ODE: ONE STEP METHODS

OSCAR REULA

1. One Step Methods

We are trying to approximate the solutions to the equation system,

$$\dot{u} = f(u) \tag{1}$$

where $u = u(t)$, is a curve en R^n, and $f(u)$ is a vector field in some open set U of R^n. In case of an equation system for which $f = f(u,t)$, we simply define $\tilde{u} = (u,t)$ and $\tilde{f} = (f,1)$. That is, we extend the system for one equation. The initial conditions at $t = t_0$ are also extended, $\tilde{u}_0 = (u_0, t_0)$.

One-step methods are those where the values of the approximation at a time $t + \Delta t$ are obtained from the values of the approximation at t. No further information from previous values are used. That is we have an algorithm of the form:

$$v^{n+1} = v^n + \Delta t A_p(v^n, \Delta t) \tag{2}$$

Where A_p is the algorithm. The index p characterize the truncation error of the method. For instance for Euler's method,

$$Euler_1(v^n, \Delta t) = f(v^n) \tag{3}$$

For the mid-point rule,

$$MidPoint_2(v^n, \Delta t) = f(v^n + \frac{\Delta t}{2} f(v^n))$$

For Heun's method,

$$Heun_2(v^n, \Delta t) = \frac{1}{2}[f(v^n) + f(v^n + \Delta t f(v^n))]$$

All Runge Kutta algorithms, as well as the Taylor ones, are of this type.

2. Truncation error

The truncation error is the error for which the algorithm approximates the equation when applied to a smooth solution, in other contexts is called the **residual**.

$$e_T := \frac{u^{n+1} - u^n}{\Delta t} - A_p(u^n, \Delta t). \tag{4}$$

We can compute it expanding all terms in Taylor series. Notice that this in particular implies smoothness of u and f. In the notation that follows $u^n = u(t_n)$, where t_n is the time at step n. For uniform time-steps $t_n = n * \Delta t$.

The first term is general,

$$\frac{u^{n+1}-u^n}{\Delta t} = \dot{u}^n + \ddot{u}^n\frac{\Delta t}{2} + \dddot{u}^n\frac{\Delta t^2}{6} + \mathcal{O}(\Delta t^3)$$

To it we can apply the equation, and use the chain rule. For instance, $\dot{u} = f(u)$, $\ddot{u} = \dot{f}(u) = f_u\dot{u} = f_u f$. This way we get rid of all time derivatives, resulting in an expression that only depends on u at that time t_n.

$$\frac{u^{n+1}-u^n}{\Delta t} = f(u^n) + f_u(u^n)f(u^n)\frac{\Delta t}{2} + [f_{uu}(u^n)f^2(u^n) + f_u^2(u^n)f(u^n)]\frac{\Delta t^2}{6} + \mathcal{O}(\Delta t^3)$$

The second term depends on the algorithm. For instance, for the Euler method, and keeping only the first two terms above we get,

$$\begin{aligned} e_T &= \frac{u^{n+1}-u^n}{\Delta t} - f(u^n) \\ &= f(u^n) + f_u(u^n)f(u^n)\frac{\Delta t}{2} - f(u^n) \\ &= f_u(u^n)f(u^n)\frac{\Delta t}{2} + \mathcal{O}(\Delta t^2) \end{aligned}$$

So, for a generic smooth $f(u)$ the local truncation error is first order ($p = 1$).

It is clear from the construction that the truncation error is a function of the solution and its derivatives at the given time, or, once the derivative have been substituted using the equation, just a functions of u at that time. We have an expression for the truncation error (or residual) in terms of f and its derivatives evaluated at the solutions. Thus, in many cases if we know f in a region, we can estimate that truncation error on a whole region.

For the mid-point method we get:

$$\begin{aligned} e_T &= \frac{u^{n+1}-u^n}{\Delta t} - f(u^n + \frac{\Delta t}{2}f(u^n)) \\ &= f(u^n) + f_u(u^n)f(u^n)\frac{\Delta t}{2} + [f_{uu}(u^n)f^2(u^n) + f_u^2(u^n)f(u^n)]\frac{\Delta t^2}{6} \\ &- [f(u^n) + f_u(u^n)\frac{\Delta t}{2}f(u^n) + f_{uu}(u^n)\frac{\Delta t^2}{8}f(u^n)^2] + \mathcal{O}(\Delta t^3) \\ &= [f_{uu}(u^n)\frac{1}{24}f(u^n)^2 + f_u^2(u^n)\frac{1}{6}f(u^n)]\Delta t^2 + \mathcal{O}(\Delta t^3) \end{aligned}$$

Thus, we see that the truncation error is order 2.

Exercise: Find all methods of second order of the form:

$$\begin{aligned} q_1 &= f(v^n) \\ q_2 &= f(v^n + \Delta t\alpha_{21}q_1) \\ RK_2(v^n, \Delta t) &= (A_1q_1 + A_2q_2). \end{aligned}$$

Hint: imposing second order should give the conditions: $A_1 + A_2 = 1$, and $A_2\alpha_{21} = \frac{1}{2}$.

