Supplementary Notes 10

Tiago Salvador (tiago.saldanhasalvador@mail.mcgill.ca)

Abstract

In this tutorial we discuss A-stability of numerical methods for IVPs. We also discuss predictor-correct methods and Runge-Kutta methods.

Contents

1 A-stability

Besides the (obvious) error we have by solving IVP approximately, there's also another type of error that exists when solving IVPs using finite difference methods (for instance Euler methods, Adams-Bashforth methods, Adams-Moulton methods). This error comes from not performing the computations exactly. In practice, neither the initial conditions nor the arithmetic that is performed is represented exactly because of the round-off error associated with finite-digit arithmetic. We then need our method to be able to handle this small perturbations, which means that we want to produce correspondingly small changes in the subsequent approximations. Methods with this property are called stable.

We study the concept of A-stability. We consider the test problem

$$\begin{cases} y'(t) = \lambda y(t) \\ y(0) = y_0 \end{cases} \tag{1}$$

where $\lambda \in \mathbb{C}$ with $\mathrm{Re}(\lambda) \leq 0$ and solve it with both methods. Here $\mathrm{Re}(\lambda)$ denotes the real part of the complex number λ.

Remark 1.1. *A problem as* (1) *with Re*(λ) *very negative it's called a stiff problem. They are usually tougher to handle than most other problems.*

We start by noticing that the exact solution to our test problem is $y(t) = y_0 e^{\lambda t}$ which has the following property:

$$y(t) \to 0 \quad \text{when } t \to \infty.$$

Naturally, we expect our numerical methods to capture this behaviour, i.e., we want

$$y_n \to 0 \quad \text{when } n \to \infty.$$

Let's take a look at an example.

Example 1.1. *Applying the forward Euler method to our test problem* (1) *leads to*

$$y_{n+1} = y_n + hf(t_n, y_n) = y_n + h\lambda y_n = (1 + h\lambda)y_n$$

Hence it follows that

$$y_n = (1 + h\lambda)^n y_0.$$

Therefore, in order to have $y_n \to 0$ *when* $n \to \infty$, *we need* $|1 + h\lambda| < 1$

Assume, for now, that $\lambda \in \mathbb{R}$. *Then,* $|1 + h\lambda| < 1$ *implies that* $h < -2/\lambda$. *The more negative* λ *is the stiffer the test problem becomes and the smaller we have to choose our time step* h.

In the general case when $\lambda \in \mathbb{C}$, *we have*

$$|1 + h\lambda| < 1 \Longleftrightarrow (1 + x)^2 + y^2 < 1$$

where $h\lambda = x + iy$ *(see Figure 1).*

Remark 1.2. *We obtain the same restriction on* h *if we look to impose that a small error in the initial guess, does not lead to arbitrarily large errors in subsequence approximations. Suppose then that a small error, say due to round-off, is introduce in the initial condition, that is, instead of* y_0, *we have* $y_0 + \delta_0$. *Then at the* n^{th} *step the round-off error is*

$$\delta_n = (1 + h\lambda)^n \delta_0$$

Hence, we need to choose h *such that* $|1 + h\lambda| < 1$, *which leads to* $h < -2/\lambda$ *if* $\lambda \in \mathbb{R}$.

This example motivates us to consider the following form of stability.

Definition 1.3. *For fixed* h, *let* y_n *be the solution at* $t = nh$ *of a given method when applied to* (1). *The region of A-stability (or absolute stability) of the method is*

$$R = \left\{ h\lambda \in \mathbb{C} : \lim_{n\to\infty} |y_n| = 0, \text{ for a fixed } h \right\}$$

A method is A-stable if R *contains the entire left half of the complex plane. Otherwise, the method is conditionally stable.*

Remark 1.4. *If a method is A-stable then the region* $Re(h\lambda) \leq 0$ *is a subset of* R *which means that there is not restriction on* h.

Example 1.2. *We study the stability of the backward Euler method. We have*

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) = y_n + h\lambda y_{n+1}$$

and so

$$y_{n+1} = \frac{1}{1 - h\lambda} y_n.$$

This leads to

$$y_{n+1} = \frac{y_0}{(1 - h\lambda)^n}.$$

Hence, we want $|1 - h\lambda|^{-1} < 1$. *If* $\lambda \in \mathbb{R}$ *this is always true since* $h > 0$. *If* $\lambda \in \mathbb{C}$, *then*

$$\frac{1}{|1 + h\lambda|} < 1 \Longleftrightarrow (1 - x)^2 + y^2 < 1$$

where $h\lambda = x + iy$ *(see Figure 1). This means that, unlike the forward Euler method, the backward Euler is A-stable: we are not required to take very small time steps when solving very stiff problems.*

