Stiff Systems of Ordinary Differential Equations

Larry Caretto

Mechanical Engineering 501A

Seminar in Engineering Analysis

November 22, 2017

California State University Northridge

Outline

- Midterm Results
- Review last class
  - Stability of numerical solutions
  - Step size variation for error control
  - Multistep methods with constant and variable step size
- Systems of equations
- Definition of stiff systems
- Algorithms for stiff systems
- Finding if system is stiff

California State University Northridge 2

Midterm Problem One

$$\frac{d^3y}{dx^3} + 3\frac{d^2y}{dx^2} + 3\frac{dy}{dx} + y = 8e^x - 2x - 1$$ y(0) = 1, y'(0) = 0, y(1) = 0

- Indicial equation, $\lambda^3 + 3\lambda^2 + 3\lambda + 1 = (\lambda + \lambda)^3 = 0$, has three identical roots of -1 so $y_H = (C_1 + C_2x + C_3x^2)e^{-x}$
- Plug trial $y_P = Ae^x + Bx + C$ into ODE to find A = 1, B = -2, and C = 5
- $y = (C_1 + C_2x + C_3x^2)e^{-x} + e^x - 2x + 5$
- $y(0) = 1 = C_1e^{-0} + e^0 - 2(0) + 5 = 1$

$C_1 = -5$

California State University Northridge 3

Midterm Problem One II

- $y' = -(C_1 + C_2x + C_3x^2)e^{-x} + e^x - 2 + (C_2 + 2C_3x)e^{-x} = 0 \; at \; x = 0$
- $-(C_1)e^{-0} + e^0 - 2 + C_2e^{-0} = 0$
- $C_2 = C_1 - 1 + 2 = C_1 + 1 = -5 + 1 = -4$
- Boundary condition that y(1) = 0 is solved to find $C_3 = 9 - 3e - e^2$
- Solution is

$$y = \left[-5 - 4x + \left(9 - 3e - e^2\right)x^2\right]e^{-x} + e^x - 2x + 5$$

California State University Northridge 4

Midterm Problem Two

- Solve using Laplace transforms [y(0) = 1 and y'(0) = -1] $$\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 2y = 10\sin t$$
- $s^2Y(s) - sy(0) - y'(0) + 3[sY(s) - y(0)] + 2Y(s) = 10\frac{1}{s^2+1^2}$ Solve for Y(s)

$$Y(s) = \frac{s+2}{(s+2)(s+1)} + \frac{10}{(s+2)(s+1)(s^2+1^2)}$$

- Rearrange/use partial fractions for Y(s)

$$Y(s) = \frac{A}{(s+2)} + \frac{B+1}{(s+1)} + \frac{Cs+D}{(s^2+1^2)}$$

California State University Northridge 5

Midterm Problem Two II

- Partial fractions results A = -2, B = 5, C = -3, D = 1 so Y(s) equation is $Y(s) = \frac{-2}{(s+2)} + \frac{6}{(s+1)} + \frac{-3s+1}{(s^2+1^2)}$
- Finding inverse transforms gives the final result
- $y(t) = -2e^{-2t} + 6e^{-t} - 3cos(t) + sin(t)$

California State University Northridge 6

Midterm Problem Three

- Rearrangement gives examples of Bessel's equation with $\nu = 2$ and $\nu = 0.5$

$$\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} + \left(\frac{x^2 - \upsilon^2}{x^2}\right)y = 0$$

- For integer $\nu = n = 2$, solution is $y = AJ_2(x) + BY_2(x)$; for non-integer $\nu = 0.5$, solution is $AJ_{0.5}(x) + BJ_{-0.5}(x)$
- Fitting boundary conditions gives first result as $3.226J_2(x) + 0.2247Y_2(x)$ and second as $1.138J_{0.5}(x) – 1.771J_{-0.5}(x)$

California State University Northridge 7

Systems of Equations

- Any problem with one or more ODEs of any order can be reduced to a system for first order ODEs
- In a previous lecture, we reduced a system of two second order equations to a system of four first order equations
- In this process the sum of the orders is constant
- Look at previous example

California State University Northridge 8

Reduction of Order Example

$$\frac{d^2y_1}{dt^2} + \frac{k_1 + k_2}{m_1}y_1 - \frac{k_2}{m_1}y_2 = 0 \qquad \frac{d^2y_2}{dt^2} - \frac{k_2}{m_2}y_1 + \frac{k_3 + k_2}{m_2}y_2 = 0$$

