Lecture 7: Numerical solution of ODEs I

The model ODE

In this lecture we will be learning how to solve the first order ODE

$$\frac{dy}{dt} = f(t, y)\,. \tag{1}$$

The reason we analyze such a simplified equation is because, as we saw in the previous lecture, all higher ODEs can be written in the form of a system of first order ODEs, which we write as

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}(t, \mathbf{y})\,. \tag{2}$$

These higher order systems can be solved using the same methods we develop to solve the model ODE (1).

Forward and backward Euler: explicit vs. implicit methods

Discretization

The model ODE (1) is written discretely by choosing a time step (or space step) at which we would like to evaluate both sides of the equation. Let's say we want to evaluate both sides of (1) at time step n. In this case, the model ODE would be written as

$$\left.\frac{dy}{dt}\right|_n = f_n\,, \tag{3}$$

where $f_n = f(t_n, y_n)$. So far the discretization is exact. We have not made any approximations yet because we are assuming that we can evaluate everything exactly. If we approximate the left hand side with the forward discrete derivative with

$$\left.\frac{dy}{dt}\right|_n = \frac{y_{n+1} - y_n}{\Delta t} + \mathcal{O}\left(\Delta t\right)\,, \tag{4}$$

then we have the first order accurate approximate to the model equation (1) as

$$\frac{y_{n+1} - y_n}{\Delta t} = f_n + \mathcal{O}\left(\Delta t\right)\,, \tag{5}$$

or

$$y_{n+1} = y_n + \Delta t f_n + \mathcal{O}\left(\Delta t^2\right)\,. \tag{6}$$

This equation is known as the **forward Euler** method because it uses the forward discrete derivative in time to evaluate the left hand side. Since in order to evaluate y_{n+1}, we use information from time step n, this is known as an **explicit** method.

If we choose to write the model equation (1) at time step $n+1$

$$\left.\frac{dy}{dt}\right|_{n+1} = f_{n+1}\,, \tag{7}$$

then this can be approximated using the backward discrete derivative to yield

$$\frac{y_{n+1}-y_n}{\Delta t} = f_{n+1} + \mathcal{O}\left(\Delta t\right)\,, \tag{8}$$

or

$$y_{n+1} = y_n + \Delta t f_{n+1} + \mathcal{O}\left(\Delta t^2\right)\,. \tag{9}$$

This is known as the **backward Euler** method because is uses the backward finite difference to evaluate the first derivative. If you were to evaluate y at time step $n+1$ you would see that you need information at time step $n+1$ in order to compute f_{n+1}. When you need information at the next time step, the method is known as an **implicit** method.

An example

Let's say you want to numerically determine the evolution of the ODE

$$\frac{dy}{dt} = y\cos y\,, \tag{10}$$

with $y(0)=1$. If we use the forward Euler method, we have

$$\begin{aligned} y_{n+1} &= y_n + \Delta t y_n \cos y_n\,, \\ y_{n+1} &= y_n(1+\Delta t\cos y_n)\,. \end{aligned} \tag{11}$$

We can easily obtain y_1 if y_0 is known because everything on the right hand side is known explicitly. If we use the backward Euler method, however, we have

$$\begin{aligned} y_{n+1} &= y_n + \Delta t y_{n+1}\cos y_{n+1}\,, \\ y_{n+1}\left(1-\Delta t\cos y_{n+1}\right) &= y_n\,. \end{aligned} \tag{12}$$

Now, instead of having the solution of y_1 in terms of y_0, we have a horrendous nonlinear equation for y_1 that must be solved using a nonlinear equation solver, such as Newton's method. Clearly, then, in this case, the explicit method is much faster than the implicit method because we do not have to iterate at every time step to find the solution. The next section shows the advantages of using implicit methods.

The linearized ODE

In the preceding example we saw how the forward Euler method was much easier and faster to use than the backward Euler method. Any time something seems too good to be true in numerical methods, it really is too good to be true. Which leads us to the first law of numerical methods: **There is no free lunch!**. The problem with the forward Euler method,

despite its simplicity, is that it can be unstable, while the implicit backward Euler method is unconditionally stable.

