ODE: INTRO NUMERICS

OSCAR REULA

1. THE PROBLEM AND THE BASIC APPROXIMATION OF EULER

We are trying to approximate the solutions to the equation,

$$\frac{du}{dt} = f(u) \tag{1}$$

For the moment we do not introduce any dependence of $f(\cdot)$ with time, this addition is rather trivial, and carrying it along all the calculations, cumbersome. We shall assume $f(\cdot)$ is several times continuously differentiable, as needed for the numerical method in question.

We know from theory that given any value u_0 in the domain of f there will be a finite interval $[0, T]$ where the solution exists $u(t)$ with $u(0) = u_0$ exists. We want to find arbitrarily good approximations to $u(t)$. That is, we want to find a norm (or more generally a topology) and a sequence of approximations, $\{\tilde{v}_i(t)\}$ such that $||u - \tilde{v}_i|| \to 0$ as $i \to \infty$.

We are not going to work with the functions $\tilde{v}_i$, but rather with sequence of values v_i^n such that then $\tilde{v}_i$ is an interpolated function satisfying $\tilde{v}_i(t_n) = v^n$, for some set of points $\{t_n\} \in [0, T], n = 1, \ldots, N$.

For simplicity also we are going to assume the set $\{t_n\} \in [0, T]$ consist of equi-spaced points,

$$t_n = (n - 1) * \Delta t \qquad \Delta t = T/(N - 1)$$

The first approximation is to assume, that knowing the value of v^n we can approximate the value of v^{n+1} applying Taylor's approximation to first order. That is, from

$$u(t_{n+1}) = u(t_n + \Delta t) \approx u(t_n) + \frac{du}{dt}|_{t=t_n}\Delta t = u(t_n) + f(u(t_n))\Delta t$$

consider the approximation,

$$v^{n+1} = v^n + f(v^n)\Delta t.$$

Given $v^1 = u_0$ this gives a unique finite sequence of points $\{v^n\}, n = 1, \ldots, N$. Out of this finite sequence we build the interpolating approximation $\tilde{v}_N$. This approximation method is called **Euler's method**.

1.1. **One-Step error.** The one-step error is defined as the error we make at one step of the approximation, assuming the error on the previous step vanishes. This is obviously a false assumption, but latter we shall lift it.

$$
\begin{aligned}
e^n &= u(t_n) - v^n = u(t_{n-1}) + \frac{du}{dt}|_{t=t_{n-1}}\Delta t + \frac{d^2u}{dt^2}|_{t=\zeta}\frac{\Delta t^2}{2} - v^{n-1} - f(v^{n-1})\Delta t \\
&= e^{n-1} + [f(u(t_{n-1})) - f(v^{n-1})]\Delta t + \frac{d^2u}{dt^2}|_{t=\zeta}\frac{\Delta t^2}{2}
\end{aligned}
\tag{2}
$$

Where we have used the mean value theorem for the second derivative. Thus, assuming for now that $e^{n-1} = 0$, we get,

$$e^n = u(t_n) - v^n \leq R\Delta t^2$$

Where

$$R = \max_{t \in [0,T]} \|\frac{1}{2}\frac{d^2u}{dt^2}|_t\|$$

Since we are going to make at least an error of order Δt^2 at every time step, and we are making $N = 1 + \frac{T}{\Delta T}$ steps, the **global error** is order Δt. Thus we say Euler's method is first order.

Example:

We apply the method to the equation

$$\frac{du}{dt} = \lambda u$$

The algorithm for obtaining the approximation is:

$$v^{n+1} = v^n + \lambda v^n \Delta t = (1 + \lambda \Delta t)v^n$$

Whose solution is:

$$v^n = (1 + \lambda \Delta t)^{(n-1)}v^1$$

with a local error given by $\frac{|\lambda\Delta t|^2}{2} \max_{t \in [0,T]} \|e^{\lambda t}\|$. We see in the figure 1 that even when $\lambda = 1$, so that $|e^{\lambda t}| = 1$ the error is very big. Indeed, for $\lambda = i$ we have that $|1 + \lambda \Delta t| = \sqrt{1 + \Delta t^2} > 1$. So the there is an amplification of the solution for any $\Delta t > 0$.
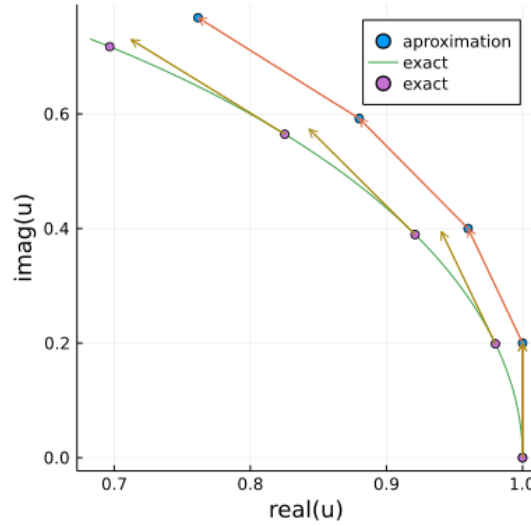

FIGURE 1. Euler steps for $\dot{x} = ix$

2. HIGHER ORDER METHODS

To derive higher order methods, that is, methods whose local error is order Δt^p with $p > 1$, the important fact to understand is that we are approximating the solution at fixed steps. Thus, given v^n we need a good approximation for v^{n+1}.

