Implicit-Explicit Methods for ODEs

Varun Shankar

January 28, 2016

1 Introduction

We have discussed several methods for handling *stiff* problems; in this situations, we concluded it was better to use an implicit time-stepping method. On the other hand, for ODE problems of the form $y' = \lambda y$, where λ is purely imaginary, we covered two points: first, that many implicit methods may add an unwanted amount of dissipation to this problem (due their stiff decay properties or other damping properties), and second, that some high-order explicit methods (like AB3, RK3 and RK4) actually have stability regions that encompass the imaginary axis.

Now we ask ourselves the following question: what if the ODE (or system of ODEs) has a stiff part and an imaginary part? Such an IVP is given below:

$$\frac{dy}{dt} = \underbrace{f(t, y)}_{\textit{Non-stiff}} + \underbrace{g(t, y)}_{\textit{Stiff}}, \tag{1}$$

$$y(t_0) = y_0. \tag{2}$$

Using a purely implicit method may be acceptable in this scenario, or depending on the problem, may dissipate the imaginary part to an unwanted extent. Further, one could envision a situation where the imaginary part is nonlinear and the stiff part linear; in this scenario, using an implicit method is the worst of both worlds: we end up solving a nonlinear problem, *and* it isn't one we want to solve (possibly due to dissipation). On the other hand, explicit methods are often unacceptable for stiff problems due to the limited size of their stability regions.

This tension between the types of solvers used for stiff and imaginary ODE problems can be resolved using a class of methods called Implicit-Explicit methods, or IMEX methods. In this chapter, we will discuss some of these IMEX schemes. While a full analysis of their stability is beyond the scope of this text, we will attempt to provide summaries where relevant. We will focus primarily on IMEX multistep methods.

2 IMEX Multistep Methods

While combining IMEX multistep methods, we typically pick an AM method for the stiff part, and an AB method for non-stiff (possibly imaginary) part; it is also common to pick both schemes of the same order, to ensure that the overall truncation error is not lowered by either scheme.

A typical and extremely popular time integration scheme of this type is Crank-Nicolson (Trapezoidal rule) Adams-Bashforth, often called CNAB or ABCN. The second-order CNAB scheme is given as

$$y^{n+1} = y^n + \Delta t \left[\frac{3}{2} f(t_n, y^n) - \frac{1}{2} f(t_{n-1}, y^{n-1})\right] + \frac{\Delta t}{2}\left[g(t_{n+1}, y^{n+1}) + g(t_n, y^n)\right] \tag{3}$$

Notice that this uses the Crank-Nicolson philosophy of trying to approximate the solution at the middle of the time-step $t_{n+\frac{1}{2}}$. This makes the AB2 scheme quite appropriate, since AB2 attempts to extrapolate f to that same time level (see previous notes).

Another popular scheme is called Crank-Nicolson Leapfrog (CNLF). The second order CNLF scheme is given by:

$$y^{n+1} = y^{n-1} + 2\Delta t f(t_n, y^n) + \frac{\Delta t}{2}\left[g(t_{n+1}, y^{n+1}) + g(t_{n-1}, y^{n-1})\right]. \tag{4}$$

Notice that instead of attempting to approximate the stiff term at $t_{n+\frac{1}{2}}$, we now approximate it at t_n. Thus, there is no need to extrapolate f in time! This scheme has been used fairly often in conjunction with spectral methods to solve PDEs that have both hyperbolic and parabolic nature (a stiff part and an imaginary part).

Another scheme, possibly the best second-order scheme for stiff problems with a non-dominant imaginary part, is built around the BDF2 scheme. To understand how this scheme is constructed, recall the BDF philosophy: we evaluate the ODE right hand side only at t_{n+1}, and use multiple levels in y. Now, in the case of our new ODE IVP, we can do this to g straightforwardly. What about f? How do we get it to level t_{n+1} without doing things implicitly? Well, all we need to do is achieve a second-order extrapolation of f, just like AB2 does, albeit to t_{n+1}. This is easily accomplished as $f(t_{n+1}, y^{n+1}) \approx 2f(t_n, y^n) - f(t_{n-1}, y^{n-1})$. Plugging this in together with the BDF2 scheme, we get the Semi-implicit BDF2 scheme (SBDF2), given by:

$$y^{n+1} = \frac{4}{3} y^n - \frac{1}{3} y^{n-1} + \frac{2}{3}\Delta t\left[2f(t_n, y^n) - f(t_{n-1}, y^{n-1})\right] + \frac{2}{3}\Delta t g(t_{n+1}, y^{n+1}). \tag{5}$$

It is also possible to generate higher-order generalizations of these schemes. However, we must be cautious: recall the shrinking stability regions of the higher-order AB methods! In such a scenario, some brave souls may opt to use an AB-AM predictor-corrector for f, and an AM or BDF method for g. However, it is more common to stick with the second order IMEX multistep methods. When going to higher-order, SBDF3 is still pretty good in terms of its stability properties, since the BDF3 scheme contains a small portion of the imaginary axis in its stability region.

3 IMEX Runge-Kutta Methods

While the analysis and presentation of IMEX RK methods is beyond the scope of this class, they follow roughly the same philosophy: pick an explicit RK scheme for f, and an implicit one for g. Further, when selecting an implicit scheme for g, it is common to select a *diagonally*-implicit RK (DIRK) method, so the non-zeros in the upper triangle of the RK matrix A are purely on the diagonal. This gives greater savings over a fully-implicit RK scheme for g, at the cost of possible order reduction for certain problems.