Stability and Convergence of Numerical Integration Methods

VNVe 2019/2020

1 Introduction

Let's solve an ordinary differential equation (ODE) - an initial value problem

$$y' = \lambda y, \quad y(0) = 1, \quad \lambda < 0, \tag{1}$$

using explicit and implicit numerical methods (Euler methods, Trapezoidal rule, Taylor series methods).

Well-known analytic solution of the ODE (1) is in the form

$$y = e^{\lambda t}. \tag{2}$$

2 Stability and convergence of numerical methods

Numerical method is *absolute stable* if and only if the local truncation error is not increasing in computation of the next values $y_k,\ k > n$ for given integration step h and for given ODE.

Definition 2.1 *The sequence of values y_i (aproximation using numerical methods) must converge to exact solution $y(t_i)$.*
Existence of limit is expected

$$\lim_{h \to 0, i \to \infty} (y_i) = y(t_i).$$

The **Stability condition** in the form must be accepted for our example (1).

$$|y_{i+1}| \leq |y_i|. \tag{3}$$

Then the **Stability function** of numerical method is defined in a form

$$R(z) = \frac{y_{i+1}}{y_i}, \quad \text{where} \quad z = h\lambda, \tag{4}$$

supposing that the constant λ is generally a complex number $z \in \mathcal{C}$.

The **(absolute)**[1] **Stability domain** of numerical method is defined in the form

$$D = \{z \in \mathcal{C}; |R(z)| \leq 1\} \,. \tag{5}$$

2.1 Explicit numerical methods

Stability domain of explicit Euler method and Taylor series method will be now analysed.

2.1.1 Explicit Euler method

Well-known explicit Euler method is in the form

$$y_{i+1} = y_i + hy'_i \tag{6}$$

after substitution the ODE (1) into (6)

$$y_{i+1} = y_i + h\lambda y_i = (1 + h\lambda)y_i = (1 + h\lambda)^i y_0 \,, \tag{7}$$

where $y_0 = y(0) = 1$.

To apply the stability condition (3), the following definition

$$|1 + h\lambda| \leq 1 \,, \tag{8}$$

must be accepted.

<u>Classification of the stability of the Euler method</u>

Let's $z = h\lambda$, then the unit circle $|z + 1| \leq 1$ Fig. 1 (highlighted part) of the complex plain with the centre (-1,0) represents the absolute stable domain (5) of the explicit Euler method.

The explicit Euler method is not of a large stability domain. The explicit Euler method can not be used to solution of "stiff" systems - ODE (1) where $|\lambda| \gg 1$.

2.1.2 Explicit Taylor method

The explicit Taylor series is in the form

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \frac{h^3}{3!}y'''_i + \cdots + \frac{h^n}{n!}y_i^{(n)} \,, \tag{9}$$

$$y_{i+1} = y_i + DY1_i + DY2_i + \cdots + DYn_i, \quad ORD = n \,. \tag{10}$$

[1]The term "Absolute" stability domain of the numerical method can be found in some literature.

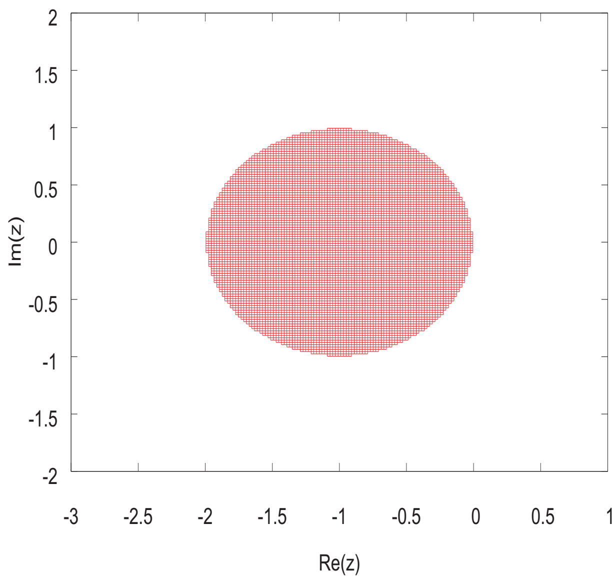

Figure 1: The stability domain of the explicit Euler method

where $DY1_i, DY2_i, \cdots, DYn_i$ are Taylor series terms and the abbreviation ORD means the order of Taylor series method (respectively the number of Taylor series terms used during the computation).