2.1. **Taylor expansions.** We could do that, for instance, using Taylor's expansion,

$$
(3) \qquad
\begin{aligned}
u(t_{n+1}) &= u(t_n + \Delta t) \\
&= u(t_n) + \frac{du}{dt}\Big|_{t=t_n}\Delta t + \frac{d^2u}{dt^2}\Big|_{t=t_n}\frac{\Delta t^2}{2} + \frac{d^3u}{dt^3}\Big|_{t=\zeta}\frac{\Delta t^3}{3!} \\
&= u(t_n) + f(u(t_n))\Delta t + \frac{df(u)}{dt}\Big|_{t=t_n}\frac{\Delta t^2}{2} + \frac{d^2f(u)}{dt^2}\Big|_{t=\zeta}\frac{\Delta t^3}{3!}
\end{aligned}
$$

Using now that $\frac{df(u)}{dt} = \frac{df}{du}\frac{du}{dt} = \frac{df}{du}f$ and $\frac{d^2f(u)}{dt^2} = \frac{d^2f}{du^2}f^2 + \frac{df}{du}^2 f$, we get,

$$
u(t_{n+1}) = u(t_n) + f(u(t_n))\Delta t + \frac{df}{du}(u)f(u)\big|_{t=t_n}\frac{\Delta t^2}{2} + [\frac{d^2f}{du^2}(u)f^2(u) + \frac{df}{du}^2(u)f(u)]\big|_{t=\zeta}\frac{\Delta t^3}{3!}
$$

Which suggests we could approximate $u(t_{n+1})$ by,

$$
v^{n+1} = v^n + f(v^n)\Delta t + \frac{df}{du}(v^n)f(v^n)\frac{\Delta t^2}{2}
$$

with a local error given by,

$$
max_{t\in[0,T]}\|[\frac{d^2f}{du^2}(u)f^2(u) + \frac{df}{du}^2(u)f(u)]|_t\|\frac{\Delta t^3}{3!}
$$

This Taylor methods can be good as long as we can compute easily and efficiently higher derivatives of a given f. With algebraic or automatic derivation packages this is feasible for many lot of cases. Of course one can continue and take higher and higher derivatives in the Taylor expansion and so obtain methods with higher and higher orders.

Example: Taking the equation of the previous example we get,

$$
\frac{du}{dt} = \lambda u
$$

The algorithm for obtaining the approximation is:

$$
v^{n+1} = v^n + \lambda v^n \Delta t + \lambda^2 v^n \frac{\Delta t^2}{2} = (1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2)v^n
$$

Whose solution is:

$$
v^n = (1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2)^{(n-1)}v^1
$$

Notice that for $\lambda = i$ in this case we find a milder amplification $|1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2| = \sqrt{1 + \frac{\Delta t^4}{4}}$.

2.2. **Multi-step Methods.** From Taylor's expansion,

$$u(t_{n+1}) = u(t_n + \Delta t) = u(t_n) + \frac{du}{dt}|_{t=t_n}\Delta t + \frac{d^2u}{dt^2}|_{t=t_n}\frac{\Delta t^2}{2} + \frac{d^3u}{dt^3}|_{t=t_n}\frac{\Delta t^3}{3!} + \frac{d^4u}{dt^4}|_{t=\zeta}\frac{\Delta t^4}{4!}$$

and taking $\Delta t \to -\Delta t$ we see that,

$$u(t_n + \Delta t) - u(t_n - \Delta t) = 2\frac{du}{dt}|_{t=t_n}\Delta t + \frac{d^3u}{dt^3}|_{t=\zeta}\frac{\Delta t^3}{3!}$$

Thus, the algorithm,

$$v^{n+1} = v^{n-1} + 2\Delta t f(v^n)$$

has also a third order local error! This scheme has the advantage that it is second order but only uses a single evaluation of f. So it is computationally very economical. On the other hand, one has to store 2 previous values, namely v^n and v^{n-1}. The problem with these methods is that it might lead to spurious solutions.

Example: Taking again the equation of the first example,

$$\frac{du}{dt} = \lambda u.$$

The algorithm for obtaining the approximation is:

$$v^{n+1} = v^{n-1} + 2\mu v^n \qquad \mu := \lambda \Delta t$$

Whose solution is:

$$v^n = A^+ S_+^n + A^- S_-^n$$

Where the constants (A_+, A_-) are computed out of the initial values, (v^1, v^2). But notice that $S_+ S_- = 1$. Thus, except in the special cases where $S_\pm = e^{\pm i\theta}$ we have always a growing mode.

