Implicit, Multistep and Extrapolation Methods

Larry Caretto
Mechanical Engineering 501AB

Seminar in Engineering Analysis

November 13, 2017

California State University Northridge

Outline

- Review last class
  - One step methods for numerical solution of differential equations
  - Local and global error
- Multistep methods with constant and variable step size
- Implicit methods using future time steps
- Extrapolation methods
- Review for Midterm

California State University Northridge 2

Review Numerical Approach

- Solve initial value problem, $dy/dx = f(x, y)$ (f a known function) with $y(x_0) = y_0$
  - 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 3

Review Notation and Order

- x_i is independent variable
- y_i is numerical solution at $x = x_i$
- f_i is derivative found from x_i y_i: $f_i = f(x_i, y_i)$
- $y(x_i)$ is the exact value of y at $x = x_i$
- $f(x_i, y(x_i))$ is the exact derivative
- $e_1 = y(x_1) - y_1$ = local truncation error
- $E_j = y(x_j) - y_j$ is global truncation error
- If local error, e, is $O(h^n)$, then global error, E, is $O(h^{n-1})$

California State University Northridge 4

Review Simple Methods

- Euler: $y_{i+1} = y_i + h_i f_i = y_i + h_i f(x_i, y_i)$
- 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

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

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

California State University Northridge 5

Review 4th Order Runge-Kutta

- Uses four derivative evaluations per step

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

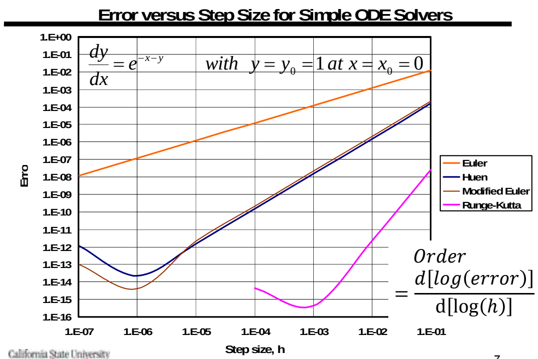
$$k_1 = h_{i+1} f(x_i, y_i)$$

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

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

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

California State University Northridge 6



California State University Northridge

7

Implicit Methods Overview

- Methods discussed previously are called explicit
  - Can find y_{n+1} in terms of values at n
  - Use f values from estimated values of y between y_n and y_{n+1} to get final y_{n+1}
- Implicit methods use f_{n+1} in algorithm
- Usually require approximate solution
- Have better stability but require more work than explicit methods
- Trapezoid method is an example

California State University Northridge

8

Derive Trapezoid Method I

- Get series for y_{n+1} and y_n about $y_{n+1/2}$

$$y_{n+1} = y_{n+1/2} + y'_{n+1/2}\frac{h}{2} + \frac{y''_{n+1/2}}{2}\left(\frac{h}{2}\right)^2 + O(h^3)$$

$$y_n = y_{n+1/2} - y'_{n+1/2}\frac{h}{2} + \frac{y''_{n+1/2}}{2}\left(\frac{h}{2}\right)^2 + O(h^3)$$

- Subtract series to get $y_{n+1} - y_n$

$$y_{n+1} - y_n = y'_{n+1/2}h + O(h^3)$$

- Need expression for $y'_{n+1/2}$

California State University Northridge

9

Derive Trapezoid Method II

- Write series for y'_{n+1} and y'_n about $y'_{n+1/2}$, add them, and solve result for $y'_{n+1/2}$

$$y'_{n+1} = y'_{n+1/2} + y''_{n+1/2}\frac{h}{2} + O(h^2) \qquad y'_n = y'_{n+1/2} - y''_{n+1/2}\frac{h}{2} + O(h^2)$$

$$y'_{n+1} + y'_n = 2y'_{n+1/2} + O(h^2) \qquad y'_{n+1/2} = \frac{y'_{n+1} + y'_n}{2} + O(h^2)$$

- Substitute expression for $y'_{n+1/2}$ into previous expression for $y_{n+1} - y_n$

$$y_{n+1} - y_n = y'_{n+1/2}h + O(h^3) = \left[\frac{y'_{n+1} + y'_n}{2} + O(h^2)\right]h = \frac{y'_{n+1} + y'_n}{2}h + O(h^3)$$

$$y_{n+1} - y_n = \frac{y'_{n+1} + y'_n}{2}h + O(h^3) = \frac{f_{n+1} + f_n}{2}h + O(h^3)$$

