : [0, T] \to \mathbb{R}^n$ so that

- $\boldsymbol{y}^{(k)}(t) = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{y}'', \ldots, \boldsymbol{y}^{(k-1)})$ (*explicit*), or
- $\boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{y}'', \ldots, \boldsymbol{y}^{(k)}) = 0$ (*implicit*)

are called explicit/implicit *kth-order ordinary differential equations* (*ODEs*). Give a simple example.

$y'(t) = \alpha y \quad \Leftarrow \quad y = c \cdot e^{\alpha t}$

Not uniquely solvable on its own. What else is needed?

Initial conditions.

$y(0) = y_0.$

$\#\text{ICs} + \#\text{ODEs} = k$

Reducing ODEs to First-Order Form

A kth order ODE can always be reduced to first order. Do this in this example:

$$y''(t) = f(y)$$

Let $\begin{cases} y_1 = y \\ y_2 = y' \end{cases} \Rightarrow \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} y_2 \\ f(y_1) \end{pmatrix}$

$\Rightarrow$ only care about 1st order wlog