In order to study the stability of numerical methods for ODEs, we first need a model equation that we can use to apply each method to and analyze its stability properties. This model equation is the linear ODE

$$\frac{dy}{dt} = -\lambda y\,, \tag{13}$$

where λ is some characteristic value of the ODE that arises from assuming that the ODE behaves in this linear manner. We need to do this because we would like to analyze the linear stability characteristics of numerical methods applied to all ODEs in general. Take for example, the ODE used in the previous example,

$$\frac{dy}{dt} = y\cos y\,. \tag{14}$$

In order to analyze the stability properties of this nonlinear ODE, we need to linearize it. When we linearize an ODE, we analyze its behavior in the vicinity of some point t_0, y_0 to determine its stability properties. To analyze the behavior of an ODE in the vicinity of y_0 and t_0, we make the subsitution $y = y_0 + y'$ and $t = t_0 + t'$, and assume that $y' = y - y_0$ and $t' = t - t_0$ represent very small quantities. Substituting these values into equation (14), we have

$$\frac{dy}{dt} = \frac{dt'}{dt}\frac{d(y_0 + y')}{dt'} = (y_0 + y')\cos(y_0 + y')\,. \tag{15}$$

In order to linearize this, we need to use the Taylor Series approximation of the cosine function

$$\cos(y_0 + y') = \cos(y_0) - y'\sin(y_0) + \mathcal{O}\left((y')^2\right)\,. \tag{16}$$

Substitution into equation (15) yields

$$\frac{dy'}{dt'} + (\cos y_0 - y_0 \sin y_0)\, y' = y_0 \cos y_0 + \mathcal{O}\left((y')^2\right)\,. \tag{17}$$

If we assume that y' is very small, then the second order term is negligible, and we have

$$\frac{dy'}{dt'} + (\cos y_0 - y_0 \sin y_0)\, y' = y_0 \cos y_0\,, \tag{18}$$

which is a linear inhomogeneous ODE in terms of y' and t' that represents the behavior of the original nonlinear ODE in equation (14) in the vicinity of y_0, t_0. If we substitute back in the values for $y' = y - y_0$ and $t' = t - t_0$ we have

$$\frac{dy}{dt} + (\cos y_0 - y_0 \sin y_0)\, y = 2y_0 \cos y_0 - y_0^2 \sin y_0\,. \tag{19}$$

If we split the linearized solution into its homogeneous and particular parts with $y = y_h + y_p$, then the homogenous solution satisfies

$$\frac{dy_h}{dt} = -\lambda y_h\,, \tag{20}$$

where $\lambda = (\cos y_0 - y_0 \sin y_0)$. If we analyze the stability properties of this linearized ODE, then we can apply that analysis to the nonlinear problem by seeing if it remains stable at all values of t_0 and y_0.

Stability

If we apply the forward Euler method to the model linearized ODE

$$\frac{dy}{dt} = -\lambda y \,, \tag{21}$$

then we have

$$\begin{aligned} y_{n+1} &= y_n - h\lambda y_n \,, \\ &= y_n \left(1 - h\lambda\right) \,, \end{aligned} \tag{22}$$

where $h = \Delta t$. If we write the amplification factor at each time step as

$$G_n = \left| \frac{y_{n+1}}{y_n} \right| \,, \tag{23}$$

then, for the forward Euler method, we have

$$G_n = \left| 1 - h\lambda \right| \,, \tag{24}$$

where the vertical bars imply the modulus, to account for the possibility that λ may not necessarily be real. If the amplification is less than 1, then we are guaranteed that the solution will not grow without bound, and hence it will be stable. If we assume that λ is real, then for stability we must have

$$-1 < 1 - h\lambda < +1 \,, \tag{25}$$

which implies that, for stability, $0 < \lambda h < 2$, if λ is real. This translates to a time step restriction for stability, for which $0 < \Delta t < 2/\lambda$.

