CS205 – Class 15

Covered in class: 3, 4, 5
Readings: 8.7, 9.1, 9.2

1. **Gaussian Quadrature** – for each subinterval $[x_i, x_{i+1}]$, use k specially spaced points to obtain a method that is exact on 2k-1 degree polynomials and thus has an order of accuracy of 2k
    a. $G = \sum \left(\frac{x_{i+1} - x_i}{2} \right) \left(f\left(\frac{x_i + x_{i+1}}{2} - \frac{x_{i+1} - x_i}{2\sqrt{3}} \right) + f\left(\frac{x_i + x_{i+1}}{2} + \frac{x_{i+1} - x_i}{2\sqrt{3}} \right) \right)$
        i. 2 points, piecewise cubic, exact for piecewise cubic functions, 4th order accurate
2. Can extend quadrature to **higher dimensions**
    a. One dimension $\int_a^b f(x)dx$ - subdivide $[a,b]$ into smaller intervals
    b. Two dimensions $\iint_A f(x,y)dA$ - subdivide A into rectangles or triangles
    c. Three dimensions $\iiint_V f(x,y,z)dV$ - subdivide V into boxes or tetrahedral
    d. **Monte Carlo methods** – usually used in higher dimensions
        i. Random or pseudo random numbers are used to generate sample points that are averaged and multiplied by the element "size" (e.g. length, area, volume)
        ii. Error decreases like $n^{-1/2}$ where n is the number of sample points
            1. 100 times more points are needed to gain one more digit of accuracy
            2. Slow convergence, but independent of the number of dimensions
            3. Not competitive for lower dimensional problems, but the only alternative for higher dimensional problems
3. **Richardson extrapolation** eliminate the leading order error term using 2 calculations.
    a. Start an integration scheme with some step size *h* whose value is I_h.
        i. This has some error associated $O(h^p)$
        ii. So we can relate it to the exact integration as $I_h = I_{exact} + O(h^p)$
    b. We can express the error more explicitly to get $I_h = a + bh^p + O(h^r)$
    c. Now write with a different step size say *qh* to get $I_{qh} = a + bq^p h^p + O(h^r)$
    d. Now by combining these two estimates we can get the order *p* error to drop out.
        i. $q^p I_h - I_{qh} = q^p a + bh^p q^p + O(h^r) - a - bq^p h^p - O(h^r) = (q^p - 1)a + O(h^r)$
        ii. Solving for *a* we get $a = \frac{q^p I_h - I_{qh}}{q^p - 1} + O(h^r)$
        iii. *a* is our new integration formula which.
    e. Usually use 2 successive grids I_h and $I_{h/2}$ i.e. *q=1/2*.

f. Not just for integrals, works for other types of equations too, e.g. differential equations

g. Need some level of smoothness and sufficient numbers of grid points.

4. **Finite differences** approximate derivatives $h \to 0$ and quantities $f(x), f'(x), f''(x), \ldots$ are $O(1)$ we have In general the Taylor expansion about x is $f(x+h) = \sum_{k=0}^{n} f^{(k)}(x)\frac{h^k}{k!} + O(h^{n+1})$. I.e. If we expand to n=2 we get $f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + O(h^3)$ get:

   a. **Taylor expansions** valid as

      i. Forward difference (1st order accurate) $f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$ which we get by starting with the Taylor expansion and

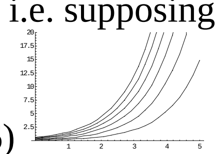
      ii. Backward difference (1st order accurate) $f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$

      iii. Central difference (2nd order accurate) $f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$

      iv. 2nd Derivative (2nd order accurate) $f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$

5. **Ordinary differential equations** (ODEs) system $\vec{y}' = \vec{f}(t, \vec{y})$, scalar $y' = f(t, y)$.

   a. Initial value problem $y' = y$ implies $dy / y = dt$, $\ln y - \ln y_o = t - t_o$, $y = y_o e^{t-t_o}$

      i. We obtain a family of solutions i.e. supposing t_0=0 and varying y_0 we get for (y_0=0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6) 

      ii. The specific solution depends on the initial condition $y_o = y(t_o)$

   b. **Higher order ode's** $y^{(n)} = f(t, y, y', y'', y''', \cdots, y^{(n-1)})$

i. Reduce to a first order system of the form

$$(y_1', y_2' \cdots, y_{n-1}', y_n') = (y_2, y_3, \cdots, y_n, f(t, y_1, \cdots y_n)) \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} y \\ y' \\ y'' \\ y''' \\ \vdots \\ y^{(n-2)} \\ y^{(n-1)} \end{pmatrix} = \begin{pmatrix} y \\ y'_1 \\ y'_2 \\ y'_3 \\ \vdots \\ y'_{n-2} \\ y'_{n-1} \end{pmatrix}$$ rewrite it as

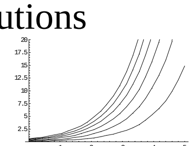
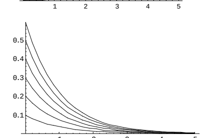
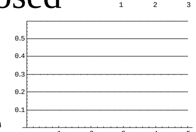
$$\begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \\ \vdots \\ y'_{n-1} \\ y'_n \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_n \\ f(t, y_1, y_2, ..., y_n) \end{pmatrix}$$

ii. Thus, we only need to consider first order systems

iii. <u>**Newton's 2nd Law**</u> F=ma is $a = x'' = F/m$ and which can be written as $\begin{pmatrix} x' \\ v' \end{pmatrix} = \begin{pmatrix} v \\ F(x, v)/m \end{pmatrix}$.

c. Model ODE Problems

i. Scalar ODE $y' = f(t, y)$ and the

1. linear model ODE is $y' = \lambda y$ which solution is $y = y_0 e^{\lambda(t - t_0)}$
2. Only three kinds of solutions
3. $\lambda > 0$ ill-posed unstable 
4. $\lambda < 0$ stable, well-posed 
5. $\lambda = 0$ linearly stable 

ii. Vector ODE $\vec{y}' = \vec{f}(t, \vec{y})$ and the linear model ODE is $\vec{y} = J\vec{y}$. Here is where it gets more interesting as the characterization of the ODE is dependent on the eigenvalues of the Jacobian matrix.