Numerical Solutions of Ordinary Differential Equations

Larry Caretto

Mechanical Engineering 501AB

Seminar in Engineering Analysis

November 8, 2017

Outline

- Review last class and homework
- Midterm Exam November 15 covers material on differential equations and Laplace transforms (no phase plots)
- Overview of numerical solutions
  - Initial value problems in first-order equations
  - Systems of first order equations and initial value problems in higher order equations
  - Boundary value problems
  - Stiff systems and eigenvalues

Newton Polynomials

- $p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \dots + a_{n-1}(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-2})$
- Terms with factors of $x - x_i$ are zero when $x = x_i$
  - Use this and rule that $p(x_i) = y_i$ to find a_i
  - $a_0 = y_0$, $a_1 = (y_1 - y_0) / (x_1 - x_0)$, etc.
- Coefficients can be obtained from divided difference table

Divided Difference Example

0	0	$\leftarrow a_0$		
		$F_0 = \frac{10-0}{10-0} = 1$	$\leftarrow a_1$	
10	10		$S_0 = \frac{3-1}{20-0} = .1$	$\leftarrow a_2$
		$F_1 = \frac{40-10}{20-10} = 3$		$T_0 = \frac{.2-.15}{30-0} = \frac{1}{600}$
20	40		$S_1 = \frac{6-3}{30-10} = .15$	$\uparrow a_3$
		$F_2 = \frac{100-40}{30-20} = 6$		
30	100			

Constant Step Size

- Divided differences work for equal or unequal step size in x
- For constant step size we can use an alternative formulation involving forward or backward differences
  - $\Delta y_k = (y_{k+1} - y_k)/h$
  - $\Delta^2 y_k = (y_{k+2} - 2y_{k-1} + y_k)$
  - $\Delta^3 y_k = (y_{k+3} - 3y_{k+2} + 3y_{k+1} - y_k)$

Polynomial Interpolation

- Data interpolation only if points are exact, use statistical fits otherwise
- Use piecewise curve fits to large number of data points (e.g. cubic splines)
- Interpolation used for other numerical methods, quadrature, differential equations, finite element basis functions, derivative expressions

Truncation Error

- If we truncate series after m terms

$$f(x) = \underbrace{\sum_{n=0}^{m} \frac{1}{n!} \frac{d^n f}{dx^n}\bigg|_{x=a} (x-a)^n}_{\text{Terms used}} + \underbrace{\sum_{n=m+1}^{\infty} \frac{1}{n!} \frac{d^n f}{dx^n}\bigg|_{x=a} (x-a)^n}_{\text{Truncation error, } \varepsilon_m}$$

Terms used   Truncation error, ε_m

- Truncation error as single term at unknown location

$$\varepsilon_m = \sum_{n=m+1}^{\infty} \frac{1}{n!} \frac{d^n f}{dx^n}\bigg|_{x=a} (x-a)^n = \frac{1}{(m+1)!} \frac{d^{m+1} f}{dx^{m+1}}\bigg|_{x=\xi} (x-a)^{m+1}$$

- Derive finite-differences for derivatives

California State University Northridge 7

Derivatives and Error Order

- Error proportional to h^n called n^{th} order error
- Reducing step size by a factor of a reduces n^{th} order error by a^n

$$\varepsilon_2 \approx \varepsilon_1 \left(\frac{h_2}{h_1}\right)^n$$

- Second derivative example

$$f_i'' = \frac{f_{i+1} + f_{i-1} - 2f_i}{h^2} + \frac{f_i''''h^2}{4!} + \frac{f_i''''''h^4}{6!} - .....$$

$$= \frac{f_{i+1} + f_{i-1} - 2f_i}{h^2} + \frac{f_\xi''''h^2}{4!} = \frac{f_{i+1} + f_{i-1} - 2f_i}{h^2} + O(h^2)$$

California State University Northridge 8

First Derivative Expressions

First order forward $\quad f_i' = \frac{f_{i+1} - f_i}{h} + O(h)$

First order backward $\quad f_i' = \frac{f_i - f_{i-1}}{h} + O(h)$

Second order central $\quad f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2)$