Now consider the backward Euler method applied to the model linearized ODE. This yields

$$\begin{aligned} y_{n+1} &= y_n - h\lambda y_{n+1} \,, \\ \left(1 + h\lambda\right) y_{n+1} &= y_n \,, \end{aligned} \tag{26}$$

and the amplification factor is given by

$$G_n = \left| \frac{1}{1 + h\lambda} \right| \,. \tag{27}$$

If λ is real, then we must have $\lambda h > 0$, or $\Delta t > 0$. The backward Euler method is hence stable in the linear sense for all Δt! While it may be more expensive to use the implicit method, as in the example discretization of equation (14), it is guaranteed to be stable.

The greatest drawback to the Euler methods is that they are first order accurate. In the next sections, we derive more accurate methods to solve ODEs.

Euler predictor-corrector method

The improved Euler method is derived by integrating the model ODE from t_n to t_{n+1} to obtain

$$\int_{t_n}^{t_{n+1}} \frac{dy}{dt}\, dt = \int_{t_n}^{t_{n+1}} f(y)\, dt\,. \tag{28}$$

Using the trapezoidal rule, we can approximate the above integral to third order accuracy with

$$y_{n+1} - y_n = \frac{\Delta t}{2}\left(f_n + f_{n+1}\right) + \mathcal{O}\left(\Delta t^3\right)\,. \tag{29}$$

to obtain the second order accurate approximation to the model ODE as

$$\frac{y_{n+1} - y_n}{\Delta t} = \frac{1}{2}\left(f_n + f_{n+1}\right) + \mathcal{O}\left(\Delta t^2\right)\,. \tag{30}$$

As it is, this method is an implicit method because we need information at time step $n+1$ in order to evaluate the right hand side. Instead of using f_{n+1}, we will use a predicted value, $f_* = f(y_*)$, where y_* is obtained with the forward Euler predictor step

$$y_* = y_n + \Delta t f_n\,. \tag{31}$$

The Euler predictor-corrector method is then given in two steps:

$$\begin{aligned} \text{Predictor:} \quad y_* &= y_n + \Delta t f_n + \mathcal{O}\left(\Delta t^2\right)\,, \\ \text{Corrector:} \quad y_{n+1} &= y_n + \frac{\Delta t}{2}\left(f_n + f_*\right) + \mathcal{O}\left(\Delta t^3\right)\,. \end{aligned} \tag{32}$$

This method is second order accurate, since y_* approximates y_{n+1} to second order accuracy. Substituting $y_* = y_{n+1} + \mathcal{O}\left(\Delta t^2\right)$ into f_* yields

$$\begin{aligned} f_* &= f(y_*)\,, \\ &= f(y_{n+1} + \mathcal{O}\left(\Delta t^2\right))\,, \\ &= f(y_{n+1}) + \mathcal{O}\left(\Delta t^3\right)\,. \end{aligned}$$

Substituting this result into the corrector yields

$$\frac{y_{n+1} - y_n}{\Delta t} = \frac{1}{2}\left(f_n + f_{n+1}\right) + \mathcal{O}\left(\Delta t^2\right)\,, \tag{33}$$

which is identical in accuracy to equation (30).

Runge-Kutta methods

The Runge-Kutta methods are the most popular methods of solving ODEs numerically. They can be derived for any order of accuracy, but we will derive the second order method first. The second order Runge-Kutta method is derived by taking two steps to get from n to $n+1$ with

$$\begin{aligned} y_{n+1} &= y_n + a k_1 + b k_2\,, \\ k_1 &= h f(t_n, y_n)\,, \\ k_2 &= h f(t_n + \alpha h, y_n + \beta k_1)\,, \end{aligned} \tag{34}$$

where $h = \Delta t$ is the time step. In order to determine what the constants a, b, α, and β are, we must use the Taylor series to match the terms and make the method second order accurate. By substituting in for k_1 and k_2, we have

$$y_{n+1} = y_n + ahf(t_n, y_n) + bhf\left[t_n + \alpha h, y_n + \beta hf(t_n, y_n)\right] . \tag{35}$$

