ZS 7

Freitag, 11. Januar 2019 15:48

7. Numerical solution of ODE's

$$\left|\begin{aligned}\dot{y}_1(t) &= 3y_1(t) + 4y_2(t) - y_3(t)\\ \dot{y}_2(t) &= y_1(t) + y_3(t)\\ \dot{y}_3(t) &= 4y_2(t) + t^2\end{aligned}\right| \Leftrightarrow \left|\begin{aligned}\dot{y}_1(t) &= f_1(t, y_1, y_2, y_3)\\ \dot{y}_2(t) &= f_2(t, y_1, y_2, y_3)\\ \dot{y}_3(t) &= f_3(t, y_1, y_2, y_3)\end{aligned}\right| \Leftrightarrow \dot{\underline{y}} = \underline{f}(t, \underline{y})$$

Goal: Find $y_1(t)$, $y_2(t)$ and $y_3(t)$ so that they fulfil the ODE.

z.B. $y_1(t) = 4t^2 + t$, $y_2(t) = e^{2t}$, $y_3(t) = t + \cos(e^t) + 5$

Note: If t^2 wouldn't be there the System would be autonomous.

First order: Only $\dot{y}$ appears. (No $\ddot{y}$, $\dddot{y}$, ...)
Higher order: When $\ddot{y}$, $\dddot{y}$ etc. appear.

We can transform any Higher order ODE in a first order ODE.
↳ And transform any non-autonomous in a autonomous one.

$$\left|\begin{aligned}\ddot{y}_1(t) &= 3\dot{y}_1(t) + \ddot{y}_2(t) + t^2\\ \ddot{y}_2(t) &= 3\dot{y}_2(t) + y_1(t)\end{aligned}\right| \Rightarrow \left|\begin{aligned}\dot{z}_1(t) &= z_2(t)\\ \dot{z}_3(t) &= z_4(t)\\ \dot{z}_2(t) &= 3\cdot z_2(t) + \dot{z}_4(t) + z_0^2(t)\\ \dot{z}_4(t) &= 3\dot{z}_4(t) + z_1(t)\\ \dot{z}_0(t) &= 1\end{aligned}\right|$$

Operator: Encodes the complete exact set of solutions of our ODE: Φ

Discrete Evolution operator: Is the approximation of the true operator: Ψ

$$\Rightarrow \Psi(h, y) \approx \Phi^h y$$

Polygonal Approximation Methods (Single Step Methods)

We want to now actually find $y(t)$ that fulfills the ODE.
Here we want a model to approximate the Operator Φ

Note: If we know the initial values we properly just want to know what values we will have in t time.

Note: The following Methods assume an ODE of first order in the form: $\dot{y}(t) = f(y(t))$

Convergence: All Single step Methods have algebraic convergence. The question is only of what order: linear, quadratic, ... relativ to the width

Explicit Euler:

$$\begin{cases} y_0 = y(0)\\ y_{k+1} = y_k + (t_{k+1} - t_k) f(y_k)\end{cases}$$

Convergence: If stepsize halfs, the error halfs.
↳ linear Dependence of error on stepsize
↳ First Order method
↳ Algebraic convergence

Implicit Euler:

$$\begin{cases} y_0 = y(0)\\ y_{k+1} \quad y_k + h f(y_{k+1})\end{cases}$$

⊕ Better Stability in case of stiffness
⊖ Need to solve LSE or potentially N-LSE each step.

⇒ Algebraic convergence, First order Method (linear)

Implicit midpoint:

$$\begin{cases} y_0 = y(0)\\ y_{k+1} = y_k + h\cdot f\left(\tfrac{1}{2}(y_k + y_{k+1})\right)\end{cases}$$

⇒ Algebraic convergence, Second order Method (quadratic)

Runge Kutta:

Order	1	2	3	4	5	6	7	8	≥ 9
stages	1	2	3	4	6	7	9	11	$\geq +3$

Convergence: Of course algebraic but the Constant grows:

- exp. fast in the length of the interval T
- exp. fast in the Lipschitz constant of f
- linearly in $\max_{\tau} \|\dot{y}(\tau)\|$

Note: Rapidly varying function require much smaller stepsizes (t_{k+1}
In Practice you will do adaptive timestepping.

Runge-Kutta-2-Method:

- 2-step-Method.
- SST-Method of order 2

$$y_{k+1} = y_k + h f\left(t_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} f(t_k, y_k)\right)$$

General form of Runge-Kutta-s-Method:

$$k_1 = f\left(t_k + c_1 h, y_k + h\sum_{j=1}^{s} a_{1j}\cdot k_j\right) \qquad c_i = \sum_{j=1}^{s} a_{ij}$$

$$\vdots \qquad \vdots$$

$$k_s = f\left(t_k + c_s h, y_k + h\sum_{j=1}^{s} a_{sj}\cdot k_j\right)$$

$$y_{k+1} = y_k + h\sum_{i=1}^{s} b_i\cdot k_i$$

Note: The a, b and c's are stored in a Butcher tableau.:

For Runge-Kutta to be consistent we need:

$$\sum_{i=1}^{s} b_i = 1 \quad \text{and} \quad \sum_{j=1}^{s} a_{ij} = c_i$$

Note: Explicit Runge-Kutta means no LSE hence in Butcher tableau A is lower triangular

Stiffness / Stability

Not the solution to the ODE, but the ODE itself.

For the ODE: $\dot{y} = \lambda y$

In order to be stable we need that $|y_{k+1}| < |y_k|$ (b.c. solution is
We can introduce a stability function $S(z)$ such that:

$$y_{k+1} = S(\lambda h)\, y_k \iff y_{k+1} = \Psi^h y_k$$

And we need that $|S(\lambda h)| < 1$. Only then it is stable.

Explicit Euler: We will need that $h < \frac{2}{|\lambda|}$

Runge-Kutta-s-Method: $\dfrac{\det(I - zA + z\mathbb{1}b^T)}{\det(I - zA)} = S(z) \overset{!}{<} 1 \qquad z = \lambda$

Note: For explizit Runge-Kutta is $\det(I - zA) = 1$

⇒ Just keep λh small.

For the linear ODE: $\dot{y} = My$ (and M diagonizable)

Runge-Kutta-s-Method:

stabel $\iff |S(\lambda_i h)| < 1$ where S is the Stability f
from above and λ_i the Eigenvalues of M.

⇒ There is no clear definition for stiffness. You more o
It is stiff if you have a large negative Eigenval

Note: Runge-Kutta method is A-stable if the stabi
region includes all negative (in the Realteil) comple
↳ You want to choose an A-stable method for problems.