Lecture 10

Continuing w/ derivatives

technique from last time is
to approximate the derivative w/
a straight line.
So this is a first-order approx.
+ the question is whether we
can do better w/ higher-order
approximations.
This did well for integration,
maybe for derivatives also?

Consider fitting a quadratic
to function of interest that
we would like to differentiate

Suppose we want derivative @ $x=0$

As in derivation for Simpson rule, we require quadratic ax^2+bx+c to fit function & so

$ah^2-bh+c=f(-h)$, $c=f(0)$,

$ah^2+bh+c=f(h)$

could then solve for a, b, c, but actually no need to.

Given $y=ax^2+bx+c$,

$$\left.\frac{dy}{dx}\right|_{x=0} = [2ax+b]\Big|_{x=0}$$

$$= b$$

so we just need b & can get it by subtracting 1st from 3rd

$f(h)-f(-h) = 2bh$

$\Rightarrow b = [f(h)-f(-h)]/2h$

But this is just the central difference!

could then repeat this for other values of x & we would find

$$\frac{df}{dx} \simeq \frac{f(x+h)-f(x-h)}{2h}$$

so it's really just central difference...

However, going to higher orders helps.

There is a distinction between even & odd orders.

sample points need to be symmetric about x

For odd orders, sample points fall halfway, e.g., ..., $-3/2\,h$, $-1/2\,h$, $\frac{1}{2}h$, $3/2\,h$, ...
(b/c even # of points for approx.)

For even orders, they fall at integer points, i.e., ..., $-2h$, $-h$, 0, h, $2h$, ...
(because odd # of points for approx.)

Bring up t higher-order-deriv.png

~~Each~~ nth approx. is exact if function is poly of degree n or less.

Order of error is given in table

Second derivative of a function

just apply first deriv approx. twice (central difference)

$$f'(x+h/2) \simeq \frac{f(x+h) - f(x)}{h}$$

$$f'(x-h/2) \simeq \frac{f(x) - f(x-h)}{h}$$

$$f''(x) = \{f'(x+h/2) - f'(x-h/2)\}/h$$

Substitute in

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

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

can use this to help solve second-order differential equations

How to estimate error?

Again use Taylor expansions as

$$f(x+h) = f(x) + h f'(x) + \tfrac{1}{2} h^2 f''(x) + \tfrac{1}{6} h^3 f'''(x) + \tfrac{1}{24} h^4 f''''(x) + \ldots$$

$$f(x-h) = f(x) - h f'(x) + \tfrac{1}{2} h^2 f''(x) - \tfrac{1}{6} h^3 f'''(x) + \tfrac{1}{24} h^4 f''''(x) + \ldots$$

Add & rearrange to find

6

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

$$- \frac{1}{12} h^2 f''''(x) + \ldots$$

1st term is approx., next is leading order error

$$\frac{1}{12} h^2 |f''''(x)|$$

rounding error is now (worst case)

$$\frac{4C\,|f(x)|}{h^2},$$ so total error is

$$\mathcal{E} = \frac{4C|f(x)|}{h^2} + \frac{1}{12} h^2 |f''''(x)|$$

take $\frac{d\mathcal{E}}{dh} = 0$ & get

$$h = \left(48C \left|\frac{f(x)}{f''''(x)}\right|\right)^{1/4}$$

Then optimal error size is

$$\varepsilon = \frac{1}{6} h^2 \left| f'''(x) \right|$$

$$= \left(\frac{4}{3} C \left| f(x) f'''(x) \right| \right)^{1/2}$$

So error $\varepsilon \simeq 10^{-8}$

which is about the same as forward & backward difference methods

---

partial derivatives: done in obvious way

$$\frac{\partial f}{\partial x} \simeq \frac{f(x+h/2, y) - f(x-h/2, y)}{h}$$

$$\frac{\partial f}{\partial y} \simeq \frac{f(x, y+h/2) - f(x, y-h/2)}{h}$$

Bring up 2-partial-approx.png

Taking derivatives of noisy data

Bring up 3-noisy-deriv.png

derivative of noisy data has noise amplified b/c of sharp transitions.

Ways around:

1) increase h to eliminate fluctuations
2) fit a curve + then differentiate a curve. (least squares fit)

Interpolation

Methods for analysis are similar to what we've used.

given value of f(x) @ a + b, what is value in between?

we can use linear interpolation..

Bring up 4-linear-interp.png

Slope of straight-line approx. is

$$m = \frac{f(b) - f(a)}{b - a}$$

distance y is $y = m(x-a)$

$$z = f(a)$$

$$\Rightarrow f(x) \simeq y + z = \frac{f(b) - f(a)}{b - a}(x-a) + f(a)$$

$$\Rightarrow f(x) \simeq \frac{(b-x)\, f(a) + (x-a)\, f(b)}{b-a}$$

one can also use this to extrapolate values outside interval (but don't go too far)

How accurate is formula?

