Simulation

Course Meeting 2

Lars Eriksson

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

2024-04-02

Solutions and Approximations

- IVP

$$\begin{aligned} \boldsymbol{y}' &= \boldsymbol{f}(t, \boldsymbol{y}), \qquad 0 \leq t \leq b \\ \boldsymbol{y}(0) &= \boldsymbol{c} \end{aligned}$$

- Solution to IVP

$$\boldsymbol{y}(t)$$

- Approximation on a mesh

$$0 = t_0 < t_1 < t_2 < \ldots < t_N = b$$

$$\boldsymbol{c} = \boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N$$

- Step length in the n-th step

$$h_n = t_n - t_{n-1}$$

The Three Simplest Methods

- Forward Euler

$$\boldsymbol{y}_n = \boldsymbol{y}_{n-1} + h_n \boldsymbol{f}(t_{n-1}, \boldsymbol{y}_{n-1})$$

  Explicit method

- Backward Euler

$$\boldsymbol{y}_n = \boldsymbol{y}_{n-1} + h_n \boldsymbol{f}(t_n, \boldsymbol{y}_n)$$

  Implicit method

- Trapezoidal method

$$\boldsymbol{y}_n = \boldsymbol{y}_{n-1} + \frac{h_n}{2} \left(\boldsymbol{f}(t_{n-1}, \boldsymbol{y}_{n-1}) + \boldsymbol{f}(t_n, \boldsymbol{y}_n) \right)$$

  Implicit method (symmetric)

In the following analyses of the methods, it is assumed that $\boldsymbol{f}$ is sufficiently many times continuously differentiable.

Local and Global Error of a Method

- A method can be expressed using a difference operator $\mathcal{N}_h$

$$\mathcal{N}_h \boldsymbol{y}_h(t_n) = 0, \text{ with } \boldsymbol{y}_0 = \boldsymbol{c}$$

- E.g. Forward Euler

$$\mathcal{N}_h \boldsymbol{y}_h(t_n) = \frac{\boldsymbol{y}_n - \boldsymbol{y}_{n-1}}{h_n} - \boldsymbol{f}(t_{n-1}, \boldsymbol{y}_{n-1})$$

- Local truncation error
  The error when the operator is applied to the exact solution

$$d_n = \mathcal{N}_h \boldsymbol{y}(t_n)$$

- Global error

$$\boldsymbol{e}_n = \boldsymbol{y}_n - \boldsymbol{y}(t_n)$$

Consistency of a Method

- Consistency – accuracy
- The difference operator approximates the differential operator
- Accuracy $O(h^p)$ – order p
- Consistency – Local property

Convergence

- Let the mesh satisfy

$$h = \max_{1 \leq n \leq N} h_n$$

  and assume that Nh is bounded independent of N. (This means that the mesh cannot be refined only locally).
- A method is *convergent of order p* if the global error $\boldsymbol{e}_n = \boldsymbol{y}_n - \boldsymbol{y}(t_n)$ with $\boldsymbol{e}_0 = 0$, satisfies

$$\boldsymbol{e}_n = O(h^p)$$

- Convergence – Global property

0-Stability

- The difference method $\mathcal{N}_h$ is 0-stable if there exist positive constants h_0 and K such that

$$|\boldsymbol{x}_n - \boldsymbol{z}_n| \leq K \left\{ |\boldsymbol{x}_0 - \boldsymbol{z}_0| + \max_{1 \leq j \leq N} |\mathcal{N}_h \boldsymbol{x}_h(t_j) - \mathcal{N}_h \boldsymbol{z}_h(t_j)| \right\}$$

  holds for arbitrary mesh functions x_h, z_h and $h \leq h_0$.
- Measures how a perturbation affects a method's result.
- Also provides information about how the method behaves as $h \to 0$.
- (0-Stability is sometimes called D-stability)

Convergence Again

- Convergence is the property we want to have.
- Theorem:

$$\text{consistency} + \text{0-stability} \;\Rightarrow\; \text{convergence}$$

  Moreover, if the method is consistent of order p, then it is also convergent of order p
- Consistency and convergence are related via 0-stability
- (Motivates the introduced concepts)

The Test Equation

- The test equation

$$y' = \lambda\, y, \qquad\qquad \lambda \in \mathbb{C}$$

- When applying a numerical method to a continuous-time system, one obtains a discrete-time system
- Determination of the absolute stability region for the method
- A numerical method can be viewed as a transformation between continuous-time and discrete-time systems