California State University Northridge

10

Implicit Methods with Iteration

- How can we use f_{n+1} in algorithm to solve for the unknown y_{n+1}?
  - One approach is trial-and-error solution
  - Euler step for first approximation of y_{n+1}
  - Iterate on implicit method

$$y_{n+1}^{(0)} = y_n + hf_n$$

$$y_{n+1}^{(m+1)} = y_n + \frac{h\left[f_n + f\left(x_{n+1}, y_{n+1}^{(m)}\right)\right]}{2}$$

California State University Northridge

11

Iteration Approach Example

- Use Newton-Raphson iteration for y_{n+1}
  - Solve g(y) = 0 by iteration: $y^{(m+1)} = y^{(m)} - g(y^{(m)}) / g'(y^{(m)})$
  - $g(y^{(m)}) = y_{n+1} - y_n - hf_n/2 - hf(x_{n+1},y_{n+1})/2$
  - $g'(y^{(m)}) = f_{n+1} - 0 - 0 - h(\partial f/\partial y)/2$

$$y_{n+1}^{(m+1)} = y_{n+1}^{(m)} - \frac{y_{n+1}^{(m)} - y_n - \frac{hf_n}{2} - \frac{hf\left(x_{n+1}, y_{n+1}^{(m)}\right)}{2}}{f\left(x_{n+1}, y_{n+1}^{(m)}\right) - \frac{h}{2}\left(\frac{\partial f}{\partial y}\right)_{n+1}^{(m)}}$$

California State University Northridge

12

f(x,y) Taylor Series Approach

- Have to f(x,y) expand for both x and y

$$f_{n+1} = f_n + \left.\frac{\partial f}{\partial x}\right|_n h + \left.\frac{\partial f}{\partial y}\right|_n (y_{n+1} - y_n) + O(h^2)$$

- Substitute for f_{n+1} in trapezoid equation

$$y_{n+1} - y_n = \frac{h(f_n + f_{n+1})}{2} + O(h^3) =$$

$$\frac{h}{2}\left[f_n + f_n + \left.\frac{\partial f}{\partial x}\right|_n h + \left.\frac{\partial f}{\partial y}\right|_n (y_{n+1} + y_n) + O(h^2) \right] + O(h^3)$$

$$y_{n+1} - y_n = \frac{h}{2}\left[2 f_n + \left.\frac{\partial f}{\partial x}\right|_n h + \left.\frac{\partial f}{\partial y}\right|_n (y_{n+1} + y_n) \right] + O(h^3)$$

California State University Northridge 13

Trapezoid Method Result

- Solve equation below from last slide for $y_{n+1} - y_n$

$$y_{n+1} - y_n = \frac{h}{2}\left[2 f_n + \left.\frac{\partial f}{\partial x}\right|_n h + \left.\frac{\partial f}{\partial y}\right|_n (y_{n+1} + y_n) \right] + O(h^3)$$

$$(y_{n+1} - y_n)\left(1 - \frac{h}{2}\left.\frac{\partial f}{\partial y}\right|_n\right) = \frac{h}{2}\left[2 f_n + \left.\frac{\partial f}{\partial x}\right|_n h \right] + O(h^3)$$

$$y_{n+1} = y_n + \frac{h f_n + \frac{h^2}{2}\left.\frac{\partial f}{\partial x}\right|_n}{1 - \frac{h}{2}\left.\frac{\partial f}{\partial y}\right|_n} + O(h^3)$$

California State University Northridge 14

Multistep Methods

- Previous methods used only information from most recent step (y_n and f_n)
- Took intermediate steps between x_n and x_{n+1} to improve accuracy
- Multistep methods use information from previous steps for improved accuracy with less work than single step methods
- Need starting procedure that is a single step method

California State University Northridge 15

Multistep Method Derivation

- Uses "interpolation" polynomial that passes through previous points
- Polynomial is integrated from x_n to x_{n+1}
- Resulting expression gives y_{n+1} in terms of values and derivatives of previous steps
- Leads to process known as predictor-corrector with two expressions for y_{n+1} and an error control expression