- Define variables $y_3 = dy_1/dt$ and $y_4 = dy_2/dt$
- Then $dy_3/dt = d^2y_1/dt^2$ and $dy_4/dt = d^2y_2/dt^2$
- Have four simultaneous first-order ODEs

$$\frac{dy_1}{dt} = y_3 = f_1(x,\mathbf{y}) \qquad \frac{dy_2}{dt} = y_4 = f_2(x,\mathbf{y})$$

$$\frac{dy_3}{dt} = -\frac{k_1 + k_2}{m_1}y_1 + \frac{k_2}{m_1}y_2 = f_3(x,\mathbf{y})$$

$$\frac{dy_4}{dt} = \frac{k_2}{m_2}y_1 - \frac{k_3 + k_2}{m_2}y_2 = f_4(x,\mathbf{y})$$

California State University Northridge 9

System of Equation Notation

- Write system as vector equation $\frac{d\mathbf{y}}{dx} = \mathbf{f}(x,\mathbf{y})$
- Individual components $\frac{dy_i}{dx} = f_i(x,\mathbf{y}) = f_i(x, y_1, y_2, \ldots, y_N)$
- Each f_i may depend on x and all y_j

$$\frac{dy_1}{dt} = y_3 = f_1(x,\mathbf{y}) \qquad \frac{dy_2}{dt} = y_4 = f_2(x,\mathbf{y})$$

$$\frac{dy_3}{dt} = -\frac{k_1 + k_2}{m_1}y_1 + \frac{k_2}{m_1}y_2 = f_3(x,\mathbf{y})$$

$$\frac{dy_4}{dt} = \frac{k_2}{m_2}y_1 - \frac{k_3 + k_2}{m_2}y_2 = f_4(x,\mathbf{y})$$

California State University Northridge 10

Solving Simultaneous ODEs

- Apply same algorithms used for single ODEs
  - Must apply each step and substep to all equations in system
  - Key is having consistent x and **y** values in determination of $f_i(x,\mathbf{y})$
  - All y_i values in **y** must be available at the same x point when computing the f_i
  - E.g., in Runge-Kutta we must evaluate k_1 for all equations before finding k_2

California State University Northridge 11

Runge-Kutta for ODE System

- $\mathbf{y}_{(n)}$ is vector of dependent variables at $x = x_n$
- $\mathbf{k}_{(1)}$, $\mathbf{k}_{(2)}$, $\mathbf{k}_{(3)}$, and $\mathbf{k}_{(4)}$, are vectors containing intermediate Runge-Kutta results
- **f** is a vector containing the derivatives
- $\mathbf{k}_{(1)} = h\mathbf{f} = h\mathbf{f}(x_n, \mathbf{y}_{(n)})$
- $\mathbf{k}_{(2)} = h\mathbf{f}(x_n + h/2, \mathbf{y}_{(n)} + \mathbf{k}_{(1)}/2)$
- $\mathbf{k}_{(3)} = h\mathbf{f}(x_n + h/2, \mathbf{y}_{(n)} + \mathbf{k}_{(2)}/2)$
- $\mathbf{k}_{(4)} = h\mathbf{f}(x_n + h, \mathbf{y}_{(n)} + \mathbf{k}_{(3)})$
- $\mathbf{y}_{(n+1)} = (\mathbf{k}_{(1)} + 2\mathbf{k}_{(2)} + 2\mathbf{k}_{(3)} + \mathbf{k}_{(4)})/6$

California State University Northridge 12

ODE System by RK4

- $dy/dx = -y + z$ and $dz/dx = y - z$ with $y(0) = 1$ and $z(0) = -1$ with $h = .1$
- $k_{(1)y} = h[-y + z] = 0.1[-1 + (-1)] = -.2$
- $k_{(1)z} = h[y - z] = 0.1[1 - (-1)] = .2$
- $k_{(2)y} = h[-(y+ k_{(1)y}/2) + z + k_{(1)z}/2] = 0.1[-(1 + -0.2/2) + (-1 + .2/2)] = -.18$
- $k_{(2)z} = h[(y+ k_{(1)y}/2) - (z + k_{(1)z}/2)] = 0.1[(1 + -0.2)/2 - (-1 + .2/2)] = .18$