Second order forward $\quad f_i' = \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} + f_\xi''' \frac{h^2}{3}$

Second order backwards $\quad f_i' = \frac{f_{i-2} - 4f_{i-1} + 3f_i}{2h} + f_\xi''' \frac{h^2}{3}$

California State University Northridge 9

Roundoff Error

- Possible in derivative expressions from subtracting close differences
- Example f(x) = e^x: $f'(x) \approx (e^{x+h} - e^{x-h})/(2h)$ and error at x = 1 is $(e^{1+h} - e^{1-h})/(2h)$ - e

$$E = \frac{3.004166 - 2.722815}{2(0.1)} - 2.718282 = 4.5x10^{-3}$$

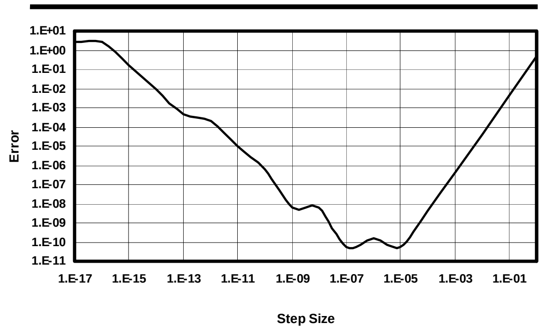
Second order error

$$E = \frac{2.7185536702 - 2.7180100139}{2(0.0001)} - 2.718281828459 = 4.5x10^{-9}$$

$$E = \frac{2.71828210028724 - 2.71828155660388}{2(0.0000001)} - 2.718281828 = 5.9x10^{-9}$$

California State University Northridge 10

Figure 2-1. Effect of Step Size on Error



California State University Northridge 11

Numerical ODE Solutions

- The initial value problem
- Euler method as a prototype for the general algorithm
- Local and global errors
- More accurate methods
- Step-size control for error control
- Applications to systems of equations
  - Reduce higher-order equations to a system of equations

California State University Northridge 12

The Initial Value Problem

- $dy/dx = f(x,y)$ (known f) with $y(x_0) = y_0$
- Basic numerical approach
  - Use a finite difference grid: $x_{i+1} - x_i = h$
  - Replace derivative by finite-difference approximation: $dy/dx \approx (y_{i+1} - y_i) / (x_{i+1} - x_i) = (y_{i+1} - y_i) / h$
  - Derive a formula to compute f_{avg} the average value of $f(x,y)$ between x_i and x_{i+1}
  - Replace $dy/dx = f(x,y)$ by $(y_{i+1} - y_i) / h = f_{avg}$
  - Repeatedly compute $y_{i+1} = y_i + h f_{avg}$

California State University Northridge 13

Notation

- x_i is the value of the independent variable at point i on the grid
  - Determined from the user-selected value of step size (or a series of h_i values)
  - Can always specify exactly the independent variable's value, x_i
- y_i is the value of the numerical solution at the point where $x = x_i$
- f_i is derivative value found from x_i and the numerical value, y_i. *I.e.*, $f_i = f(x_i, y_i)$

California State University Northridge 14

More Notation

- $y(x_i)$ is the exact value of y at $x = x_i$
  - Usually not known but notation is used in error analysis of algorithms
- $f(x_i,y(x_i))$ is the exact value of the derivative at $x = x_i$
- $e_1 = y(x_1) - y_1$ is the local truncation error
  - This is error for one step of algorithm starting from known initial condition

California State University Northridge 15

Local *versus* Global Error

- At the initial point, x, we know the solution, y, from given initial condition
- First step introduces some error
- Remaining steps have single step error plus previous accumulated error
- $E_j = y(x_j) - y_j$ is global truncation error
  - Difference between numerical and exact solution after several steps
  - This is the error we want to control

California State University Northridge 16

Euler's Method

- Simplest algorithm, example used for error analysis, not for practical use
- Define $f_{avg} = f(x_i, y_i) = f_i$
- Euler's method algorithm is $y_{i+1} = y_i + hf_i = y_i + hf(x_i, y_i)$
- Example $dy/dx = x + y$, $y = 0$ at $x = 0$
- Choose $h = 0.1$
- We have $x_0 = 0$, $y_0 = 0$, $f_0 = x_0 + y_0 = 0$, $x_1 = x_0 + h = 0.1$, $y_1 = y_0 + hf_0 = 0 + 0 = 0$