California State University Northridge 16

Adams-Bashforth-Moulton

- Predictor corrector method
- Predictor equation uses four points

$$y_{n+1}^P = y_n + \frac{h}{24}(55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3})$$

- Corrector equation uses four points including point n+1 with predicted y^P

$$y_{n+1}^C = y_n + \frac{h}{24}\left(9 f\left(x_{n+1}, y_{n+1}^P\right) + 19 f_n - 5 f_{n-1} + f_{n-2}\right)$$

California State University Northridge 17

Adams-Bashforth-Moulton II

- Use difference between predictor and corrector results to get error estimate

$$y_{n+1}^P = y_n + \frac{h}{24}(55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3})$$

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

- Error estimate, E_C, derivation on next three slides – result below

$$E_C = -\frac{19}{720} h^5 y^{(v)}(\xi_C) \approx \frac{19}{270}\left(y_{n+1}^P - y_{n+1}^C\right)$$

California State University Northridge 18

Derive Error Equation (1/3)

- From an error analysis of the integrated interpolation polynomials we can find

$$y(x_{n+1}) = y^P_{n+1} + \frac{251}{720}h^5 y^{(v)}(\xi_P)$$

$$y(x_{n+1}) = y^C_{n+1} - \frac{19}{720}h^5 y^{(v)}(\xi_C)$$

1. Subtract equations
2. Subtract and add $251y^{(v)}(\xi_C)/720$

$$0 = y^P_{n+1} + \frac{251}{720}h^5 y^{(v)}(\xi_P) - y^C_{n+1} + \frac{19}{720}h^5 y^{(v)}(\xi_C)$$

$$0 = y^P_{n+1} - y^C_{n+1} + \frac{251}{720}h^5\left[y^{(v)}(\xi_P) - y^{(v)}(\xi_C)\right] + \left(\frac{19}{720} + \frac{251}{720}\right)h^5 y^{(v)}(\xi_C)$$

California State University Northridge 19

Derive Error Equation (2/3)

$$0 = y^P_{n+1} - y^C_{n+1} + \left(\frac{251}{720} + \frac{19}{720}\right)h^5 y^{(v)}(\xi_C) + \frac{251}{720}h^5\left[y^{(v)}(\xi_P) - y^{(v)}(\xi_C)\right]$$

- In equation above ignore $\left[y^{(v)}(\xi_P) - y^{(v)}(\xi_C)\right]$

$$y^C_{n+1} - y^P_{n+1} = \left(\frac{251}{720} + \frac{19}{720}\right)h^5 y^{(v)}(\xi_C) = \frac{270}{720}h^5 y^{(v)}(\xi_C)$$

- This allows calculation of truncation error term

$$h^5 y^{(v)}(\xi_C) = \frac{720}{270}\left(y^C_{n+1} - y^P_{n+1}\right)$$

California State University Northridge 20

Derive Error Equation (3/3)

- The corrector term truncation error was given by the following equation

$$y(x_{n+1}) = y^C_{n+1} - 19h^5 y^{(v)}(\xi_C)/720$$

- Use the equation for the truncation error that $h^5 y^{(v)}(\xi_C) = 720\left(y^C_{n+1} - y^P_{n+1}\right)/270$

$$y(x_{n+1}) = y^C_{n+1} - \frac{19}{720}\frac{720}{270}\left(y^C_{n+1} - y^P_{n+1}\right)$$

$$E_C = \left|y(x_{n+1}) - y^C_{n+1}\right| = \left|\frac{19}{270}\left(y^C_{n+1} - y^P_{n+1}\right)\right|$$

- Error estimate, E_C, for step size control
- How to change h in multistep method?