Exercise: Compute the truncation error for Runge Kuta of order four,

$$\begin{aligned} q_1 &= f(v^n) \\ q_2 &= f(v^n + \frac{\Delta t}{2} q_1) \\ q_3 &= f(v^n + \frac{\Delta t}{2} q_2) \\ q_4 &= f(v^n + \Delta t q_3) \\ RK_4(v^n, \Delta t) &= \frac{1}{6}(q_1 + 2q_2 + 2q_3 + q_4). \end{aligned}$$

Exercise: For a function $f(u,t)$ the generic RK method is of the form:

$$\begin{aligned} q_1 &= f(v^n, t) \\ q_2 &= f(v^n + \alpha_{21} q_1, t + \beta_2) \\ q_3 &= f(v^n + \alpha_{31} q_1 + \alpha_{32} q_2, t + \beta_3) \\ \cdots & \quad \cdots \\ q_n &= f(v^n + \alpha_{n1} q_1 + \cdots + \alpha_{n(n-1)} q_{n-1}) \\ Gen_R K_n(v^n, \Delta t) &= (A_1 q_1 + A_2 q_2 + \cdots + A_n q_n). \end{aligned}$$

Find the values for the β's by considering the same expression for an autonomous system $\dot{\tilde{u}} = \tilde{f}(\tilde{u})$.

3. Error Propagation

So far we have computed the truncation error or residual of the algorithm. We now relate it to the actual error and its propagation along the calculation. We shall ignore rounding errors and asume a machine of infinite precision. It is important to see how the error propagates for this implies that one can check convergence easily and even improve on the solution. We assume we are given a method with truncation error of order p, and define the error as:

$$e^n := \frac{(u^n - v^n)}{\Delta t^p}, \qquad \text{that is,} \qquad u^n = v^n + \Delta t^p e^n, \tag{5}$$

where $u^n := u(t_n)$ is the exact solution, and $v^0 = u(t_0)$ (same initial data).

From the truncation error order assumption we have,

$$u^{n+1} - u^n - \Delta t A_p(u^n, \Delta t) = R(u(t_n)\Delta t^{p+1} + \mathcal{O}(\Delta t^{p+2}).$$

Where from the expansion we know the remainder term is a function only of $u(t)$.

On the other hand the algorithm gives,

$$v^{n+1} - v^n - \Delta t A_p(v^n, \Delta t) = 0.$$

Thus, substracting, we have,

$$e^{n+1} = e^n + \Delta t[\frac{(A_p(u^n, \Delta t) - A_p(v^n, \Delta t)}{\Delta t^p} + R(u(t_n))] + \mathcal{O}(\Delta t^2) \tag{6}$$

Since $u^n = v^n + \Delta t^p e^n$ we can use Taylor to get,

$$\begin{aligned} e^{n+1} &= e^n + \Delta t \frac{(\partial_u A_p(u^n,\Delta t) e^n \Delta t^p)}{\Delta t^p} + \Delta t R(u(t_n)) + \mathcal{O}(\Delta t^2) \\ &= e^n + \Delta t[\partial_u A_p(u^n,\Delta t)e^n + R(u(t_n))] + \mathcal{O}(\Delta t^2) \end{aligned}$$

Since $A_p(u^n,\Delta t)$ depends smoothly on Δt, to that order we can approximate it by $A_p(u^n,0)$. Thus, we see that the error satisfies,

$$e^{n+1} = e^n + \Delta t[\partial_u A_p(u(t_n),0)e^n + R(u(t_n))], \tag{7}$$

And we can interpret the sequence e^n as a first order (Euler) approximation to the equation:

$$\dot{\phi}_p = \partial_u A_p(u(t),0)\phi_p + R(u(t)), \tag{8}$$

Where the last term is just a source term. Notice that this equation does not depends on Δt! Furthermore, $\partial_u A_p(u(t),0) = \partial_u f(u(t))$.

Therefore we have the following result:

Assertion: *Given $f(u)$ smooth, and u_0, there exists a time interval $[0,T]$ for which the solution exists and the approximation to it, given by a scheme of order p has the form,*

$$u(t_n) = v^n + \Delta t^p \phi_p(t) + \mathcal{O}(\Delta t^{p+1}). \tag{9}$$

With $\phi_p(t)$ a solution to **??**. Notice that initially the initial data for the error is cero, but the higher orders act as sources and it kicks up.

For example, if the take $f(u) = \lambda u$, then the equation for the ϕ_1 is the same as the original equation, so if $Real(\lambda)$ is positive the error, as well as the solution would grow exponentially. So the error can be substantial, even for simple cases.

Exercise: Find the error propagation equation for ϕ_1 for the case of Euler's equation.

4. Error Control

Assume now that we know some solution $u(t)$, $t \in [0,T]$ and we approximate is with our favorite one step numerical scheme, A_p, and get a sequence $\{v^n\}$, $n \in [1,N]$, that is, using a time step $\Delta t = T/N$. How can we trust that we are in the convergence regime of the method? That is how do we know that Δt is small enough so that we can ignore the subsequent terms in the Taylor series?