California State University Northridge 17

Euler Example Continued

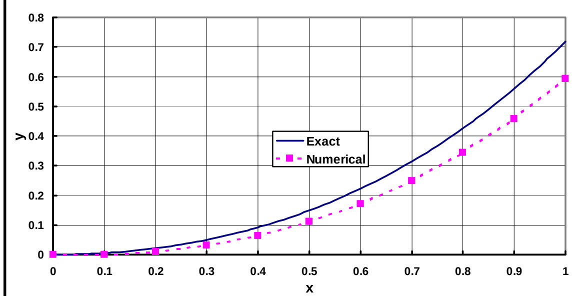
- Next step is from $x_1 = 0.1$ to $x_2 = 0.2$
- $f_1 = x_1 + y_1 = 0.1 + 0 = 0.1$
- $y_2 = y_1 + hf_1 = 0 + (.1)(.1) = .01$
- Can continue in this fashion
- For $dy/dx = x + y$, we know the exact solution is $y = (x_0 + y_0 + 1)e^{x-x_0} - x - 1$
- For $x_0 = y_0 = 0$, $y = e^x - x - 1$
- Look at application of Euler algorithm for a few steps and compute the error

California State University Northridge 18

Euler Example

x_i	y_i	f_i	$f(x_i,y(x_i))$	$y(x_i)$	$E(x_i)$
0	0	0	0	0	0
.1	0	.1	.1052	.0052	.0052
.2	.01	.21	.2214	.0214	.0114
.3	.031	.331	.3499	.04986	.01886
.4	.0641	.4641	.4918	.091825	.027725

Euler Example Plot



Error Propagation

- Behavior of Euler algorithm is typical of all algorithms for numerical solutions
- Error grows at each step
- We usually do not know this global error, but we would like to control it
- Look at local error for Euler algorithm
- Then discuss general relationship between local and global error

Taylor Series to Get Error

- Expand y(x) in Taylor series about x = a

$$y(x) = y(a) + \left.\frac{dy}{dx}\right|_{x=a}(x-a) + \frac{1}{2!}\left.\frac{d^2y}{dx^2}\right|_{x=a}(x-a)^2 + \frac{1}{3!}\left.\frac{d^3y}{dx^3}\right|_{x=a}(x-a)^3 + \ldots$$

- Look at one step from known initial condition, a = x_0, to $x_0 + h$ so $x - a = h$

$$y(x_0+h) = y(x_0) + \left.\frac{dy}{dx}\right|_0 h + \frac{1}{2!}\left.\frac{d^2y}{dx^2}\right|_0 h^2 + \frac{1}{3!}\left.\frac{d^3y}{dx^3}\right|_0 h^3 + \ldots$$

- In ODE notation, $dy/dx|_0 = f(x_0, y(x_0))$

Local Euler Error

- Result of Taylor series on last chart

$$\underbrace{y(x_0+h) = y(x_0) + hf(x_0, y(x_0))}_{\text{Euler Algorithm}} + \underbrace{\frac{1}{2!}\left.\frac{d^2y}{dx^2}\right|_0 h^2 + \frac{1}{3!}\left.\frac{d^3y}{dx^3}\right|_0 h^3 + \ldots}_{\text{Truncation Error}}$$

Euler Algorithm — Truncation Error

- This is only the Euler algorithm for the first step when we know $f(x_0,y(x_0))$
- This gives the local truncation error
- Local truncation error for Euler algorithm is second order

Global Error

- We will show that a local error of order n, has a global error of order n-1
- To show this consider the global error at $x = x_0 + kh$ after k algorithm steps
  - Is approximately k times the local error
  - If local error is $O(h^n) \approx Ah^n$, approximate global error after k steps is $k\,O(h^n) \approx kAh^n$
  - A new step size, h/r, takes kr steps to get to the same x value