Note: The Taylor series terms (10) can be calculated recurrently in the form

$$\begin{aligned} DY1_i &= h\lambda y_i\,, \\ DY2_i &= \tfrac{h}{2}\lambda DY1_i\,, \\ &\vdots \\ DYn_i &= \tfrac{h}{n}\lambda DY(n-1)_i\,. \end{aligned}$$

The higher derivatives of the ODE (1) can be calculated analytically in the form

$$\begin{aligned} y' &= \lambda y\,, \\ y'' &= \lambda y' = \lambda^2 y\,, \\ &\vdots \\ y^{(n)} &= \lambda^n y\,. \end{aligned}$$

thus with the respect to (9) we have

$$
\begin{aligned}
y_{i+1} &= y_i + h\lambda y_i + \tfrac{h^2}{2}\lambda^2 y_i + \tfrac{h^3}{3!}\lambda^3 y_i + \cdots + \tfrac{h^n}{n!}\lambda^n y_i\,, \\
y_{i+1} &= (1 + h\lambda + \tfrac{h^2}{2}\lambda^2 + \tfrac{h^3}{3!}\lambda^3 + \cdots + \tfrac{h^n}{n!}\lambda^n) y_i\,.
\end{aligned}
$$

Homework:
– *Determine the stability function R(z) = ? for the explicit Taylor series of the ORD = 2,3,4,10.*
– *Plot the graphs of the stability domains D = ? with the GNUplot software (use command - 'load C: \...\GNUplotStability.gnu').*

2.2 Stability domains of numerical methods

Definition 2.2 *Dahlquist 1963: A method, whose stability domain D satisfies*

$$D \supset \mathcal{C}^- = \{z \in \mathcal{C}; \Re e(z) \le 0\}\,,$$

is called **A-stable**.

The stability domain of A-stable numerical method is overlaying the whole left half-plane of complex plain $\mathcal{C}^- = \{z \in \mathcal{C}; \Re e(z) < 0\}$ see Fig. 2.

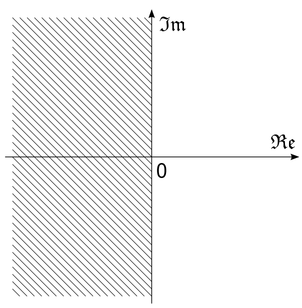

Figure 2: The stability domain of A-stable numerical method

Definition 2.3 *Ehle 1969: A method is called* **L-stable**, *if it is A-stable and if in addition*

$$\lim_{\Re e(z) \to -\infty} R(z) = 0\,.$$

The L-stable numerical methods are suitable to stiff systems solution.

2.3 Implicit numerical methods

Let's find stability domains for implicit Euler method, Trapezoidal rule and implicit Taylor series method.

2.3.1 Implicit Euler method

The implicit Euler method is in the form

$$y_{i+1} = y_i + hy'_{i+1}\,. \tag{11}$$

Substitute the ODE (1) into (11) and obtain the form

$$y_{i+1} = y_i + h\lambda y_{i+1}\,. \tag{12}$$

The stability function of implicit Euler method is obtained after some modification

$$R(z) = \frac{1}{1-z}\,. \tag{13}$$

Note: that z is generally complex number $z = a + ib$, where $a < 0$ and $R(z) = \frac{1}{(1-a-ib)}$.

When $|R(z)| \leq 1$ and $\lim\limits_{a\to-\infty} R(z) = 0$ than the implicit Euler method is L-stable. The stability domain of implicit Euler method is in Fig. 3.