California State University Northridge 21

Step Size Control

- Establish e_{min} and e_{max} to achieve desired problem accuracy
- If $e_{min} \le E_C \le e_{max}$, do not change h
- If $E_C < e_{min}$ double step size, h
- If $E_C > e_{max}$ half step size, h
- Carry extra steps to be ready for step-size doubling
- Interpolate data if h is cut in half

California State University Northridge 22

Grid halving if error too large

- Normal operation, no step size change

i-3 i-2 i-1 i i+1 (old step)

●---------●---------●---------●---------●---------●

(new) i-3 i-2 i-1 i i+1

- Error too large: Half grid size and repeat step

i-3 i-2 i-1 i i+1 (old step)

●----o----●----o----●----o----●----o----●

(repeated) i-3 i-2 i-1 i i+1

California State University Northridge (interpolated points) 23

Grid doubling for very small error

- Normal operation, no step size change

i-5 i-4 i-3 i-2 i-1 i i+1 (old step)

o-----o-----o-----o-----o-----o-----o-----o

i-5 i-4 i-3 i-2 i-1 i i+1 (new)

- Error very small: Double grid size

i-5 i-4 i-3 i-2 i-1 i i+1 (old step)

●-----o-----●-----o-----●-----o-----●-----------●

i-3 i-2 i-1 i i+1

California State University Northridge (Retained to use for doubling) 24

Grid Halving and Doubling

- Keep extra values f_{i-4} and f_{i-5} in memory to be ready for grid doubling
  - $f_{i-3,new} = f_{i-5}$; $f_{i-2,new} = f_{i-3}$; $f_{i-1,new} = f_{i-1}$; $f_{i,new} = f_{i+1}$
- Grid halving requires interpolation for missing values in old grid
  - $f_{i-2,new} = f_{i-1}$; $f_{i,new} = f_i$

$$f_{i-1,new} = \frac{1}{128}\left[-5 f_{i-4} + 28 f_{i-3} - 70 f_{i-2} + 140 f_{i-1} + 35 f_i\right]$$

$$f_{i-3,new} = \frac{1}{64}\left[3 f_{i-4} - 16 f_{i-3} + 54 f_{i-2} + 24 f_{i-1} - f_i\right]$$

California State University Northridge 25

Use of Multistep Methods

- Many different equations possible with different orders and errors
- Used for high accuracy computation requirements with less computer time
- Used in high-accuracy MATLAB solver ode113 (variable step and order)
- Runge-Kutta type methods easier to apply, and can have error control for lower accuracy requirements

California State University Northridge 26

Extrapolation Methods Basis

- Use infinite series truncation error dependence on h to get better estimate from results on two values of h
  - Richardson extrapolation is example
  - t = true result, n(h) = numerical result with step size, h; error = $Ah^m + Bh^{m+a} + \ldots$ so $t = n(h) + Ah^m + Bh^{m+a} + \ldots$
  - For step size h/2, $t = n(h/2) + A(h/2)^m + \ldots$
  - Multiply equation for n(h/2) by 2^m and subtract equation for n(h) from result

California State University Northridge 27

Extrapolation Methods Basis II

- Continue process from previous chart

$$t = n(h) + Ah^m + Bh^{m+a} + \cdots$$

$$2^m\left[t = n\left(h/2\right) + A\left(h/2\right)^m + B\left(h/2\right)^{m+a} + \cdots\right]$$

$$\left(2^m - 1\right)t = 2^m n\left(h/2\right) - n(h) + B\left(\frac{2^m}{2^{m+a}} - 1\right)h^{m+a} + \cdots$$

$$t = \frac{2^n n\left(h/2\right) - n(h)}{2^n - 1} + B\left(\frac{1}{2^a} - 1\right)h^{m+a} + \cdots$$

California State University Northridge 28

Extrapolation Method Example

- Look at first order forward difference for the first derivative $f_i' = (f_{i+1} - f_i)/h + O(h^2)$
  - Apply to $f(x) = e^x$ at x = 1
  - Use h = 0.1 and h = 0.05
  - $f_i'(h=.1) = (e^{1.1} - e^1)/0.1 = 2.858842$
  - $f_i'(h=.05) = (e^{1.05} - e^1)/0.1 = 2.78736$
  - Extrapolation $[2^1(2.78736) - 2.858842] / (2^1 - 1) = 2.71593$
  - Error in three values is .14, .07 and .002

California State University Northridge 29

Extrapolation Benefits

- In general the truncation error infinite series, written as $Ah^m + Bh^{m+a} + \ldots$ has a = 1 so truncation error gets one higher order for extrapolation
- Have greater improvement if a = 2 as it does in some cases
  - Romberg integration is best example
- Can apply extrapolation to extrapolated results to reduce truncation error further

