PhD Course in Simulation

Course Meeting 3

Lars Eriksson

Associate Professor
Department of Electrical Engineering
Linköping University
`lars.erikssonk@liu.se`

2024-04-02

Today's Agenda

- Brief recap
  - Concepts: Consistency, convergence, and stability
  - Explicit single-step methods
- Perturbation analysis
  - Method independent
- Implicit single-step methods
- Multistep methods
  - Implicit
  - Explicit

Runge-Kutta Methods

$$
\begin{aligned}
Y_i = &\quad y_{n-1} + h \textstyle\sum_j a_{ij} f(t_{n-1} + c_j\, h, Y_j), \qquad 1 \le i \le s \\
y_n = &\quad y_{n-1} + h \textstyle\sum_j b_j f(t_{n-1} + c_j\, h, Y_j)
\end{aligned}
$$

The method in tableau form (Butcher (1964))

$$
\begin{array}{c|cccc}
c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\
c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}
$$

Explicit method if $a_{ij} = 0$ for $i \le j$.
Maximum attainable order of s-stage Explicit RK (ERK) methods

stages	1	2	3	4	5	6	7	8	9	10	11
order	1	2	3	4	4	5	6	6	7	7	8

Today's Agenda

- Brief recap
    - Concepts: Consistency, convergence, and stability
    - Explicit single-step methods
- Perturbation analysis
    - Method independent
- Implicit single-step methods
- Multistep methods
    - Implicit
    - Explicit

Perturbation Analysis

Applications in parameter estimation, sensitivity analysis, optimal control, shooting, multiple shooting for BVP.

- Consider an IVP with parameters $\boldsymbol{p}$

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{p}), \qquad \boldsymbol{y}(0) = \boldsymbol{c}$$

- Study a perturbation vector $\bar{\boldsymbol{p}} = \boldsymbol{p} + \phi$

$$|\bar{\boldsymbol{y}}(t) - \boldsymbol{y}(t)| \leq |P(t)\phi| + O(\phi^2), \qquad P = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{p}}$$

- Can obtain an estimate of the error by simulating a linear (time-varying) ODE

$$P' = \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}\right) P + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}$$

Implicit Runge-Kutta Methods

- Implicit methods, $a_{ij} \neq 0$ for $i \geq j$

  $$\begin{array}{c|cccc} c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\ c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

  Solve many systems of equations
- The methods are based on numerical integration

Implicit Runge-Kutta Methods

- Gauss methods: order $p = 2s$

  Simplest example: implicit midpoint method.

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3-2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

- Radau methods: $p = 2s - 1$. Include one endpoint of the interval.

  Example: backward Euler.

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

- Lobatto methods: $p = 2s - 2$. Include both endpoints of the interval.

  Example: trapezoidal method.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Stiffly Accurate

- A method with A nonsingular and $a_{sj} = b_j$ is called "stiffly accurate"
- Stiffly accurate gives stiff decay
- Radau: stiffly decaying
- Gauss and Lobatto: symmetric, A-stable, not stiffly decaying
- All implicit methods so far are A-stable

Singly Diagonally Implicit RK – SDIRK

- Solving the system of equations can be time consuming.
  Especially Radau has no zeros in the A matrix.
  An $sm \times sm$ system of equations to solve
- Design methods with special properties

$$Y_i - h\,\gamma\, f(t_{n-1} + c_i\, h, Y_i) = y_{n-1} + h \sum_{j=1}^{i-1} a_{ij}\, f(t_{n-1}, +c_j\, h, Y_j)$$

$$\begin{array}{c|cccc}
\gamma & \gamma & 0 & \cdots & 0 \\
c_2 & a_{2,1} & \gamma & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s,1} & a_{s,2} & \cdots & \gamma \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}$$

Now solve s systems of size $m \times m$.
With the same $(I - h\gamma J)^{-1}$.

Other Aspects

- Order reduction in the stiff limit
  SDIRK
- Dense output (interpolation)
  plotting, event detection, time delays

Linear Multistep Methods

Basic problem

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad t \geq 0$$

- Runge-Kutta methods

$$\begin{aligned} Y_i &= \quad y_{n-1} + h \textstyle\sum_j a_{ij} f(t_{n-1} + c_j\, h, Y_j), \qquad 1 \leq i \leq s \\ y_n &= \quad y_{n-1} + h \textstyle\sum_j b_j f(t_{n-1} + c_j\, h, Y_j) \end{aligned}$$

- Linear multistep methods

$$\sum_{j=0}^{k} \alpha_j \boldsymbol{y}_{n-j} = h \sum_{j=0}^{k} \beta_j \boldsymbol{f}_{n-j}$$

- Linear, since $\boldsymbol{f}$ enters linearly in the method

Linear Multistep Methods

- Linear multistep methods

$$\sum_{j=0}^{k} \alpha_j \boldsymbol{y}_{n-j} = h \sum_{j=0}^{k} \beta_j \boldsymbol{f}_{n-j}$$