California State University Northridge 13

ODE System by RK4 II

- $k_{(3)y} = h[-(y+ k_{(2)y}/2) + z + k_{(2)z}/2] = 0.1[-(1 + -0.18/2) + (-1 + .18/2)] = -.182$
- $k_{(3)z} = h[(y+ k_{(2)y}/2) - (z + k_{(2)z}/2)] = 0.1[(1 + -0.18)/2 - (-1 + .18/2)] = .182$
- $k_{(4)y} = h[-(y+ k_{(3)y}) + z + k_{(3)z}] = 0.1[-(1 + -0.182) + (-1 + .182)] = -.1636$
- $k_{(4)z} = h[(y+ k_{(3)y}) - (z + k_{(3)z})] = 0.1[(1 + -0.182) - (-1 + .182)] = .1636$

California State University Northridge 14

ODE System by RK4 III

- $y_{i+1} = y_i + (k_{(1)y} + 2k_{(2)y} + 2k_{(3)y} + k_{(4)y})/6 = 1 + [(-.2) + 2(-.18) + 2(-.182) + (-.1636)]/6 = .8187$
- $z_{i+1} = z_i + (k_{(1)z} + 2k_{(2)z} + 2k_{(3)z} + k_{(4)z})/6 = -1 + [(.2) + 2(.18) + 2(.182) + (.1636)]/6 = -.8187$
- Continue in this fashion until desired final x value is reached
- No x dependence for f in this example

California State University Northridge 15

Numerical Software for ODEs

- Usually written to solve a system of N equations, but will work for N = 1
- User has to code a subroutine or function to compute the **f** array
  - Input variables are x and **y**; **f** is output
  - Some codes have one dimensional parameter array to pass additional information from main program into the function that computes derivatives

California State University Northridge 16

Derivative Subroutine Example

- Visual Basic code for system of ODEs at right is shown below

$$\frac{dy_1}{dx} = -y_1 + \sqrt{y_2} + y_3 e^{2x}$$

$$\frac{dy_2}{dx} = -2y_1^2 \qquad \frac{dy_3}{dx} = -3y_1 y_2$$

```
Sub fsub( x as Double, y() as Double, _
  f() as Double )
  f(1) = -y(1) + Sqr(y(2)) + y(3)*Exp(2*x)
  f(2) = -2 * y(1)^2
  f(3) = -3 * y(1) * y(2)
End Sub
```

California State University Northridge 17

Derivative Subroutine Example

- Fortran code for system of ODEs at right is shown below

$$\frac{dy_1}{dx} = -y_1 + \sqrt{y_2} + y_3 e^{2x}$$

$$\frac{dy_2}{dx} = -2y_1^2 \qquad \frac{dy_3}{dx} = -3y_1 y_2$$

```
subroutine fsub( x, y, f )
  real(KIND=8) x, y(:), f(:)
  f(1) = -y(1) + sqrt(y(2)) +y(3)*exp(2*x)
  f(2) = -2 * y(1)**2
  f(3) = -3 * y(1) * y(2)
end subroutine fsub
```

California State University Northridge 18

Derivative Subroutine Example

- C++ code for system of ODEs at right is shown below

$$\frac{dy_1}{dx} = -y_1 + \sqrt{y_2} + y_3 e^{2x}$$

$$\frac{dy_2}{dx} = -2y_1^2 \qquad \frac{dy_3}{dx} = -3y_1 y_2$$

```
void fsub(double x, double y[], double f[])
{
 f[1] = -y[1] + sqrt(y[2]) + y[3]*exp(2*x);
 f[2] = -2 * y[1] * y[1];
 f[3] = -3 * y[1] * y[2];
}
```

California State University Northridge 19

Stiff Systems of Equations

- Several definitions
- Basic problem is that there are several length or time constants (eigenvalues) in the system
  - If one is negative and large in magnitude compared to others, this will set stability
  - Such terms quickly drop to zero and do not affect physical solution
  - However, they force small h for stability

California State University Northridge 20

Single Stiff Equation

- Nonhomogenous equation with different length (time) scales
- dy/dx = f(x,y) = -a[y – F(x)] + dF/dx
- $y = [y_0 - F(0)]e^{-ax} + F(x)$
  - Solution details on next two charts
- F(x) = cx + b: $y = [y_0 - b]\, e^{-ax} + cx + b$
  - $y/y_0 = (1 - b/y_0)e^{-ax} + cx/y_0 + b/y_0$
  - $y/y_0 = (1 - b/y_0)e^{-ax} + ax(c/y_0) + b/y_0$
  - $y/y_0 \to e^{-ax}$ as $x \to 0$ and $y/y_0 \to b/y_0 + (c/y_0)(ax)$ as $x \to \infty$