To check that we use once more the scheme, but this time with half the time step, and get a sequence $\{v^n_{\frac{1}{2}}\}$, $n \in [1,2N]$. So, using our previous result we get:

$$u(t_n) = v^{2n} + (\Delta t)^p \phi_p(t_n) + \mathcal{O}(\Delta t^{p+1}) \quad \text{and}$$

and

$$u(t_n) = v^{2n}_{\frac{1}{2}} + (\frac{\Delta t}{2})^p \phi_p(t_n) + \mathcal{O}(\Delta t^{p+1}) \quad \text{and}$$

Thus,

$$Q := \frac{u(t_n) - v^n}{u(t_n) - v^{2n}_{\frac{1}{2}}} = \frac{\Delta t^p \phi_p(t_n)}{(\frac{\Delta t}{2})^p \phi_p(t_n)} = 2^p.$$

Getting this value in the numerical computation means that we are in the range of convergence of the method (that is, all other terms we have taken as small are really small). Some times people plots directly e^n and $2^p e^{2n}_{\frac{1}{2}}$, since in theory both should give $\phi_1(t_n)$, the plots should roughly coincide.

What happens if we do not have an exact solution at hand. Well, in that case we notice that the difference,

$$v^n - v^{2n}_{\frac{1}{2}} = \Delta t^p \phi_p(t_n) - (\frac{\Delta t}{2})^p \phi_p(t_n) = \Delta t^p (1 - \frac{1}{2^p}) \phi_1(t_n)$$

therefore, if we also compute $\{v^n_{\frac{1}{4}}\}$, that is, the approximation with $\Delta t = \frac{\Delta t}{4}$, we get,

$$Q = \frac{v^n - v^{2n}_{\frac{1}{2}}}{v^{2n}_{\frac{1}{2}} - v^{4n}_{\frac{1}{4}}} = 2^p.$$

Thus we can use also this calculation to assert convergence. This is the usual one in practice.

One can compute this ratio for any component of u, or one can use a norm, and compute the norm ratios.

The difference of two approximations with different time steps, not necessarily one being half the other, allows us to infer the local error, thus, many numerical schemes proceed at each time step taking two steps, usually Δt and $2\Delta t$. Then compare the values, thus, finding the error. If that is bigger than prescribed, $\Delta t \to \Delta t/2$ and the step is taken again. This procedure is repeated until satisfaction. But since using these two different time-steps we have control over the error, we can combine them to get a better approximation, as the following example shows.

Exercise: Consider the combination: $\tilde{v}^n := \frac{1}{2^p - 1}(2^p v^n_{\frac{1}{2}} - v^n_1)$ Use our result on the form of the error to conclude that this approximation is one order higher. Using this information is called *extrapolation*.

5. Stability Region

For applications on the method of lines, it is impractical to take $dt \to 0$ at a rate faster than $dx \to 0$, thus we end up solving a system with a linear part which does not behave as usual, in the sense that is very stiff. If we consider the method of lines to approximate a wave equation, after diagonalizing the space finite difference operators, we can see that we are solving a problem like the following,

$$v^{n+1} = v^n + \Delta t[\lambda v^n + F(v^n)],$$

with the particularity that $\mu := \Delta t \lambda \approx \frac{\Delta t}{\Delta x}$ stays at a fixed, finite value as $\Delta t \to 0$. The second term is bounded, so it does not pose any problem when $\Delta t \to 0$. We can concentrate in the first term. In particular, for hyperbolic equations we expect μ to be purely imaginary.

For Euler's method we get,

$$v^{n+1} = v^n + \mu v^n = (1 + \mu) v^n,$$

Which has a solution given by,

$$v^n = (1 + \mu)^n v^0$$

Therefore if $|1 + \mu| > 1$ the iteration would result in an unbounded sequence and so in an unstable method. Writing it as $|-1 - \mu| \leq 1$ we see that the stability region is a disk centered at -1 of radius 1. In particular the whole imaginary axis, where we expect μ would lie for hyperbolic problems, is outside it. But notice also that for $Real(\mu) < -2$ the scheme is also unstable, so, even for parabolic problems one runs into instabilities.

Exercise: Find the stability region for the Heun scheme.

Exercise: Find the stability region for the RK4 scheme and plot it.

Most low orders systems are unstable at the imaginary axis, one needs to use at least three function evaluations to find methods containing a segment of the imaginary axis.

For the case of implicit methods the situation is better, for instance, the Implicit Euler method, given by,

$$v^{n+1} = v^n + \mu v^{n+1} = \frac{v^n}{1-\mu}$$

Leads to a solution given by

$$v^n = \frac{v^0}{(1-\mu)^n},$$

so giving a stability region for $|1-\mu| \geq 1$, that is, the outside of the disk of radius 1 centered at 1. In particular this method contains the whole half space $Real(\mu) \leq 0$, so it is appropriate for both, hyperbolic and parabolic problems for which much larger time intervals can be takes. The problem is that in general this methods requieres to invert some matrices.

Exercise: Find the stability region of the semi-implicit family of schemes given by:

$$v^{n+1} = v^n + \mu[\theta v^{n+1} + (1-\theta)v^n]$$