Stability regions of Runge-Kutta methods

Stephan Houben
Eindhoven University of Technology

February 19, 2002

Overview of the talk

1. Quick review of some concepts
2. Stability regions
3. L-stability and the increment function
4. One-step methods as numerical low-pass filters

Quick review of some concepts

- Runge-Kutta methods
- Butcher tableau
- Increment function (stability function)

Runge-Kutta methods

Also called: linear one-step methods.
Given an Ordinary Differential Equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{1}$$

we compute $\boldsymbol{x}_{n+1}$ from $\boldsymbol{x}_n$ with only the following operations.

1. Vector addition: $\boldsymbol{u} + \boldsymbol{v}$,
2. Scalar-vector multiplication: $a\boldsymbol{u}$,
3. Solving the unknowns $\boldsymbol{y}_i$, $1 \leq i \leq N$, from the system

   $$\boldsymbol{a}_i + \sum_j b_{ij}\boldsymbol{y}_j = \boldsymbol{f}(\boldsymbol{c}_i + \sum_j d_{ij}\boldsymbol{y}_j), \tag{2}$$

   for $1 \leq i \leq N$.

That's all we are ever allowed to do if we don't want to make any special assumptions on $\boldsymbol{x}_0$ or $\boldsymbol{f}$.

Butcher tableau

Generic way to describe any Runge-Kutta method

$$\begin{array}{c|ccc} c_1 & a_{11} & a_{12} & a_{13} \\ c_2 & a_{21} & a_{22} & a_{23} \\ c_3 & a_{31} & a_{32} & a_{33} \\ \hline & b_1 & b_2 & b_3 \end{array}$$

$$\begin{aligned} \boldsymbol{k}_1 &= \boldsymbol{f}(\boldsymbol{x}_n + a_{11}h\boldsymbol{k}_1 + a_{12}h\boldsymbol{k}_2 + a_{13}h\boldsymbol{k}_3), \\ \boldsymbol{k}_2 &= \boldsymbol{f}(\boldsymbol{x}_n + a_{21}h\boldsymbol{k}_1 + a_{22}h\boldsymbol{k}_2 + a_{23}h\boldsymbol{k}_3), \\ \boldsymbol{k}_3 &= \boldsymbol{f}(\boldsymbol{x}_n + a_{31}h\boldsymbol{k}_1 + a_{32}h\boldsymbol{k}_2 + a_{33}h\boldsymbol{k}_3), \end{aligned}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + b_1 h\boldsymbol{k}_1 + b_2 h\boldsymbol{k}_3 + b_3 h\boldsymbol{k}_3.$$

Increment function (stability function)

We look at the model problem

$$\dot{x} = \lambda x, \tag{3}$$

and consider the relation between x_n and x_{n+1}.

For any “reasonable” (i.e. scaling-invariant) one-step method, this relation must have the form

$$x_{n+1} = \zeta(h\lambda) x_n. \tag{4}$$

1. x_{n+1} depends linearly on x_n, i.e. the dimension in which we express x (meters, inches, lightyears) doesn't influence the answer.

2. Changing the units in which we measure time changes h and λ, but not the dimensionless quantity $h\lambda$. Therefore, to be scaling-invariant in the time, ζ can only depend on $h\lambda$.

Increment function of exact solution

Note that for the *exact* solution, we have

$$\boldsymbol{x}_{n+1} = e^{h\lambda}\boldsymbol{x}_n, \tag{5}$$

and thus

$$\zeta(z) = e^z. \tag{6}$$

Some increment functions

Euler forward: $\zeta(z) = 1 + z$

Euler backwards: $\zeta(z) = \frac{1}{1-z}$

Trapezoidal rule: $\zeta(z) = \frac{2+z}{2-z}$

Stability regions

The iteration

$$\boldsymbol{x}_{n+1} = \zeta(h\lambda)\boldsymbol{x}_n \tag{7}$$

is divergent for $|\zeta(h\lambda)| > 1$, convergent otherwise.