In order to expand the third term in equation (35), we need to use the Taylor series expansion of a function of more than one variable, which is given by

$$f(t + \Delta t, y + \Delta y) = f(t, y) + \Delta t \frac{\partial f}{\partial t} + \Delta y \frac{\partial f}{\partial y} + \mathcal{O}\left(\Delta t \Delta y\right) , \tag{36}$$

which, when applied to the third term in equation (35), results in

$$f\left[t_n + \alpha h, y_n + \beta hf(t_n, y_n)\right] = f + \alpha h \frac{\partial f}{\partial t} + \beta hf \frac{\partial f}{\partial y} , \tag{37}$$

where all functions and derivatives are evaluated at time step n, and we have left off the truncation error. Substituting this into equation (35) results in

$$y_{n+1} = y_n + h(a + b)f + \alpha bh^2 \frac{\partial f}{\partial t} + \beta bh^2 f \frac{\partial f}{\partial y} . \tag{38}$$

Since y is only dependant on the variable t, then the Taylor series expansion about y_{n+1} is given by the ordinary derivatives with

$$y_{n+1} = y_n + h\frac{dy}{dt} + \frac{h^2}{2}\frac{d^2y}{dt^2} + \mathcal{O}\left(h^3\right) . \tag{39}$$

But since the ODE we are trying to solve is given by

$$\frac{dy}{dt} = f , \tag{40}$$

then we know that

$$\frac{d^2y}{dt^2} = \frac{df}{dt} , \tag{41}$$

so equation (39) becomes

$$y_{n+1} = y_n + hf + \frac{h^2}{2}\frac{df}{dt} , \tag{42}$$

where we have left off the truncation error. Since from the chain rule, if f is a function of t and y, then

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial y} dy , \tag{43}$$

then

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f . \tag{44}$$

Substitution into equation (42) yields

$$y_{n+1} = y_n + hf + \frac{h^2}{2}\frac{\partial f}{\partial t} + \frac{h^2}{2} f \frac{\partial f}{\partial y} . \tag{45}$$

Comparing this equation to equation (38),

$$\begin{aligned} y_{n+1} &= y_n + hf + \frac{h^2}{2}\frac{\partial f}{\partial t} + \frac{h^2}{2} f \frac{\partial f}{\partial y}\,, \\ y_{n+1} &= y_n + h(a+b)f + \alpha b h^2 \frac{\partial f}{\partial t} + \beta b h^2 f \frac{\partial f}{\partial y}\,, \end{aligned} \tag{46}$$

in order for the terms to match, we must have

$$\begin{aligned} a + b &= 1\,, \\ \alpha b &= \frac{1}{2}\,, \\ \beta b &= \frac{1}{2}\,. \end{aligned} \tag{47}$$

This is a system of three equations in four unknowns. Therefore, we are free to choose one independantly and the others will then be determined, and the method will still be a second order method. If we let $a = 1/2$, then the other parameters must be $b = 1/2$, $\alpha = 1$, and $\beta = 1$, so that the second order Runge-Kutta method is given by

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2\,, \\ k_1 &= hf(t_n, y_n)\,, \\ k_2 &= hf(t_n + h, y_n + hf(t_n, y_n))\,, \end{aligned} \tag{48}$$

which is just the Euler predictor-corrector scheme, since $k_2 = hf_*$, and substitution results in

$$y_{n+1} = y_n + \frac{h}{2}(f_n + f_*)\,. \tag{49}$$

Higher order Runge-Kutta methods can be derived using the same technique. The most popular method is the fourth order Runge-Kutta method, or RK4 method, which is given by

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)\,, \\ k_1 &= hf(t_n, y_n)\,, \\ k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{1}{2}k_1\right)\,, \\ k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{1}{2}k_2\right)\,, \\ k_4 &= hf\left(t_n + h, y_n + k_3\right)\,. \end{aligned}$$

Although this method is a fourth order accurate approximation to the model ODE, it requires four function evaluations at each time step. Again, there is never any free lunch!