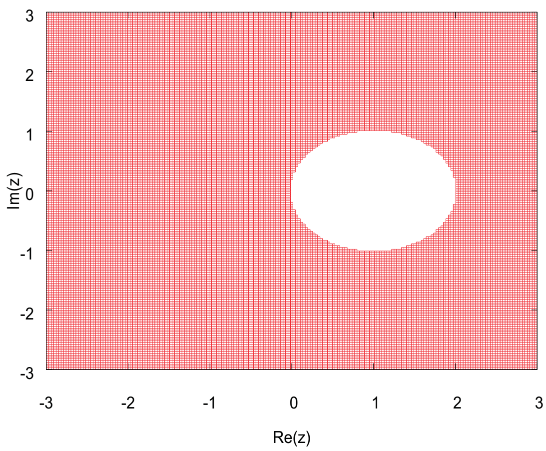

Figure 3: The stability domain - implicit Euler method

2.3.2 Implicit trapezoidal rule

The implicit trapezoidal rule is in the form

$$y_{i+1} = y_i + \frac{h}{2}(y'_i + y'_{i+1}) \,. \tag{14}$$

again for the ODE (1) we have

$$y_{i+1} = y_i + \frac{h}{2}(\lambda y_i + \lambda y_{i+1}) = \frac{y_i(1 + h\lambda/2)}{1 - h\lambda/2} \,. \tag{15}$$

Homework:
– Determine the stability function R(z) = ? for the implicit Trapezoidal rule.
– Plot the graph of the stability domains D = ? with the GNUplot software (use command - 'load C: \...\GNUplotStability.gnu').
– Is the Trapezoidal rule A-stable numerical method?
– BONUS QUESTION: Is the Trapezoidal rule L-stable numerical method (see Def. 2.3))?

2.3.3 Implicit Taylor series

The implicit Taylor series is in the form

$$\begin{aligned} y_{i+1} &= y_i + hy'_{i+1} - \frac{h^2}{2!}y''_{i+1} + \frac{h^3}{3!}y'''_{i+1} - \cdots - \frac{(-h)^n}{n!}y^{(n)}_{i+1} \,, && (16) \\ y_{i+1} &= y_i - DY1_{i+1} - \cdots - DYn_{i+1}, \quad ORD = n \,. && (17) \\ & && (18) \end{aligned}$$

The higher derivatives of the ODE (1) can be calculated analytically in the form

$$\begin{aligned} y'_{i+1} &= \lambda y_{i+1} \,, \\ y''_{i+1} &= \lambda y'_{i+1} = \lambda^2 y_{i+1} \,, \\ &\vdots \\ y^{(n)}_{i+1} &= \lambda^n y_{i+1} \,. \end{aligned}$$

thus with respect to (16) we have

$$\begin{aligned} y_{i+1} &= y_i + h\lambda y_{i+1} - \tfrac{h^2}{2!}\lambda^2 y_{i+1} + \tfrac{h^3}{3!}\lambda^3 y_{i+1} - \cdots - \tfrac{(-h)^n}{n!}\lambda^n y_{i+1} \,, \\ y_{i+1} &= \left(\frac{1}{1 - h\lambda + \frac{h^2}{2!}\lambda^2 - \frac{h^3}{3!}\lambda^3 + \cdots + \frac{(-h)^n}{n!}\lambda^n}\right) y_i \,. \end{aligned}$$

Newton's method
The well-known Newton's iteration method (also known as Newton-Raphson method) is used to find the roots of a function

$$x : f(x) = 0.$$

The Newton-Raphson method in one variable is implemented as follows: Given a function f defined over the reals x, and its derivative f', we begin with a first aproximation x_0 for a root of the function f. Provided the function satisfies all the assumptions made in the derivation of the formula, a better approximation x_{j+1} is

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)}\,,$$

The process is repeated until sufficiently accurate value TOL is reached

$$|x_{j+1} - x_j| < TOL\,.$$

Implicit Taylor series - Newton's iteration method
Let's solve ODE (1) using implicit Taylor series (16). The Taylor series terms (17) can be calculated recurrently in the form

$$\begin{aligned}
DY1_{i+1} &= -h\lambda y_{i+1}\,, \\
DY2_{i+1} &= -\tfrac{h}{2}\lambda DY1_{i+1}\,, \\
&\vdots \\
DYn_{i+1} &= -\tfrac{h}{n}\lambda DY(n-1)_{i+1}\,.
\end{aligned}$$