Global Error Concluded

- Compare error for same x = kh with step sizes h and h/r
- $E_{x=kh}(h) \approx kAh^n$
- $E_{x=kh}(h/r) \approx krA(h/r)^n$

$$\frac{E_{x=kh}\left(h/r\right)}{E_{x=kh}(h)} \approx \frac{kr\left(h/r\right)^n}{k(h)^n} = \frac{1}{r^{n-1}}$$

- When we reduce the step size by a factor of 1/r we reduce the error by a factor of $1/r^{n-1}$; this is the behavior of an algorithm whose error is order n-1

California State University Northridge 25

Euler Local and Global Error

- Previously showed Euler algorithm to have second order local error
- Should have first order global error
- Results for previous Euler example at x = 1 with different step sizes

Step size	First step	Final error
h = 0.1	5.17×10^{-3}	1.25×10^{-1}
h = 0.01	5.02×10^{-5}	1.35×10^{-2}
h = 0.001	5.00×10^{-7}	1.36×10^{-3}

California State University Northridge 26

Better Algorithms

- Seek high accuracy with low computational work
- Could improve Euler accuracy by cutting step size, but this is not efficient
- Use other algorithms that have higher order errors
- Runge-Kutta methods commonly used
  - This is a class of methods that use several derivative evaluations per step

California State University Northridge 27

Second-order Runge Kutta

- Huen's method

$$y_{i+1}^0 = y_i + h_{i+1} f(x_i, y_i) \qquad x_{i+1} = x_i + h_{i+1}$$

$$y_{i+1} = y_i + \frac{h_{i+1}}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)\right] = \frac{y_i + y_{i+1}^0 + h_{i+1} f(x_{i+1}, y_{i+1}^0)}{2}$$

- Modified Euler method

$$y_{i+1/2} = y_i + \left[\frac{h_{i+1}}{2}\right] f(x_i, y_i) \qquad x_{i+1/2} = x_i + \frac{h_{i+1}}{2}$$

$$y_{i+1} = y_i + h_{i+1} f(x_{i+1/2}, y_{i+1/2})$$

California State University Northridge 28

Fourth-order Runge Kutta

- Uses four derivative evaluations per step

$$y_{i+1} = y_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \qquad x_{i+1} = x_i + h_{i+1}$$

$$k_1 = h_{i+1} f(x_i, y_i)$$

$$k_2 = h_{i+1} f\left(x_i + \frac{h_{i+1}}{2}, y_i + \frac{k_1}{2}\right)$$

$$k_3 = h_{i+1} f\left(x_i + \frac{h_{i+1}}{2}, y_i + \frac{k_2}{2}\right)$$

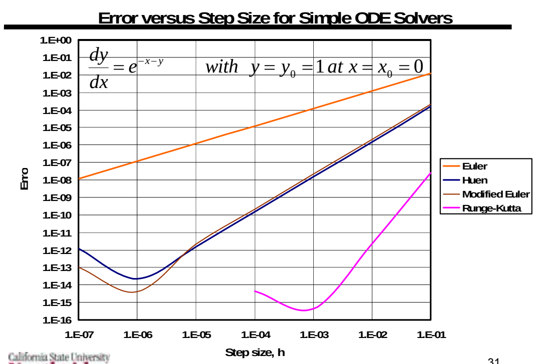
$$k_4 = h_{i+1} f(x_i + h_{i+1}, y_i + k_3)$$

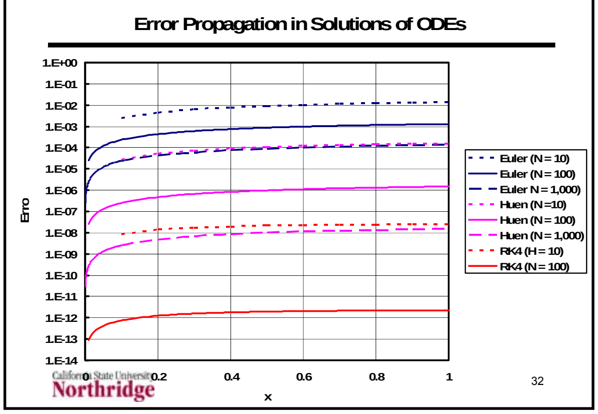
California State University Northridge 29