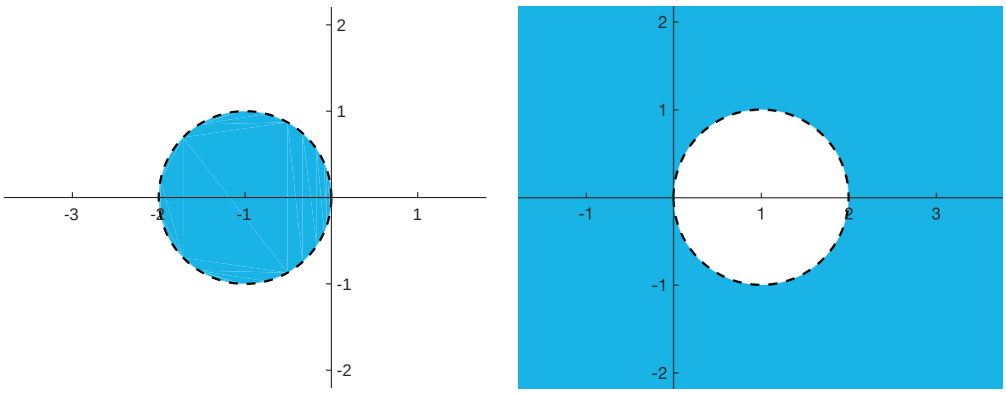

Figure 1: A-stability regions for the forward Euler (left) and backward Euler (right)

It can be shown that in fact all explicit methods are conditionally stable and in general implicit methods have bigger regions of A-stability (see Figure (3)).

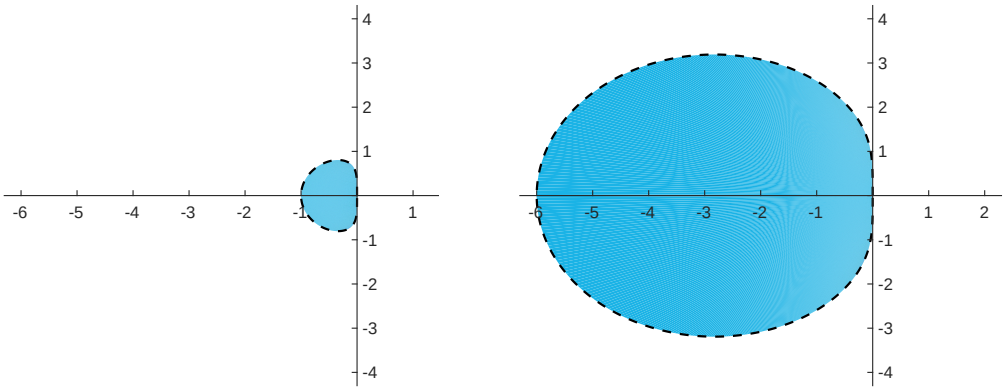

Figure 2: A-stability regions for the 2-step Adam-Bashforth method (left) and the 2-step Adam-Moulton method (right).

In practice, when solving an IVP with a A-stable method, we still need to take h small for two main reasons: accuracy since the error is $\mathcal{O}(h^p)$; to ensure convergence of the root finding methods which are required when using an implicit method (if h is too large, the root finding methods may not converge.

Remark 1.5 (no free-lunch)**.** *In general, implicit methods have bigger regions of stability and so, in particular, they perform better when handling stiff problems. However, at each step some root finding has to be done, making them more expensive computationally. As for the explicit methods they typically have restrictions for h but are faster.*

We can generalize the idea of A-stability to problems of the form

$$y'(t) = f(y(t)).$$

An equilibrium solution y^* to this problem satisfies $f(y^*) = 0$ with $f'(y^*) < 0$. We can Taylor expand the equation near any equilibrium solution y^* to obtain

$$y' = f(y) = f'(y^*) + f'(y^*)(y - y^*) + \mathcal{O}(|y - y^*|^2).$$

Then as $y(t) \to y^*$ with $t \to \infty$, the I.V.P. becomes approximately

$$y' \approx f'(y^*)(y - y^*) \quad \text{or} \quad \tilde{y}' \approx f(y^*)\tilde{y}$$

where $\tilde{y} = y - y^*$. Thus we are back at the situation of the test problem for $\tilde{y}$ with $\lambda = f'(y^*)$.

2 Predictor-Corrector methods

Each step of an implicit method requires the use of a root finding method (e.g. secant or Newton's method) which complicates the procedure considerably. The alternative is then to improve approximations obtained by explicit methods with an implicit method. The combination of an explicit method to predict and an implicit to improve the prediction is called a predictor-correct method.