2.3. **One Step Methods.** In the multiple-steps method we just discussed above we saw that we got a higher precision by using an intermediate point, the nth, as the point where to compute the derivative, but we got an unstable system for we needed to take two points for computing the next, and that type of scheme has always two possible solutions. This leads to instabilities. But instead of taking two previous steps, we could take a first step, and go only half the way, that is multiply for $\frac{\Delta t}{2}$, to a point $v^* = v^n + \frac{\Delta t}{2}f(v^n)$. At this midpoint we could compute $f(v^*)$ to advance a whole interval. In short, the scheme is:

$$
\begin{aligned}
k_1 &= \Delta t f(v^n) \\
k_2 &= \Delta t f(v^n + \frac{1}{2}k_1) \\
v^{n+1} &= v^n + k_2
\end{aligned}
$$

This scheme is called **Midpoint** or **improved Euler**. In the next chapter we shall study the error of this method, here we just shall look at our example, which, apart from indicating that the method has a third order local error, it behaves much better for orbits.

Example: Taking again the equation of the first example,

$$\frac{du}{dt} = \lambda u.$$

The algorithm for obtaining the approximation is:

$$k_1 = \mu v^n$$

$$v^{n+1} = v^n + \mu(v^n + \frac{1}{2}k_1) = (1 + \mu + \frac{\mu^2}{2})v^n$$

Which is identical (for this particular equation) to the algorithm we got from taking an extra term in Taylor's expansion.
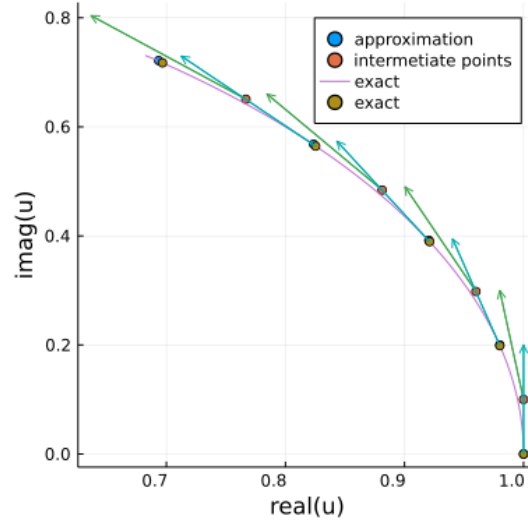

FIGURE 2. Mid-Point steps for $\dot{x} = ix$

Alternatively we could have evolved a whole step Δt, to a point $v^* = v^n + \Delta t f(v^n)$. At this approximate point we could compute $f(v^*)$, and then advance from v^n taking the average value of f, a whole interval. In this case the scheme is:

$$k_1 = \Delta t f(v^n)$$

$$k_2 = \Delta t f(v^n + k_1)$$

$$v^{n+1} = v^n + \frac{1}{2}(k_1 + k_2)$$

And it is called **Heum method**.

Actually this are not the only second order methods (third order local error). There is a whole family of them. To find out all of them we start with a generic method having just two evaluations of the function f.

$$k_1 = \Delta t f(v^n)$$

$$k_2 = \Delta t f(v^n + \alpha_{21}k_1)$$

$$v^{n+1} = v^n + (A_1 k_1 + A_2 k_2)$$

The Midpoint method corresponds to $\alpha_{21} = \frac{1}{2}$, $(A_1 = 0, A_2 = 1)$, Heum's to $\alpha = 1$, $(A_1 = \frac{1}{2}, A_2 = \frac{1}{2})$.

As before we have,

$$u(t_{n+1}) = u(t_n) + f(u(t_n))\Delta t + \frac{df}{du}(u)f(u)|_{t=t_n}\frac{\Delta t^2}{2} + [\frac{d^2 f}{du^2}(u)f^2(u) + \frac{df}{du}^2(u)f(u)]|_{t=\zeta}\frac{\Delta t^3}{3!},$$

and

$$\begin{aligned}
v^{n+1} &= v^n + (A_1 k_1 + A_2 k_2) \\
&= v^n + A_1\Delta t f(v^n) + A_2\Delta t f(v^n + \alpha_{21}k_1) \\
&= v^n + \Delta t(A_1 f(v^n) + A_2[f(v^n) + \frac{df}{du}(v^n)\alpha_{21}\Delta t f(v^n)]) + O(\Delta t^3) \\
&= v^n + \Delta t(A_1 + A_2)f(v^n) + A_2\alpha_{21}\frac{df}{du}(v^n)f(v^n)\Delta t^2 + O(\Delta t^3)
\end{aligned}$$

So a comparison among the last two equations implies that to have a local error $O(\Delta t^3)$ we need that

$$A_1 + A_2 = 1 \quad A_2\alpha_{21} = \frac{1}{2}.$$

Thus, we see that there are a one-parameter family of second order methods.