NSDE 1: LECTURE 2

TYRONE REES*

Recap

$$u' = f(t,u), \qquad u(t_0) = u_0.$$

No guarantee of a unique solution. However, there is one if the conditions for **Picard's theorem** are satisfied:

Let $R = \{z \in \mathbb{R}^2 : |t - t_0| \leq h, |u - u_0| \leq k\}$

- Suppose that $f(\cdot,\cdot)$ is a continuous function in $U \supset R$ with

$$M = \max_{(t,u)\in R} |f(t,u)|$$

- Suppose that $\exists L > 0$ s.t. $\forall (t,u_1),(t,u_2) \in R$,

$$|f(t,u_1) - f(t,u_2)| \leq L|u_1 - u_2|$$

  Lipschitz condition

- Suppose that $Mh \leq k$

Then there exists a unique continuously differentiable function $t \to u(t)$ satisfying the IVP:

$$u' = f(t,u), \qquad u(t_0) = u_0$$

for all $t \in [t_0 - h, t_0 + h]$.

Example 0.1.

$$u' = 3u^{2/3}, \qquad u(0) = 0$$

We know from last time that there is not a unique solution, so we should expect that u violates a condition of Picard's theorem.

Suppose there exist constants $h > 0$ *and* $k > 0$ *so that, if* $|t| \leq h$, $|u| \leq k$, *there is a constant* $L > 0$ *such that*

$$|f(t,u_1) - f(t,u_2)| \leq L|u_1 - u_2|.$$

Then, if we set $u_1 = u \in [0,k]$, $u_2 = 0$, *then we have that*

$$|3u^{2/3}| = 3|u|^{2/3} < L|u| \iff |u| > (L/3)^3$$

This is clearly a contradiction, as u can be arbitrarily close to zero.

```
ezplot('3*x^(3/2)',[0,2]);
```

*Rutherford Appleton Laboratory, Chilton, Didcot, UK, tyrone.rees@stfc.ac.uk

Before we dive into an example that is unique, we remind ourselves of a useful result:

THEOREM 0.2. **The Mean Value Theorem** *Suppose that a function g is defined and continuous on a closed interval $[a, b]$ on the real line, and that g is differentiable on the open interval (a, b).*

Then, for every $y_1, y_2 \in [a, b]$, there exists $\xi \in (y_1, y_2)$ such that

$$g(y_1) - g(y_2) = g'(\xi)(y_1 - y_2).$$

This is very useful for deriving Lipschitz constants!

EXAMPLE 0.3.

$$u' = u^2 \qquad u(0) = u_0$$

First, let us define the region

$$R = \{(t, u) \in \mathbb{R}^2 : |t| \leq h, |u - u_0| \leq k\}$$

for some $h > 0$ and $k > 0$.

Note that

$$\begin{aligned} |f(t, u_1) - f(t, u_2)| &= |u_1^2 - u_2^2| \\ &\leq 2(u_0 + k)|u_1 - u_2| \qquad \text{(by MVT)} \end{aligned}$$

So $f(\cdot, \cdot)$ has a Lipschitz constant of $2(u_0 + k)$ in any neighbourhood R, for any h and k.

Also, $\max |f(t, u)| = (k + u_0)^2$, so given any $k > 0$, taking $h \leq k/(k + u_0)^2$ will ensure that the conditions of Picard's theorem are satisfied.

We must be careful about what Picard's theorem says, though. The analytic solution here is

$$u = \frac{u_0}{1 - u_0 t},$$

which blows up at time $t = 1/u_0$.

```
ezplot('a/(1-a*x)',x=[0,1/a])
```

1. Solving ODEs numerically. How do we approximate the solution of ODEs?

Recall our standard problem:

$$u' = f(t, u), \text{ for } t \in [t_0, T_M], \qquad u(t_0) = u_0.$$

First, we seek approximate the solution at a discrete number of points in time:

Since we have the inital value, this will be exact, but the other points only approximate the true solution.

Notation

We approximate the solution of the ODE at points that we call

$$t_0, t_1, \ldots, t_m.$$

We will often (for simplicity) consider equally spaced points, so that

$$t_{n+1} = t_n + h, \text{ where } h = (t_n - t_0)/n.$$

The exact solution at these points would be

$$u(t_0), u(t_1), \ldots, u(t_m).$$

We call the approximate solution at these points

$$U_0, U_1, \ldots, U_m.$$

How can we approximate the solution?

$$\begin{aligned} u' &= f(t,u) \\ \lim_{t\to 0} \frac{u(t+h)-u(t)}{h} &= f(t,u) \\ \frac{u(t+h)-u(t)}{h} &\approx f(t,u) \\ u(t_n+h) &\approx u(t_n) + hf(t_n, u(t_n)) \end{aligned}$$

We can turn this approximation into an algorithm:

Euler's (explicit) method

$$\begin{aligned} U_0 &= u(t_0) \\ U_{n+1} &= U_n + h\, f(t_n, U_n) \end{aligned}$$

Systems

We developed this for first-order IVPs, but the method works more generally. For example, consider the system

$$\begin{aligned} u'' &= f(t,u,u') \\ u(0) &= u_0 \\ u'(0) &= u_0'. \end{aligned}$$

This can be re-written as a system of first order equations:

$$\begin{aligned} u' &= v \qquad v(0) = u_0' \\ v' &= f(t,u,v) \qquad u(0) = u(0), \end{aligned}$$

or, in vector form:

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}, \qquad \mathbf{u}(0) = \begin{bmatrix} u_0 \\ u_0' \end{bmatrix}$$

where

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \text{ and } \mathbf{f} = \begin{bmatrix} v \\ f(t, u, v) \end{bmatrix}$$

We can then apply Euler's method, say, in exactly the same way as for the scalar case:

$$\mathbf{U}_{n+1} = \mathbf{U}_n + h\mathbf{f}(t_n, \mathbf{U}_n).$$

This is just shorthand for writing Euler separately on each of the components:

$$\begin{aligned} U_{n+1} &= U_n + h\,V_n \\ V_{n+1} &= V_n + h\,f(t_n, U_n, V_n) \end{aligned}$$