Absolute Stability

- The condition for absolute stability

$$|y_n| \leq |y_{n-1}|$$

  (compare with absolute convergence from analysis)
- Apply a method with step size h to the test equation
- The region in the complex plane $z = \lambda\, h$ where

$$|y_n| \leq |y_{n-1}|$$

  is called the absolute stability region
- Regions for:

Forward Euler	$\lvert 1 + h\,\lambda \rvert \leq 1$
Backward Euler	$\frac{1}{\lvert 1 - h\,\lambda \rvert} \leq 1$
Trapezoidal method	$\left\lvert \frac{2+\lambda\, h}{2-\lambda\, h} \right\rvert \leq 1$

Implementation of the Methods

- Test on: $y' = -10\,y, \qquad y(0) = 1$
- Stability region
- Global error
- Order

	Euler f			Euler b			Trapez.		
h	t	e_n	p	t	e_n	p	t	e_n	p
1	0	9		0	0.091		0	0.67	
0.1	10	4.5e-005	5.3	10	0.00093	2.0	10	2.8e-005	4.4
0.01	90	1.9e-005	0.4	50	2.7e-005	1.5	70	3.8e-007	1.9
0.001	230	2.2e-006	0.9	501	2.3e-006	1.1	701	3.8e-009	2.0
0.0001	1893	2.3e-007	1.0	5067	2.3e-007	1.0	7100	3.8e-011	2.0
1e-005	18978	1.8e-008	1.1	49781	2.4e-008	1.0	75789	4.5e-009	-2.1
1e-006	187209	1.8e-009	1.0	427295	1.3e-009	1.3	630807	4.5e-010	1.0

A-Stability

- It would be good if the method preserves the problem's properties.
- If the problem is stable, then it is good if the method also gives a stable solution.
- A-stability:
  If the absolute stability region includes the left half-plane, then the method is said to be A-stable.

The examples show two *problems* with A-stability.

- The circle example shows the connection between problem and stability region.
  Choose the right method for the problem.
- The example with control towards $\cos(t)$ also shows that certain methods with large stability region do not always give the expected behavior.

Stiff Problems

A definition:
An IVP is stiff in an interval [0,b] if the step length needed to maintain stability with forward Euler is much smaller than that needed to have an accurate representation of the solution.

- Stiffness depends on
  - the differential equation
  - initial values
  - time scale
  - the method's absolute stability region

Stiff Decay, Stiff Limit

- Consider the system

$$y' = \lambda(y - g(t))$$

  where $\mathrm{Re}(\lambda) << 0$.
- The system consists of a fast mode and a slow mode.
- If the method has

$$\lim_{h_n \,\mathrm{Re}(\lambda) \to -\infty} (y_n - g(t_n)) = 0$$

  then it is said to have "stiff decay".
- Examples clearly show the property.

Non-Smooth Problems

- Example: Bouncing ball
- Property from the theory:
  Continuation to the boundary
- Switching equation and equation solver to find the correct switching point
- Not necessarily sufficient to rely on step size adaptation

Single-Step Methods

- Taylor approximation, approximates the function using the derivative
- Approximate the derivative with function evaluations.
- Runge-Kutta methods
- So far we have worked with three RK methods.
- Have also seen that the order seems to improve performance.

Taylor's methods

- Know $f(t, y)$ and its derivatives
- Taylor expand $y(t)$ in a step with length h.

Runge-Kutta methods

- Approximate the derivatives using function evaluations

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

- An s-stage method

Runge-Kutta Methods

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

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

Order of Runge-Kutta Methods

- Numerical method y_n is determined by y_{n-1}.

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

- Consider $\hat{y}(t)$ which is the solution to

$$\hat{y}' = f(t, \hat{y}), \qquad \hat{y}(t_{n-1}) = y_{n-1}$$

- Compare the Taylor expansions of y_n and $\hat{y}(t_n)$ around t_{n-1}.
- Identify the terms in the Taylor expansions and set them equal up to $O(h^p)$
- Order p
- What do the conditions mean? Extra conditions that facilitate $c_i = \sum_j a_{ij}$ $-Y_i$ gets *correct* values in each internal stage for $y' = 1$.

Order of Runge-Kutta Methods

- The methods contain many parameters/degrees of freedom
- The number of conditions that must be considered increases with the order

order	1	2	3	4	5	6	7	8	9	10
# conditions	1	2	4	8	17	37	85	200	486	1205

- Can check what order a method has
- *Cannot* design a method
- There are also several families of the same order...

