Numerical Integration

Area under the curve

Single Integral

Trapezoidal Rule

fit linear function

Single Segment A=(f(a)+f(b))*(b-a)/2

Multiple Segments A=(f(xo)+2sum(f(xi))+f(xn))*(b-a)/2n

n is number of intervals- same width

increase accuracy.....increase intervals

Exception: -linear function -fluctuating function

as dx decreases, come closer to function

* inefficient but no limit on # of intervals

if non equal intervals, calculate separately and add

Simpson's 1/3 Rule

fit quadratic function

Single A=(f(x0)+4f(x1)+f(x2))*(b-a)/6

equidistant x1

Multiple A=(f(x0)+4sum_odd(f(xi))+2sum_even(f(xi))+f(xn))*(b-a)/3n

# of intervals is even

* even # of intervals

most popular bec accuracy is not that significant from 3/8 with less computation

Simpson's 3/8 Rule

fit cubic function

A= (f(x0)+3f(x1)+3f(x2)+f(x3))*(b-a)/8

1/3 rule is most widely used as computational efficiency it provides outweighs the accuracy provided by 3/8 rule

Trapezoidal rule can reach same accuracy of 3/8 rule by increasing number of intervals

* multiple of 3 # of intervals

Multiple Integral

Step 1 at y=0 find A **repeat**

Step 2 find A of A(y)

Tavg= A(A(y))/area or T=A(A(y))

Numerical Differentiation

Taylor series

f(xi+1)= f(xi)+f'(xi)h+f''(xi)h^2/2!+............+f^n(xi)h^n/n!

Exponential, you need infinite order because f^n(xi)=e^x which is never 0

First Forward difference

f'(xi)= (f(xi+1)-f(xi))/h

to increase accuracy, decrease h that will decrease the rest of Taylor series

First Backward difference

f'(xi)= (f(xi)-f(xi-1))/h

First Centered difference

f'(xi)= (f(xi+1)-f(xi-1))/2h

Higher Order

First Forward difference

f'(xi)= -f(xi+2)+4f(xi+1) -3f(xi) /(2h)

First Backward difference

f'(xi) = 3f(xi) -4f(xi-1) +f(xi-2) /(2h)

First Centered difference

f'(xi)= -f(x+2) +8f(xi+1)-8f(xi-1)+f(xi-2) /(12h)

Second Forward difference

f''(xi)= (f(xi+2)-2f(xi+1)+f(xi))/h^2

Second Backward difference

f''(xi)= (f(xi)-2f(xi-1)+f(xi-2))/h^2

Second Centered difference

f''(xi)= (f(xi+1)-2f(xi)+f(xi-1))/h^2

* more accurate

Lagrange

fit interpolated polynomial then differentiate- function that passes by all points

general method

f'(x)= pt1+pt2+pt3

pt1: 2x-xi-(xi+1)/ ((xi-1)-xi) ((xi-1)-(xi+1)) * f(xi-1)

pt2: 2x-(xi-1)-(xi+1)/ (xi-(xi-1)) (xi-(xi+1)) * f(xi)

pt3: 2x-(xi-1)-xi/ ((xi+1)-(xi-1)) ((xi+1)-xi) * f(xi+1)

Ordinary Differential Equations

Why solve numerically?

efficiency or cannot solve analytically

butterfly effect --sensitive to minor changes

chaotic systems --system sensitive to initial conditions but with predictable behavior

ODE with respect to 1 independent variable

PDE with respect to more than 1 independent variable

both can have many dependent variables

Euler's Method

f(xi+1)=f(xi)+ k1h

to decrase error, either decrease timestep (h) or take more slopes

* to decrease timestep (h), take into account round off error propagates and computational efficiency

assume slope constant in interval

Heun's Method

second order

yi+1=yi +kavg h

implicit method (yi+1 predictor) ---iterative

Midpoint Method

yi+1= yi + k2 h

Explicit Method

ymid= yi+ (k1 h)/2

Runge Kutte

generalized formula for methods

yi+1= yi + phi h

phi is weighted average of slopes

number of ks reflects order

there are infinite methods

fourth order Runge Kutte

yi+1 = yi + h/6 (k1+ 2k2+2k3+k4)

Ralston's Method

phi= 1/3 K1 + 2/3 k2

k2 calculated at 3/4th of interval

third order Runge Kutte

phi = (k1 + 4k2+ k3)/6

k2 is midway and k3 is at the end

systems of odes

Ordinary Differential Equations (cont)

need to look at each independently but solve simultanously
#intial conditions = # dependent variables