- Explicit if $\beta_0 = 0$ otherwise implicit
- Assume the last k integration steps are the same size
- Polynomial interpolation is the basis for many methods

Polynomial Interpolation in Newton Form

$$\phi(t) = f[t_l] + \sum_{i=2}^{k} f[t_1, \dots, t_k] \prod_{j=1}^{k-1} (t - t_j)$$

Recursive definition of *divided differences*

$$f[t_l] = f(t_l), \qquad f[t_l, \dots, t_{l+i}] = \frac{f[t_{l+1}, \dots, t_{l+i}] - f[t_l, \dots, t_{l+i-1}]}{t_{l+i} - t_l}$$

$$\begin{array}{c|c|c|c|c|c}
t_0 & f[t_0] & & & & \\
t_1 & f[t_1] & f[t_0, t_1] & & & \\
t_2 & f[t_2] & f[t_1, t_2] & f[t_0, t_1, t_2] & & \\
t_3 & f[t_3] & f[t_2, t_3] & f[t_1, t_2, t_3] & f[t_0, t_1, t_2, t_3] & \\
t_4 & f[t_4] & f[t_3, t_4] & f[t_2, t_3, t_4] & f[t_1, t_2, t_3, t_4] & f[t_0, t_1, t_2, t_3, t_4]
\end{array}$$

The Adams Family

$$\sum_{j=0}^{k} \alpha_j \boldsymbol{y}_{n-j} = h \sum_{j=0}^{k} \beta_j \boldsymbol{f}_{n-j}$$

- Adams $\alpha_0 = 1$, $\alpha_1 = -1$ and $\alpha_j = 0$, $j \geq 1$
- Explicit Adams methods $\beta_0 = 0$ – Adams-Bashforth
- Implicit Adams methods $\beta_0 \neq 0$ – Adams-Moulton

Adams-Bashforth – Explicit Adams Methods $\beta_0 = 0$

- Interpolate $f(t)$ for time steps backward $t_{n-k}, \ldots, t_{n-1}$
- Integrate the polynomial from t_{n-1} to t_n

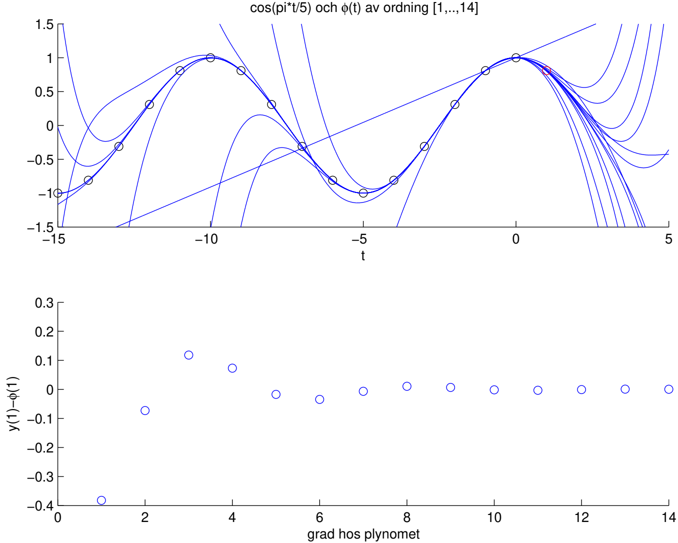
$$y_n = y_{n-1} + h \sum_{j=1}^{k} \beta_j f_{n-j}$$

Coefficients for the k-step method are computed as

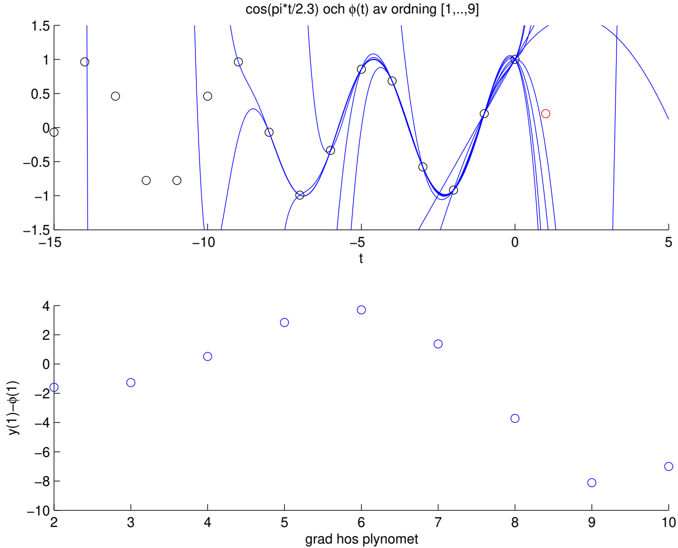
$$\beta_j = (-1)^{j-1} \sum_{i=j-1}^{k-1} \binom{i}{j-1} \gamma_i, \qquad \gamma_i = (-1)^i \int_0^1 \binom{-s}{i} ds$$

