1 - Ordinary Differential Equations

Scribed by Tinghao Li, Luca Venturi

1 Introduction

An Ordinary Differential Equation (ODE) is an equation of the form

\[
\dot{y}(t) = f(t, y(t)) \\
y(0) = y_0
\]

where \( f : [-T, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a continuous function and \( y_0 \in \mathbb{R}^n \) an initial value. We call solution a continuous map \( t \mapsto y(t) \) defined in a neighborhood of \( t = 0 \) satisfying equations (1).

A standard result tells us that if \( f \) is uniformly Lipschitz (i.e. \( \|f(t, y) - f(t, y)\| \leq L\|y - x\| \)) in a neighborhood of \( (0, y_0) \), then a solution exists and it’s unique (cfr. [1], Theorem 12.1). This results can be proven by constructing a solution in an iterative way. Equation (1) is equivalent to its integral form

\[
y(t) = y(0) + \int_0^t f(s, y(s)) \, ds
\]

If we define the functions \( y_0 \equiv y_0 \) and

\[
y_{N+1}(t) = y_0 + \int_0^t f(s, y_N(s)) \, ds
\]

for \( N \geq 0 \), then one can use the Lipschitz property of \( f \) to show that \( y_{N+1} \) converge uniformly on a neighborhood of \( t = 0 \) to a continuous function. In particular, calling \( y(t) = \lim_{N \to \infty} y_N(t) \), we get that

\[
y(t) = \lim_{N \to \infty} y_N(t) \\
= y_0 + \lim_{N \to \infty} \int_0^t f(s, y_N(s)) \, ds \\
= y_0 + \int_0^t f(s, \lim_{N \to \infty} y_N(s)) \, ds \\
= y_0 + \int_0^t f(s, y(s)) \, ds
\]

This means that \( y(t) \) is thus the solution to our ODE. Unfortunately, this is not a viable method to evaluate the solution numerically.
2 One-step methods

A simple way to approximate the solution to \( \dot{y}(t) \) numerically comes from numerically approximating its derivative. By definition

\[
\dot{y}(t) = \lim_{h \to 0} \frac{y(t+h) - y(t)}{h}
\]

Therefore, if we choose \( h \) sufficiently small, we have that

\[
\dot{y}(t) \approx \frac{y(t+h) - y(t)}{h}
\]  

(2)

Plugging this into the equation (1), we get

\[
y(t+h) \approx y(t) + h f(t, y(t))
\]

Now, suppose we want to compute an approximation \( \hat{y}(T) \) of \( y(T) \) at time \( T > 0 \). We can break the interval \([0, T]\) into \( N \) intervals, and find approximations \( \hat{y}(0) = y(0) \) and \( \hat{y}_n = \hat{y}(t_n) \approx y(t_n) \), where \( t_n = nh, h = T/N \). Using the above formula, we can find \( \hat{y}_n \) by the recurrence

\[
\hat{y}_{n+1} = \hat{y}_n + h f(t_n, \hat{y}_n)
\]

Intuitively, we expect this to be a good approximation to the actual solution if \( h \) is small enough (or equivalently, \( N \) is large enough). This method is known as Explicit Euler. Other methods can be derived in the same way, starting from a formula to approximate the derivative of a function.

2.1 A simple example

We can start to understand how well different methods work by looking at a simple example (see also the Colab notebook to implement and play with a similar example). Consider the ODE

\[
\dot{y}(t) = -100 y(t)
\]

\[
y(0) = 1
\]

In this case we know the exact solution: \( y(t) = e^{-100t} \)

2.1.1 Forward Euler’s method

Let’s look at how the solution given by Explicit Euler looks like. We have that

\[
\hat{y}_1 = y_0 - 100h y_0 = (1 - 100h)y_0
\]

\[
\hat{y}_1 = \hat{y}_1 - 100h \hat{y}_1 = (1 - 100h)\hat{y}_1 = (1 - 100h)^2 y_0
\]

Iterating, we see that it holds \( \hat{y}_n = (1 - 100h)^n y_0 \). How good is this as an approximation to the true solution? Call \( K(h) = (1 - 100h) \). If \( K(h) < -1 \) (or equivalently \( h > 0.02 \)) then it easy to see that the solution diverges as \( n \) increases, which is the opposite behavior of \( y(t) \)!