Comparison of Methods

- Look at Euler, Heun, Modified Euler and fourth-order Runge-Kutta
- Solve $dy/dx = e^{-y-x}$ with y(0) = 1
- Compare numerical values to exact solution $y = \ln(e^{y_0} + e^{-x_0} - e^{-x})$
- Look at errors in the methods at x = 1 as a function of step size
- Compare error propagation (increase in error as x increases)

California State University Northridge 30





Generic Runge-Kutta

- Generic formulas shown below
  - Step size control based on $y_i - y_i^*$

$$y_{i+1} = y_i + h\sum_{i=1}^{s} b_i k_i \qquad y_{i+1}^* = y_i + h\sum_{i=1}^{s} b_i^* k_i$$

$$k_1 = hf(x_i, y_i)$$
$$k_2 = hf(x_i + c_2 h, y_i + a_{21}hk_1)$$
$$k_3 = hf(x_i + c_3 h, y_i + a_{31}hk_1 + a_{32}hk_2)$$
$$\cdots$$
$$k_s = hf(x_i + c_s h, y_i + a_{s1}hk_1 + a_{s2}hk_2 + \cdots a_{s,s-1}hk_{s-1})$$

Butcher Tableau

- Coefficients for generic Runge-Kutta

0						
c_2	$a_{2,1}$					
c_3	$a_{3,1}$	$a_{3,2}$				
c_4	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$			
⋮	⋮	⋮	⋮	⋮	⋮	⋮
c_s	$a_{s,1}$	$a_{s,2}$	$a_{s,3}$	…	$a_{s,s-1}$	
	b_1	b_2	b_3	…	b_{s-1}	b_s
	b_1^*	b_2^*	b_3^*	…	b_{s-1}^*	b_s^*

Class Exercise

- Apply RK4 to y' = x + y with y = 0 at x = 0

$$y_{i+1} = y_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \qquad x_{i+1} = x_i + h_{i+1}$$

$$k_1 = h_{i+1} f(x_i, y_i)$$
$$k_2 = h_{i+1} f\left(x_i + \frac{h_{i+1}}{2}, y_i + \frac{k_1}{2}\right)$$
$$k_3 = h_{i+1} f\left(x_i + \frac{h_{i+1}}{2}, y_i + \frac{k_2}{2}\right)$$
$$k_4 = h_{i+1} f(x_i + h_{i+1}, y_i + k_3)$$

Take 3 steps with h = 0.1. Compare result to exact solution $y = e^x - x - 1$

Solution to Class Exercise I

$$k_1 = hf(x_i, y_i) = x_i + y_i = 0.1(0 + 0) = 0$$
$$k_2 = hf\left(x_i + \frac{h_{i+1}}{2}, y_i + \frac{k_1}{2}\right) = 0.1\left(0 + \frac{0.1}{2} + 0 + \frac{0}{2}\right) = 0.005$$
$$k_3 = hf\left(x_i + \frac{h_{i+1}}{2}, y_i + \frac{k_2}{2}\right) = 0.1\left(0 + \frac{0.1}{2} + 0 + \frac{0.005}{2}\right)$$
$$= 0.00525$$
$$k_4 = hf(x_i + h_{i+1}, y_i + k_3) = 0.1(0 + 0.1 + 0 + 0.00525)$$
$$= 0.010525$$
$$y_{i+1} = y_i + (k_1 + 2k_3 + 2k_3 + k_4)/6 =$$
$$0 + [0 + 2(0.005) + 2(0.00525) + 0.01052]/6 = 0.005171$$

Solution to Class Exercise II

- From x = 0.1 (y = 0.0005171) to x = 0.2

$x_1 = x_0 + h = 0 + 0.1 = 0.1$

$k_1 = hf(x_1, y_1) = h(x_1 + y_1) = 0.1(0.1 + 0.005171) = 0.010517$

$k_2 = hf\left(x_1 + h/2, y_1 + k_1/2\right) = 0.1\left(0.1 + 0.1/2 + 0.005171 + 0.010517/2\right) = 0.016043$