- The stability region decreases with k

Shrinking Stability Region – Possible Explanation



Shrinking Stability Region – Possible Explanation



Adams-Moulton – Implicit Adams Methods $\beta_0 \neq 0$

- Interpolate $f(t)$ for the time steps $t_{n-k}, \ldots, t_n$
- Integrate the polynomial from t_{n-1} to t_n

$$y_n = y_{n-1} + h \sum_{j=0}^{k} \beta_j f_{n-j}$$

- Order $p = k + 1$
- Absolute stability region larger than Adams-Bashforth but lies in the left half-plane (even though it is implicit). Not A-stable.
- The absolute stability region decreases with k

Backward Differentiation Formula (BDF) Family

- Backward differentiation (Notebook example)

$$\nabla^0 f_l = f_l$$

$$\nabla^i f_l = \nabla^{i-1} f_l - \nabla^{i-1} f_{l-1}$$

Backward Differentiation Formula (BDF) Family

- The Adams methods were built from a polynomial interpolating $f(t_{n-k}, y_{n-k})$ and integrated from y_{n-1} to y_n.
- BDF is built from a polynomial that interpolates previous values y_{n-k} and sets the backward derivative at t_n equal to $f(t_n, y_n)$.

$$\sum_{i=1}^{k} \frac{1}{i} \nabla^i y_n = h\, f(t_n, y_n)$$

- BDF is implicit with order $p = k$ normally implemented together with a modified Newton method.
- Unstable for $k > 6$.

Order, 0-Stability, Convergence

- Order – Use the difference operator.
- Stability analysis – Straightforward with time-discrete system.
  Characteristic polynomials.
  Principal root and *extraneous roots*.
- Stability and asymptotic stability – Poles inside the unit circle.
  0-stability, poles in the unit circle and possibly simple poles on the unit circle.
- *Strongly stable* all roots except $\xi = 1$ lie in the interior of the unit circle.
  Weakly stable 0-stable but not strongly stable.
- Absolute stability regions.
- Stiff decay.

Absolute Stability

- The absolute stability region decreases with increasing order.
- Few methods are A-stable; BDF of order 2 is the one with highest order.
- A-stability does not say everything about the method.
- Stiff decay is more important (especially for stiff systems).
- BDF trades A-stability for stiff decay.

Implementation of Multistep Methods

- Starting the methods.
- Implicit – Methods for solving systems of equations
- Both implicit and explicit – Step size change and error control

Starting Multistep Methods

- The methods need k old values.
- Start with single-step methods.
- Start with lower order methods within the family.

Equation Solving – Predictor-Corrector Methods

- Two methods that interact – One explicit and one implicit
- Initial guess and fixed-point iteration that is stopped

  P: $y_n^0 + \sum_{j=1}^{k} \hat{\alpha}_j y_{n-j} = h\,(\sum_{j=1}^{k} \hat{\beta}_j f_{n-j}), \quad \nu = 0$
  E: $f_n^\nu = f(t_n, y_n^\nu)$
  C: $y_n^{\nu+1} + \sum_{j=1}^{k} \alpha_j y_{n-j} = h\,(\beta_0 f_n^\nu + \sum_{j=1}^{k} \beta_j f_{n-j})$
- PEC, PECE, P(EC)$^\nu$, P(EC)$^\nu$E
  (Ending with E is preferred because the last function evaluation improves the interpolation polynomial and thus the method's stability)
- Belongs to a family of general linear methods
- Suitable for non-stiff problems

Equation Solving – Modified Newton Methods

- Implicit methods

$$y_n - h\beta_0 f(t_n, y_n) = -\sum_{j=1}^{k} \alpha_j y_{n-j} + h\,(\sum_{j=1}^{k} \beta_j f_{n-j})$$

Move to the form $F(t, y) = 0$

- Newton iteration $y^{\nu+1} = y^{\nu} - \frac{\partial F(t,y)}{\partial y}^{-1} F(t, y)$

$$y_n^{\nu+1} = y_n^{\nu} - \big(I - h\beta_0 \frac{\partial f}{\partial y}\big)^{-1} \Big[\sum_{j=0}^{k} a_j y_{n-j} - h\,(\sum_{j=0}^{k} \beta_j f_{n-j})\Big]$$

- LU factorize to solve the inverse and store it.
  Update $\frac{\partial f}{\partial y}$ and the LU factorization only when needed.

Step Size Change

The methods are based on fixed step size.

- Fixed coefficients
  Re-interpolate data to the new step size.
- Variable coefficients
  Derive a new method for nonuniform steps.
- Fixed first coefficient
  Predictor polynomial (nonuniform grid)

$$\phi(t_{n-i}) = y_{n-i}, \qquad 1 \leq i \leq k+1$$

a second polynomial $\psi(t)$ on a uniform grid

$$\psi(t_n - ih_n) = \phi(t_n - i\,h_n), \qquad 1 \leq i \leq k$$

$$\psi'(t_n) = f(t_n, \psi(t_n)), \qquad y_n = \psi(t_n)$$

Closing

Tolerance and error control – Straightforward
Software tips – Read
Note that DASSL, DASPK were developed by the authors.