Moreover, if \( K(h) < 0 \) (or equivalently \( h > 0.01 \)) the solution keeps oscillating around 0, while we know that the actual solution is always positive. Only for \( h < 0.01 \) our approximation starts to resemble the behaviour of \( y(t) \).
2.2 Backward Euler’s method

The Implicit Euler’s method iteration is given by

\[ \dot{y}_{n+1} = \dot{y}_n + hf(t_n, \dot{y}_{n+1}) \]

(this formula can be derived from equation (2) by choosing \( h < 0 \)). Re-arranging, for the considered example one finds

\[ \dot{y}_n = (1 + 100h)^{-n} y_0 \]

We can notice that this method does not have the same issues as the Explicit Euler’s method: the approximation behaves as the actual solution even for larger \( h \)’s.

2.3 Trapezoidal method

The trapezoidal method iteration is given by

\[ \dot{y}_{n+1} = \dot{y}_n + \frac{h}{2} \left[ f(t_n, \dot{y}_n) + f(t_n, \dot{y}_{n+1}) \right] \]

Re-arranging, for the considered example one finds

\[ \dot{y}_n = \left( \frac{1 - 50h}{1 + 50h} \right)^n y_0 \]

We see that in this case \( \dot{y}_n \to 0 \) as \( n \to \infty \) for any value for \( h \), but if \( \frac{1-50h}{1+50h} < 0 \) (i.e. \( h > 0.02 \)) the approximation oscillates around 0.

3 Convergence analysis

Which of the different methods works best? To understand this, one needs to quantify the error due to the approximation of the ODE. We consider a generic one-step method, i.e. which can be written as

\[ \dot{y}_{n+1} = \dot{y}_n + h \Phi(t_n, \dot{y}_n, h) \]

for some function \( \Phi \) depending on \( f \).

3.1 Truncation error

The first error we define is the one due to the use of the approximation (2) at one time step. It is called truncation error:

\[ T_n = \frac{y_{n+1} - y_n}{h} - \Phi(t_n, y_n, h) \]

where \( y_n = y(t_n) \) is the actual solution. Let’s try to get an explicit bound for the Euler method. We have

\[ hT_n = y_{n+1} - y_n - hf(t_n, y_n) \]

Subsisting \( y_{n+1} \) with its first order Taylor approximation, we get

\[ y_{n+1} = y_n + h \dot{y}_n + O(h^2) = y_n + h f(t_n, y_n) + O(h^2) \]
which gives
\[ hT_n = O(h^2) \]

Therefore \( T_n = O(h) \) goes to 0 as \( h \to 0 \). In general, a method such that
\[
T(h) = \max_{0 \leq n \leq T/h} |T_n| \to 0
\]
as \( h \to 0 \) is said consistent.

### 3.2 Convergence

The error at time \( t_n \) is given by \( e_n = y_n - \hat{y}_n \). The total error is given by \( e(h) = \max_{0 \leq n \leq T/h} |e_n| \).

We say that the method converges if \( e(h) \to 0 \) as \( h \to 0 \). We can see that if \( \Phi \) is (uniformly) Lipschitz and the method is consistent, then it is also convergent. Indeed, we have that
\[
e_{n+1} = y_n + h \Phi(t_n, y_n, h) + h T_n - \hat{y}_n - h \Phi(t_n, \hat{y}_n, h)
= e_n + h (\Phi(t_n, y_n, h) - \Phi(t_n, \hat{y}_n, h)) + h T_n
\leq e_n + hL(y_n - \hat{y}_n) + +h T_n
\leq (1 + hL)e_n + h T_n
\leq (1 + hL)^2 e_{n-1} + h(1 + hL)T_{n-1} + h T_n
\leq \cdots
\leq (1 + hL)^{n+1} e_0 + h \sum_{i=0}^{n} (1 + hL)^i T_{n-i}
\]

It follows that
\[
e(h) \leq h T(h) \sum_{i=0}^{T/h} (1 + hL)^i
= hT(h) \frac{(1 - (1 + hL)^{T/h})}{1 - 1 - hL}
= T(h) \frac{((1 + hL)^{T/h} - 1)}{L}
\leq T(h) \frac{e^{TL}}{L}
\]

Therefore, convergence holds if the method is consistent. Moreover, the order of consistency dictates the speed of convergence.

References