Again resort to Taylor expansions

$$f(a) = f(x) + (a-x) f'(x) + \frac{1}{2}(a-x)^2 f''(x) + \dots$$

$$f(b) = f(x) + (b-x) f'(x) + \frac{1}{2}(b-x)^2 f''(x) + \dots$$

Substitute & find that

$$f(x) = \underbrace{\frac{(b-x)\, f(a) + (x-a)\, f(b)}{b-a}}_{\text{interp. formula}} + \underbrace{(a-x)(b-x) f''(x)}_{\text{leading order error}} + \dots$$

(gets smaller as [illegible])

If $f''(x)$ varies slowly, then largest error is in middle of interval

If width of interval is $h = b - a$, then being in the middle means that
$x - a = b - x = \frac{1}{2}h$ & leading order error is $\frac{1}{4} h^2 |f''(x)|$

worst-case error is then $O(h^2)$

Need not be too careful w/ rounding error inside interval b/c we have the sum of $f(a)$ & $f(b)$

moving on to Chapter 6

Solving linear equations $Ax = v$

More desirable to use Gaussian elimination than to invert matrix

Suppose

$$2w + x + 4y + z = -4$$
$$3w + 4x - y - z = 3$$
$$w - 4x + y + 5z = 9$$
$$2w - 2x + y + 3z = 7$$

Can rewrite as matrix eqn.

$$\begin{bmatrix} 2 & 1 & 4 & 1 \\ 3 & 4 & -1 & -1 \\ 1 & -4 & 1 & 5 \\ 2 & -2 & 1 & 3 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \\ 9 \\ 7 \end{bmatrix}$$

faster than calculating inverse is
to use Gaussian elimination

Notice that

1) We can multiply any row by a constant + equation / solution does not change

2) can take any linear combo of two equations to get another equation which is consistent w/ original solution.

So now we have some moves

How to solve our set?

1) Divide 1st row by top-left element:

$$\text{unchanged}\left\{\begin{bmatrix} 1 & 1/2 & 2 & 1/2 \\ \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} \end{bmatrix}\right. \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ - \\ - \\ - \end{bmatrix}$$

2) Now Subtract 3 times 1st row from 2nd,

giving

$$\begin{bmatrix} 1 & 1/2 & 2 & 1/2 \\ 0 & 5/2 & -7 & -5/2 \\ \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 9 \\ 4 \\ 7 \end{bmatrix}$$

3) Same trick for 3rd & 4th rows, which gives

$$\begin{bmatrix} 1 & 1/2 & 2 & 1/2 \\ 0 & 5/2 & -7 & -5/2 \\ 0 & -9/2 & 1 & 9/2 \\ 0 & -3 & -3 & 2 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 9 \\ 11 \\ 11 \end{bmatrix}$$

4) Now we repeat this algorithm for the second row on, giving

$$\begin{bmatrix} 1 & .5 & 2 & .5 \\ 0 & 1 & -2.8 & -1 \\ 0 & -4.5 & 1 & 4.5 \\ 0 & -3 & -3 & 2 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 3.6 \\ 11 \\ 11 \end{bmatrix}$$

5) repeat the same, giving

$$\begin{bmatrix} 1 & 0.5 & 2 & 0.5 \\ 0 & 1 & -2.8 & -1 \\ 0 & 0 & -13.6 & 0 \\ 0 & 0 & -11.4 & -1 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 3.6 \\ 27.2 \\ 21.8 \end{bmatrix}$$

6) final step:

$$\begin{bmatrix} 1 & .5 & 2 & .5 \\ 0 & 1 & -2.8 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 3.6 \\ -2 \\ 1 \end{bmatrix}$$

matrix is now upper triangular & we can solve by back substitution

Backsubstitution

Starting from

$$\begin{pmatrix} 1 & a_{01} & a_{02} & a_{03} \\ 0 & 1 & a_{12} & a_{13} \\ 0 & 0 & 1 & a_{23} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

this gives the equations

$$w + a_{01} x + a_{02} y + a_{03} z = v_0$$

$$x + a_{12} y + a_{13} z = v_1$$

$$y + a_{23} z = v_2$$

$$z = v_3$$

so start from last &

back substitute

Bring up 5 - gausselim.py

Pivoting

What if our system is

$$\begin{pmatrix} 0 & 1 & 4 & 1 \\ 3 & 4 & -1 & -1 \\ 1 & -4 & 1 & 5 \\ 2 & -2 & 1 & 3 \end{pmatrix} \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -4 \\ 3 \\ 9 \\ 7 \end{pmatrix}$$

How to handle? just "pivot", i.e., exchange a row w/ non-zero 1st entry & then proceed.

Best idea is to use partial pivoting, in which you find entry farthest from zero & swap it in. This will lead to smallest rounding error