California State University Northridge 21

Stiff Equation Solution

- dy/dx = f(x,y) = -a[y – F(x)] + dF/dx solution from formula for a first-order ODE

$$\frac{dy}{dx} + f(x)y = g(x) \qquad f(x) = a \qquad g(x) = aF + \frac{dF}{dx}$$

$$y = e^{-\int f(x)dx}\left[C + \int\left(g(x)e^{\int f(x)dx}\right)dx\right]$$

$$y = e^{-\int adx}\left[C + \int\left(g(x)e^{\int adx}\right)dx\right]$$

$$y = e^{-ax}\left[C + \int g(x)e^{ax}dx\right]$$

California State University Northridge 22

Stiff Equation Solution II

$$\int g(x)e^{ax}dx = \int e^{ax}\left(aF + \frac{dF}{dx}\right)dx$$

$$= \int e^{ax}aFdx + \int e^{ax}dF = \cancel{\int e^{ax}aFdx}$$

$$+ e^{ax}F - \int \cancel{F(ae^{ax}dx)} = e^{ax}F$$

By parts
∫udF =
uF - ∫Fdu

$$y = e^{-ax}\left[C + \int g(x)e^{ax}dx\right] = e^{-ax}\left[C + e^{ax}F\right] = Ce^{-ax} + F(x)$$

$$C = y_0 - F(0) \quad so \quad y = \left[y_0 - F(0)\right]e^{-ax} + F(x)$$

California State University Northridge 23

Gear's Method

- Use implicit algorithms with solution by Newton's method
- General algorithm shown below
  - k is global order of the method
  - Coefficients: See table next slide

$$y_{n+1} = \gamma\beta h f_{n+1} + \sum_{j=0}^{k} \alpha_{-j} y_{n-j}$$

- Solving g(y) = 0 by Newton's method $\quad y^{(m+1)} = y^{(m)} - \dfrac{g\left(y^{(m)}\right)}{g'\left(y^{(m)}\right)}$

California State University Northridge 24

Gear's Method Coefficients

k	γ	β	α₀	α_{-1}	α_{-2}	α_{-3}	α_{-4}	α_{-5}
1	1	1	1					
2	1/3	2	4	-1				
3	1/11	6	18	-9	2			
4	1/25	12	48	-36	16	-3		
5	1/137	60	300	-30	200	-75	12	
6	1/147	60	360	-450	400	-225	72	-10

California State University Northridge 25

Newton's Method

- Iterate to find y such that g(y) = 0, write Taylor series for step from old y value, $y^{(m)}$ to new y value, $y^{(m+1)}$

$$g(y^{(m+1)}) = g(y^{(m)}) + \left.\frac{dg}{dy}\right|^{(m)} \left(y^{(m+1)} - y^{(m)}\right) + O(h^2)$$

- Set $g(y^{(m+1)}) = 0$ and solve for $y^{(m+1)} - y^{(m)}$
  - Use this to iterate since solution is not exact

$$y^{(m+1)} - y^{(m)} = \frac{-g(y^{(m)})}{\left.\frac{dg}{dy}\right|^{(m)}}$$

California State University Northridge 26

Solving $y_{n+1} = \gamma\beta h f_{n+1} + \sum_{j=0}^{k} \alpha_{-j} y_{n-j}$

$$g(y_{n+1}) = y_{n+1} - \gamma\beta h f(x_{n+1}, y_{n+1}) - \sum_{j=0}^{k} \alpha_{-j} y_{n-j} = 0$$

$$y_{n+1}^{(m+1)} - y_{n+1}^{(m)} = \frac{-g(y_{n+1}^{(m)})}{\left.\frac{dg}{dy}\right|_{n+1}^{(m)}}$$

- Iterate to get y_{n+1} (m is iteration index)

$$\left.\frac{dg}{dy}\right|_{n+1}^{(m)} = 1 - \gamma\beta h \left.\frac{\partial f}{\partial y}\right|_{n+1}^{(m)}$$

California State University Northridge 27

Multivariable Gear Method

- Implicit equation in $\mathbf{y}_{n+1}$

$$y_{i,n+1} = \gamma\beta h f\left(x_{n+1}, \mathbf{y}_{n+1}\right) + \sum_{j=0}^{k} \alpha_{-j} y_{i,n-j}$$