$k_3 = hf\left(x_1 + h/2, y_1 + k_2/2\right) = 0.1\left(0.1 + 0.1/2 + 0.005171 + 0.016043/2\right)$

$= 0.016319$

$k_4 = hf(x_1 + h, y_1 + k_3) = 0.1(0.1 + 0.1 + 0.005171 + 0.022149)$

$= 0.022149$

$y_2 = y_1 + (k_1 + 2k_3 + 2k_3 + k_4)/6 = 0.005171 +$

$[0.010517 + 2(0.016043) + 2(0.016319) + 0.022149]/6 = 0.0214027$

California State University Northridge

37

Solution to Class Exercise

x	y	k1	k2	k3	k4	New y
0	0	0	0.005	0.00525	0.010525	0.005170833
0.1	0.005171	0.010517	0.016043	0.016319	0.022149	0.021402665
0.2	0.021403	0.022140	0.028247	0.028553	0.034996	0.049858704
0.3	0.049859	0.034986	0.041735	0.042073	0.049193	0.091824583
0.4	0.091825	0.049182	0.056642	0.057015	0.064884	0.148721144
0.5	0.148721	0.064872	0.073116	0.073528	0.082225	0.222118661
0.6	0.222119	0.082212	0.091322	0.091778	0.101390	0.313752553
0.7	0.313753	0.101375	0.111444	0.111947	0.122570	0.425540758
0.8	0.425541	0.122554	0.133682	0.134238	0.145978	0.559602923
0.9	0.559603	0.145960	0.158258	0.158873	0.171848	0.718281620
1.0	0.718282	0.171828	0.185420	0.186099	0.200438	0.904165794

California State University Northridge

38

Solution to Class Exercise

x	y	k1	k2	k3	k4	New y	y Exact	Error
0.0	0	0	0.005	0.00525	0.010525	0.005171	0.005171	8.47E-08
0.1	0.005171	0.010517	0.016043	0.016319	0.022149	0.021403	0.021403	9.37E-08
0.2	0.021403	0.022140	0.028247	0.028553	0.034996	0.049859	0.049859	1.04E-07
0.3	0.049859	0.034986	0.041735	0.042073	0.049193	0.091825	0.091825	1.14E-07
0.4	0.091825	0.049182	0.056642	0.057015	0.064884	0.148721	0.148721	1.26E-07
0.5	0.148721	0.064872	0.073116	0.073528	0.082225	0.222119	0.222119	1.40E-07
0.6	0.222119	0.082212	0.091322	0.091778	0.101390	0.313753	0.313753	1.54E-07
0.7	0.313753	0.101375	0.111444	0.111947	0.122570	0.425541	0.425541	1.71E-07
0.8	0.425541	0.122554	0.133682	0.134238	0.145978	0.559603	0.559603	1.89E-07
0.9	0.559603	0.145960	0.158258	0.158873	0.171848	0.718282	0.718282	2.08E-07
1.0	0.718282	0.171828	0.185420	0.186099	0.200438	0.904166	0.904166	2.30E-07
1.1	0.904166	0.200417	0.215437	0.216188	0.232035	1.120117	1.120117	2.55E-07
1.2	1.120117	0.232012	0.248612	0.249442	0.266956	1.369296	1.369297	2.81E-07
1.3	1.369297	0.266930	0.285276	0.286193	0.305549	1.655200	1.655200	3.11E-07
1.4	1.655200	0.305520	0.325796	0.326810	0.348201	1.981689	1.981689	3.44E-07
1.5	1.981689	0.348169	0.370577	0.371698	0.395339	2.353032	2.353032	3.80E-07
1.6	2.353032	0.395303	0.420068	0.421307	0.447434	2.773947	2.773947	4.20E-07
1.7	2.773947	0.447395	0.474764	0.476133	0.505008	3.249647	3.249647	4.64E-07
1.8	3.249647	0.504965	0.535213	0.536725	0.568637	3.785894	3.785894	5.13E-07
1.9	3.785894	0.568589	0.602019	0.603690	0.638958	4.389056	4.389056	5.67E-07
2.0	4.389056	0.638906	0.675851	0.677698	0.716675	5.066169	5.066170	6.26E-07