The Newton's iteration is in the form

$$y_{i+1,j+1} = y_{i+1,j} - \frac{f(y_{i+1,j})}{f'_y(y_{i+1,j})}\,, \tag{19}$$

where

$$f(y_{i+1,j}) = -y_{i+1,j} + y_i - DY1_{i+1,j} - DY2_{i+1,j} - \cdots - DYn_{i+1,j}\,. \tag{20}$$

As a starting itteration $y_{i+1,0}$

$$y_{i+1,0} = y_i$$

is used.

The derivation $f'_y(y_{i+1,j})$ can be computed **symbolically** or using **differential formulae**.

Symbolic computation of the derivative $f'_y(y_{i+1,j})$
Let's derive (20) according to $y_{i+1,j}$

$$f'_y(y_{i+1,j}) = -1 - DY1'_{i+1,j} - DY2'_{i+1,j} - \cdots - DYn'_{i+1,j}\,,$$

where Taylor series terms are in the form

$$\begin{aligned} DY1'_{i+1,j} &= -h\lambda\,, \\ DY2'_{i+1} &= -\tfrac{h}{2}\lambda DY1'_{i+1,j}\,, \\ &\vdots \\ DYn'_{i+1,j} &= -\tfrac{h}{n}\lambda DY(n-1)'_{i+1,j}\,. \end{aligned}$$

Differential formulae used for computation of derivative $f'_y(y_{i+1,j})$
The derivation $f'_y(y_{i+1,j})$ can be also computed using differential formula

$$f'_y(y_{i+1,j}) = \frac{f(y_{i+1,j} + h_N) - f(y_{i+1,j})}{h_N}\,,$$

where

$$f(y_{i+1,j}+h_N) = -(y_{i+1,j}+h_N)+y_i-DY1_{N,i+1,j}-DY2_{N,i+1,j}-\cdots-DYn_{N,i+1,j}\,.$$

Taylor series terms are in the form

$$\begin{aligned} DY1_{N,i+1,j} &= -h\lambda(y_{i+1,j}+h_N)\,, \\ DY2_{N,i+1,j} &= \tfrac{(-h)^2}{2!}\lambda^2(y_{i+1,j}+h_N) = \tfrac{-h}{2}\lambda DY1_{N,i+1,j}\,, \\ &\vdots \\ DYn_{N,i+1,j} &= \tfrac{(-h)^n}{n!}\lambda^n(y_{i+1,j}+h_N) = \tfrac{-h}{n}\lambda DY(n-1)_{N,i+1,j}\,. \end{aligned}$$

Homework:
– Determine the stability function R(z) = ? for the implicit Taylor series of the ORD = 2,3,4,10.
– Plot the graphs of the stability domains D = ? with the GNUplot software (use command - 'load C: \...\GNUplotStability.gnu').

3 Numerical solution - MATLAB

Homework:

– Run the script for numerical solution of ODE (1) in MATLAB software. Start first with explicit numerical methods (script: "explicit.m"). See a simple implementation of explicit numerical methods (m-files: "eul.m", "tay.m"). Let the integration step size be the same for all numerical computations $h = 0.1$*. Set the constant lambda* $= -10$ *(then variable* $z = -1$*). Verify the stability and convergence of numerical computations from plotted graphs. Then select the constant* λ *on the boundary of the stability domain of Euler explicit method and observe the stability and convergence of numerical computations.*

– Run the script for implicit numerical solution of ODE (1) in MATLAB software (script: "implicit.m"). See a simple implementation of implicit numerical methods (m-files: "impl_eul.m", "impl_trap.m", "impl_tay.m"). Let the integration step size be the same for all numerical computations $h = 0.1$*. Set the constant lambda* $= -10$ *(then variable* $z = -1$*). Verify the stability and convergence of numerical computations from plotted graphs. Increase the absolute value of the constant* $|\lambda| > 10$ *and observe the stability and convergence of numerical solutions. See the behaviour of the Trapezoidal rule (A-stable, but not L-stable numerical method).*