- Expand f_{n+1} in Taylor series between iteration m and iteration m+1 for y_{n+1}

$$y_{i,n+1}^{(m+1)} = \gamma\beta h \left[f_i\left(x_{n+1}, \mathbf{y}_{n+1}^{(m)}\right) + \sum_{j=1}^{N_{ODE}} \left.\frac{\partial f_i}{\partial y_j}\right|_{n+1}^{(m)} \bullet \left(y_{j,n+1}^{(m+1)} - y_{j,n+1}^{(m)}\right) + O(h^2) \right] + \sum_{j=0}^{k} \alpha_{-j} y_{i,n-j}$$

California State University Northridge 28

Multivariable Gear Method II

- We have a set of simultaneous linear equations to solve for $\mathbf{y}_{n+1}$ components at each iteration

$$\sum_{j=1}^{N_{ODE}} \left[\delta_{ij} - \gamma\beta h \left.\frac{\partial f_i}{\partial y_j}\right|_{n+1}^{(m)} \right] \left(y_{j,n+1}^{(m+1)} - y_{j,n+1}^{(m)}\right)$$

$$= -y_{j,n+1}^{(m)} + \gamma\beta h f_i\left(x_{n+1}, \mathbf{y}_{n+1}^{(m)}\right) + \sum_{j=0}^{k} \alpha_{-j} y_{i,n-j}$$

California State University Northridge 29

Partial Derivatives

- Need partial derivatives of each f_i with respect to each y_j

$$\frac{dy_1}{dx} = -y_1 + \sqrt{y_2} + y_3 e^{2x}$$

$$\frac{dy_2}{dx} = -2y_1^2 \qquad \frac{dy_3}{dx} = -3y_1 y_2$$

$$\frac{\partial f_1}{\partial y_1} = -1 \qquad \frac{\partial f_1}{\partial y_2} = \frac{1}{2} y_2^{-1/2} \qquad \frac{\partial f_1}{\partial y_3} = e^{2x}$$

$$\frac{\partial f_2}{\partial y_1} = -4y_1 \qquad \frac{\partial f_2}{\partial y_2} = 0 \qquad \frac{\partial f_2}{\partial y_3} = 0$$

$$\frac{\partial f_3}{\partial y_1} = -3y_2 \qquad \frac{\partial f_3}{\partial y_2} = -3y_1 \qquad \frac{\partial f_2}{\partial y_3} = 0$$

California State University Northridge 30

Subroutine for Partials

- This can come in different forms depending on array allocation of code
- One simple form is to use repeated calls to the subroutine for f_i
- The index, i, and the values of x and the **y** array are passed to the routine and the values of the partial derivatives of f_k with respect to each y_j are returned in a one dimensional array, p

California State University Northridge 31

C++ Function for Partials

```
void pderiv( double x, double y[],
             int k, double p[] )
{
    if ( k == 1 )
    {   p[1] = -1;
        p[2] = 0.5 / sqrt( y[2] );
        p[3] = exp( 2 * x );
    }
    else if ( k == 2 )
```

California State University Northridge 32

C++ Function for Partials II

```
    {   p[1] = -4 * y[1];
        p[2] = p[3] = 0;
    }
    else if ( k == 3 )
    {   p[1] = -3 * y[2];
        p[2] = -3 * y[1];
        p[3] = 0;
    }
}
```

California State University Northridge 33

Group Work

- Start work on the first problem on the homework for November 29
- Solve the problem y' = -0.2xy with y(0) = 1 for four steps, with h = 0.2, using the Adams-Moulton method
- See the next slide for the algorithm as well as the starting values from the fourth-order Runge-Kutta
- Exact solution is y = $e^{-0.1x^2}$

California State University Northridge 34

Group Work II

- Use Adams method to solve y' = -0.2xy with y(0) = 1 for four steps, with h = 0.2

$$y_{i+1}^{P} = y_i + \frac{h}{24}\left[-9f_{i-3} + 37f_{i-2} - 59f_{i-1} + 55f_i\right]$$

$$y_{i+1}^{C} = y_i + \frac{h}{24}\left[f_{i-2} - 5f_{i-1} + 19f_i + 9f(x_{i+1}, y_{i+1}^{P})\right]$$

i	x_i	y_i	exact y_i	error	f_i
0	0	1	1	0	0
1	0.2	0.996008	0.996008	1.07E-11	-0.03984
2	0.4	0.984127	0.984127	1.04E-10	-0.07873
3	0.6	0.964640	0.964640	3.38E-10	-0.11576

California State University Northridge 35