Example 2.1 (improved Euler's method)**.** *Using the forward Euler as the predictor method and the trapezoid method as the correct method, leads to*

$$\begin{aligned} \tilde{y}_{n+1} &= y_n + hf(t_n, y_n) && \textit{(predict)} \\ y_{n+1} &= y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})) && \textit{(correct)} \end{aligned}$$

Notice that this method can be rewritten as

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_{n+1}, y_n + hf(t_n, y_n)) + f(t_n, y_n)\right)$$

which is a nonlinear explicit 1*-step method. Using the Taylor expansion, it is possible to show that is has order* 2*.*

In a nutshell, for a given implicit method, the idea of a predictor-corrector method is the following:

- use an explicit method to "predict" y_{n+1}
- "correct" y_{n+1} using the implicit method

In practice the correction part is repeated to improve the accuracy: provide h is small enough, only a few iterations are required in practice. The idea works because the iterated correction can be seen as a fixed point iteration to find y_{n+1} in the implicit method.

Example 2.2. *We can generalize the improved Euler's method by repeating* k *times the correction part. This leads to*

$$\begin{aligned} \tilde{y}_{n+1}^{(0)} &= y_n + hf(t_n, y_n) \\ \tilde{y}_{n+1}^{(1)} &= y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1}^{(1)})\right) \\ &\vdots \\ \tilde{y}_{n+1}^{(k)} &= y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1}^{(k-1)})\right) \\ y_{n+1} &= \tilde{y}_{n+1}^{(k)} \end{aligned}$$

Alternatively, we can write it as the fixed point iteration $\tilde{y}_{n+1}^{(k)} = g\left(\tilde{y}_{n+1}^{(k-1)}\right)$ *where*

$$g(x) = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, x)\right)$$

3 Runge-Kutta methods

Explicit 1-step methods are fast and easy to implement but have low order accuracy. One idea to generalize to higher order is to take fractional time steps and compute y_{n+1} in "stages". These are called Runge-Kutta (RK) methods. We provide two examples.

Example 3.1 (improved Euler's method as a RK method)**.** *We can write the improved Euler as a RK method:*

$$y_{n+1} = y_n + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)$$

where

$$k_1 = f(t_n, y_n) \quad \text{and} \quad k_2 = f(t_n + h, y_n + hk_1).$$

where $k_1 = f(t_n, y_n)$ *and* $k_2 = f(t_n + h, y_n + hk_1)$.

Definition 3.1 (RK method of order four)**.**

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{1}{2}k_2\right) \\ k_2 &= hf\left(t_{n+1}, y_n + k_3\right) \end{aligned}$$

This method is the most popular RK method and it is known as RK4. It is implemented in MATLAB in the ODE45 function.

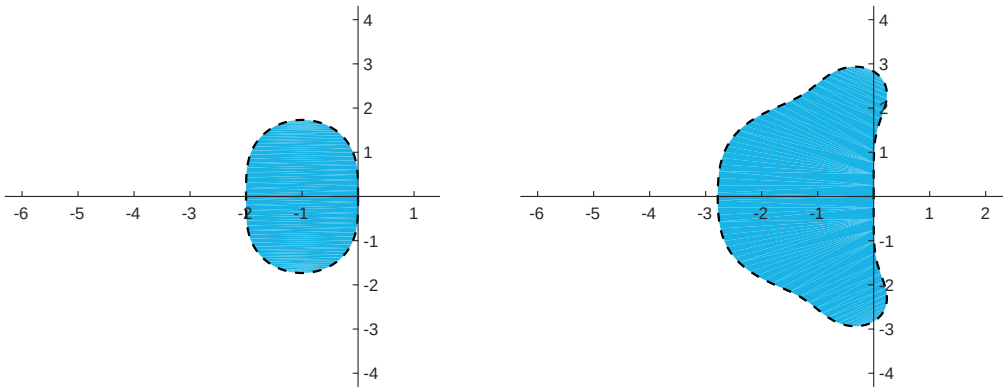

Figure 3: A-stability regions for the improved Euler's method (left) and RK4 method (right).

Remark 3.2. *The RK methods are conditionally stable but tend to have larger regions of A-stability than other explicit methods, like forward Euler or Adam-Bashforth method.*

References

[1] R. L. Burden and J. D. Faires. *Numerical Analysis*. 9^{th} edition. Brookes/Cole, 2004.

[2] A. Quarteroni, R. Sacco and F. Saleri. *Numerical Mathematics*. 2^{nd} edition. Springer, 2006.

[3] Tutorial notes written by Jan Feys

[4] Class notes written by Andy Wan for Math 317 Fall 2014.