California State University Northridge 30

Extrapolation for ODE Solution

- Basis is solution method known as midpoint method
- Construct large step, H, between two x values, x and x + H
- Subdivide H into n smaller steps, h = H/n
- Compute intermediate approximations to y, called z_m for the substeps
- Use central difference approximations wherever possible

California State University Northridge 31

Extrapolation for ODE Solution II

- Start with results at x define $z_0 = y(x)$
- Compute $z_1 = z_0 + hf(x, z_0)$
- Central difference intermediate steps
  - $z_{m+1} = z_{m-1} + 2hf(x+mh,z_m)$ m = 1, 2, .. n-1
- Final value at x + H, called y_n, is an average of the central difference value, z_n, and a backward difference value $z_{n-1} + hf(x+H,z_n)$
  - $y_n = [z_n + z_{n-1} + hf(x+H,z_n)] / 2$

California State University Northridge 32

Bulirsch-Stoer Method

- Three main ideas
  - Use large step size H and compute results at x + H for several values of n then extrapolate results to h = 0
  - Use midpoint method whose truncation error is $Ah^n + Bh^{n+2} + Ch^{n+4} \ldots$ to improve accuracy of interpolation process
  - Use rational function approximation instead of simple polynomial interpolation for extrapolating to h = 0

California State University Northridge 33

Rational Function Approximation

- Like polynomial interpolation except that the ratio of two polynomials is used

$$y = f(x) \qquad y \approx R(x) = \frac{P_n(x)}{Q_m(x)} = \frac{\sum_{i=0}^{n} a_i x^i}{1 + \sum_{j=1}^{m} b_j x^j}$$

- Need n + m + 1 (x_k, y_k) data points to determine coefficients in polynomials
- Use process similar to divided-difference table to compute R(x) for one x

California State University Northridge 34

Some Method Details

- Divide interval x to x + H into n = 2, 4, 6, 8, 12, 16, 24, ($n_j = 2n_{j-2}$) substeps
- Use only last M (typically M = 7) steps in rational function interpolation
- Error estimate from rational function approximation used to stop substep sequence if desired error is obtained
- Have strategy for increasing or decreasing large step size H

California State University Northridge 35

Midterm Review I

- Differential equations with constant coefficients: $d^ny/dx^n + \alpha_1 d^{n-1}y/dx^{n-1} + \ldots + \alpha_n y = r(x)$: find $\lambda_1 \ldots \lambda_n$ as roots of equation $\lambda^n + \alpha_1\lambda^{n-1} + \cdots + \alpha_{n-1}\lambda + \alpha_{n-1} = 0$
  - r(x) = 0 gives homogenous solution, $y_H = C_1e^{\lambda_1 x} + C_2e^{\lambda_2 x} + \cdots C_ne^{\lambda_n x}$
  - Complex λ pairs give sine/cosine terms
  - Repeated real λ give $(C_1+C_2x+C_3x^2+\ldots)e^{\lambda x}$
  - For $r(x) \neq 0$ $y = y_H + y_P$

California State University Northridge 36

Midterm Review II

- For nonhomogeneous solutions find solution y = y_H + y_P
- To get particular solution, y_P
  - Write form for y_P, based on form for r(x)
  - Substitute postulated y_P with unknown constant(s) into particular equation
  - Equate coefficients of like terms to find unknown constants
  - Use y = y_H + y_P to find constants from homogenous solution from boundary values

California State University Northridge 37

Midterm Review III

- Bessel's Equation: $\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} + \frac{x^2-\nu^2}{x^2}y = 0$
- Solutions for integer-n/non-integer-ν are $y = AJ_n(x) + BY_n(x)/y = AJ_\nu(x) + BY_{-\nu}(x)$
- No questions on power series solutions or Frobenius method
- Use of Laplace transforms to solve differential equations
  - Use of partial fractions for inverse transforms

California State University Northridge 38

Midterm Exam

- Open book and notes, including homework solutions
- Make your own notes to use for exam
  - You are in trouble if you have to use the book on an open-book exam
- May be useful to have integral tables
- More credit given for showing how to obtain solution than for providing final details of algebra or arithmetic

California State University Northridge 39