Briefly on the stability of methods for numerically solving ordinary differential equations

October 22, 2015

During lectures I demonstrated how Euler's method, a first-order Runge–Kutta method, behaved when solving

$$y'(x) = -3y(x).$$

The solution seemed decent enough with a step size of $h = 1/4$, but with $h = 1$ it oscillated wildly and never settled down. This is in stark contrast to the analytical/exact solution $y(x) = y(0)e^{-3x}$. The witnessed behavioral difference between the numerical and the exact solution is one definition of a numerical method being *unstable* (for a given step size).

1 A test for stability

One can say something about the stability of numerical methods for ODEs in general by studying the differential equation

$$y'(x) = \lambda y(x), \tag{1}$$

where $y(0) \neq 0$ and λ is a *complex number* with strictly negative real part, i.e. $\operatorname{Re}\lambda < 0$.

We know that equation (1) has the exact solution

$$y(x) = y(0)e^{\lambda x} = y(0)e^{x \operatorname{Re}\lambda}e^{ix \operatorname{Im}\lambda}. \tag{2}$$

Since $\operatorname{Re}\lambda < 0$, the solution will be dampened by the first exponential factor, and we therefore have

$$\lim_{x\to\infty} y(x) = 0.$$

One measure of stability is then to demand that that our numerical solution, denoted $y_0, y_1, \dots, y_n, \dots$, behaves in the same way when we solve equation (1), i.e. we want

$$\lim_{n\to\infty} y_n = 0.$$

In our context, we will say that a method is *stable* (for a given λ and a given step size) if $\lim_{n\to\infty} y_n = 0$. We have seen during lectures that Euler's method is *not* stable in the case when $\lambda = -3$ and $h = 1$.

2 The stability regions of Runge–Kutta methods

The above definition of stability can be applied to and investigated for any Runge–Kutta method. In the following we consider equation (1) with $\lambda \in \mathbb{C}$ and $\operatorname{Re}\lambda < 0$.

2.1 Euler's method

For equation (1), Euler's method becomes

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

Applied $n \geq 0$ times, we find

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

From complex analysis we know that $\lim_{n\to\infty} y_n = 0$, with y_n as above, if and only if

$$|1 + h\lambda| < 1.$$

This holds if and only if $h\lambda$ (which is a complex number) lies in a disk of radius 1 centered at $-1 + 0\mathrm{i}$ in the complex plane.

Suppose we are given a λ (with $\operatorname{Re}\lambda < 0$). In order to be guaranteed stability, we must thus choose a step size h that is small enough that $h\lambda$ lies inside said disk in the complex plane. It could happen that we must choose h tiny, with the downside that we have to do very many steps to get to the $y_n = y(x_0 + nh)$ we are interested in.

2.2 Other Runge–Kutta methods

The analysis above can be carried out for all Runge–Kutta methods, including the four that we have seen in the course: Euler, improved Euler (Heun), RK4, and backwards Euler. Figure 1 shows the result, i.e. the *regions of stability* for the various methods. You are encouraged to do the analysis above on your own for all these methods (especially for backwards Euler, for reasons that will soon become clear).

Notice how *backwards Euler* is stable for $h\lambda$ in the entire left half plane (gray area in figure 1). A method with this property is said to be *absolutely stable*. With absolutely stability, the solution we get from the method will converge to 0 (as the exact one does) nomatter how large a step size we choose! Such a stability property is typical of implicit methods, of which backwards Euler is the only example we have covered.

The other methods, which are explict ones, have finite stability regions. Also in those cases there are differences to note: If λ has a real part very close to 0 (corresponding to the exact solution in equation (2) being very slow in its decay) and a large imaginary part (corresponding to a highly oscillating exact solution), then we must choose *h extremely small* for $h\lambda$ to end up in Euler's method's (blue) or Heun's method's (turquoise) stability regions. On the other hand, RK4's region (red) is somewhat easier to reach with a larger h.

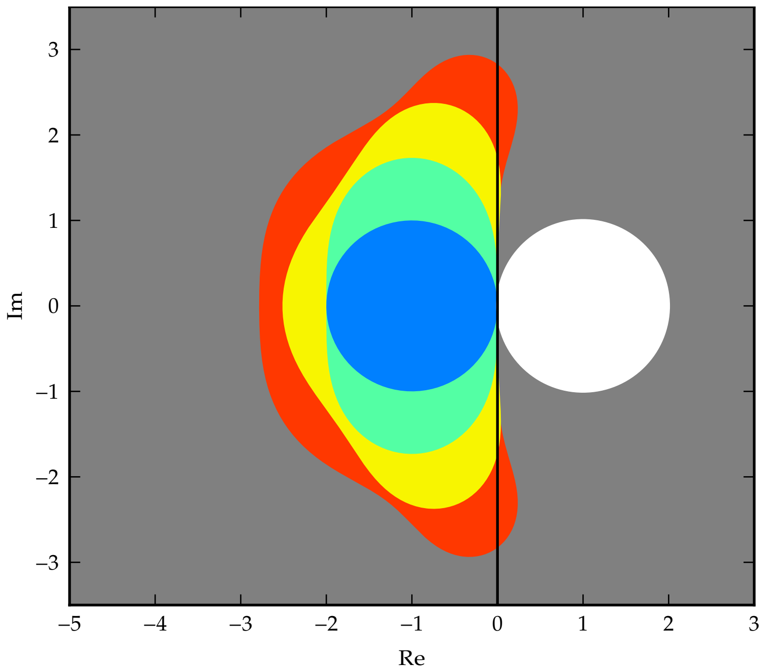

Figure 1: Stability regions for some Runge–Kutta methods. With $h\lambda$ in the **gray region** *backwards Euler* is stable, in the **red region** *RK4* is stable, in the **yellow region** a third-order RK method (which we have not looked at) is stable, in the **turquoise/green region** *improved Euler (Heun)* is stable, and in the **blue region** *Euler* is stable (as we just showed). Note that some of the red cover the gray, some of the yellow cover the red, and so on.