Error Estimation and Step Size Control

- Want a certain accuracy in the solution.
- The global error is difficult, the local error is easier.
- Standing at t_{n-1} should choose h_n in the next step.
- Control the step size so that the local error is constant during the integration

$$|\boldsymbol{l}_k| \approx \text{ETOL}$$

often choose $|\boldsymbol{l}_k| \leq C \cdot \text{ETOL}$ with $C < 1$ to ensure the error tolerance.

- The components in $\boldsymbol{y}$ sometimes have different orders of magnitude

$$|(l_j)_n| \leq C \cdot [\text{ATOL}_j + |(y_j)_n|\text{RTOL}]$$

Controls absolute error tolerance ATOL_j and relative error tolerance RTOL.

- The local truncation error is complicated
  Example: A family of second-order ERK methods

$$h\,d_n = \frac{h^3}{6}\left[\frac{3}{4\gamma}(f_{yy}f^2 + 2f_{ty}f + f_{tt}) - y'''\right] + O(h^4)$$

- Do not want to depend on f_{yy} etc

Error Estimation and Step Size Control

Fundamental idea for step size methods

- Calculate two solutions y_n and $\hat{y}_n$ at t_n
- $|\hat{y}_n - y_n|$ gives an estimate of the error for the least accurate method
- If $|\hat{y}_n - y_n| \geq C \cdot \text{ETOL}$ then the step is rejected and h is decreased.
- If the order is p, then the new step $\tilde{h}$ can be chosen according to

$$(\frac{\tilde{h}}{h})^{p+1}|\hat{y}_n - y_n| \approx C \cdot \text{ETOL}$$

- If the step is accepted, then one can also use the formula to increase the step size.
- How should we choose y_n and $\hat{y}_n$?

Step Size Control – Embedded Methods

- Use a pair of methods that give y_n and $\hat{y}_n$ with order p and $p+1$
- Search for an s-stage method that has order $p+1$ such that there exists another method with order p embedded in it, i.e. the embedded method uses the same computational stages.

$\boldsymbol{c}$	A
	$\hat{\boldsymbol{b}}$

$\boldsymbol{c}$	A
	$\boldsymbol{b}$

$\boldsymbol{c}$	A
	$\boldsymbol{b}$
	$\hat{\boldsymbol{b}}$

- Simplest example: forward Euler and modified trapezoidal method

0	0	0
1	1	0
	1	0
	$\frac{1}{2}$	$\frac{1}{2}$

Error Estimation – Two Pairs

- Fehlberg 4(5) pair: 6-stage.
  Gives a method of order 4 with error estimation.

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	-7200/2197	7296/2197			
1	439/216	-8	3680/513	-845/4104		
1/2	-8/27	2	-3544/2565	1859/4104	-11/40	
	25/216	0	1408/2565	2197/4104	-1/5	0
	16/135	0	6656/12825	28561/56430	-9/50	2/55

- The constants are chosen so that the local error in y_n is minimized.

Error Estimation – Two Pairs

- Dormand and Prince 4(5) pair: 7-stage.
  The last stage is the same as the first in the next step
  computational cost same as 6 stages.
  Gives a method of order 4 with error estimation.

0							
1/5	1/5						
3/10	3/40	9/40					
4/5	44/45	-56/15	32/9				
8/9	19372/6561	-25360/2187	64448/6561	-212/729			
1	9017/3168	-355/33	46732/5247	49/176	-5103/18656		
1	35/384	0	500/1113	125/192	-2187/6784	11/84	
	5179/57600	0	7571/16695	393/640	-92097/339200	187/2100	1/40
	35/384	0	500/1113	125/192	-2187/6784	11/84	0

- The constants are chosen so that the local error in $\hat{y}_n$ is minimized.
- This method is now most common.

Error Estimation – Step Doubling

- Calculate y_n in two ways: once with h, and once with $h/2$.
- Gives accurate estimation of the local error.
- Local error is estimated better than with embedded methods.
- The method is general and can be applied to arbitrary methods.
- Costs more to use step doubling than embedded methods.

Error Estimation – Global Error

- Cumbersome to estimate the global error.
- The user who specifies the global error may not have exact knowledge of what global error is needed.
- Local error may be sufficient.

For Next Time

- Read up to 4.6 on page 95.
- Next time:
  - implicit single-step methods
  - linear multi-step methods