The (absolute) stability region is the set

$$\{h\lambda \in \mathbb{C} \,|\, |\zeta(h\lambda)| \leq 1\}, \tag{8}$$

i.e. the complete original of the unit circle under ζ.

The relative stability region (or *Order star*) is the set

$$\{h\lambda \in \mathbb{C} \,|\, |\zeta(h\lambda)| \leq |e^{h\lambda}|\}. \tag{9}$$

The relative stability region compares the growth of the iteration to the growth of the exact solution $e^{t\lambda}\boldsymbol{x}_0$.

Important points to note

1. Relative stability diagram is always stable far to the right, and always unstable far to the left.

2. Relative stability diagram shows “fingers” around the origin.
   Because of the star-like shapes of these fingers we call them *order stars*.

3. Stable (light-blue) fingers contain a zero of $\zeta(z)$.

4. Unstable (dark-blue) fingers contain a pole of $\zeta(z)$.

Why the name “order star”?

Suppose $\zeta(z)$ is an approximation to e^z of order p, i.e.

$$e^z = \zeta(z) + Cz^{p+1} + O(z^{p+2}). \tag{10}$$

From this we have

$$\frac{\zeta(z)}{e^z} = 1 - Cz^{p+1} + O(z^{p+2}), \quad C \neq 0. \tag{11}$$

We take a small $\varepsilon > 0$ and take $z = \varepsilon e^{\phi}$.

As ω goes from 0 to 2π, Cz^{p+1} winds $p+1$ times around 0. Thus $\frac{\zeta(z)}{e^z}$ winds $p+1$ times around 1.

Therefore we have $p+1$ stable and $p+1$ unstable regions in any sufficiently small neighbourhood of the origin.

Stability concepts

1. A-stable: stable if exact solution is stable and ODE is non-stiff

$$|\zeta(z)| < 1 \quad \text{for } \mathrm{Re}(z) < 0.$$

2. A(α)-stable: weaker than A-stable

$$|\zeta(z)| < 1 \quad \text{for } \mathrm{Re}(z) < -\tan\alpha|\mathrm{Im}(z)|.$$

3. I-stable: the imaginary axis is stable

4. L-stable: stability concept for stiff ODE's and DAE's

L-stability

We look at the model ODE

$$\dot{x} = \lambda x \tag{12}$$

and wonder what happens with the increment function ζ as $\lambda \to -\infty$.

For the exact solution, we have

$$\zeta_{\mathtt{exact}}(z) = e^z \to 0 \quad \text{for } z \to -\infty. \tag{13}$$

For Euler forward we have:

$$\zeta_{\text{EF}}(z) = 1 + z \to -\infty \quad \text{for } z \to -\infty. \tag{14}$$

For Euler backward we have:

$$\zeta_{\text{EB}}(z) = \frac{1}{1-z} \to 0 \quad \text{for } z \to -\infty. \tag{15}$$

For Trapezoidal rule we have:

$$\zeta_{\text{TR}}(z) = \frac{2+z}{2-z} \to -1 \quad \text{for } z \to -\infty. \tag{16}$$

Of the methods shown, only Euler Backward is L-stable.

A closer look at the increment function

For a Runge-Kutta method, the increment function ζ is of the form

$$\zeta(z) = \frac{p(z)}{q(z)}, \tag{17}$$

i.e. a rational function in z.

For an explicit method, $q(z) = 1$. This means that the stability region of an explicit method is a bounded set.

A Runge-Kutta method is L-stable if and only if $\deg(p(z)) < \deg(q(z))$. The stability region of an L-stable method is a neighborhood of ∞.

One-step methods as numerical low-pass filters

Oscillations in the computed solution can be divided in two categories.

1. Real oscillations that correspond to oscillations in the exact solution.

2. Numerical artefacts.

When oscillatory solutions have to be found, we want to suppress 2 and conserve 1. Euler Backward does a good job at suppressing 2, but also suppresses 1.

Therefore

We want a method that acts as a “numerical low-pass filter”, suppressing high frequencies and leaving low frequencies intact. Note that such a method is L-stable.