HUMBOLDT-UNIVERSITÄT ZU BERLIN

BERNSTEIN CENTRE FOR COMPUTATIONAL NEUROSCIENCE

BERNSTEIN CENTER FOR COMPUTATIONAL NEUROSCIENCE
HUMBOLDT-UNIVERSITÄT ZU BERLIN
PHILIPPSTR. 13 HOUSE 6

PHONE: 030/2093-9110
FAX: 030/2093-6771
WEBPAGE: HTTP://WWW.BCCN-BERLIN.DE/

Models of Neural Systems I, WS 2007/08
Computer Practical 3
Discussed on 5th Nov 2007

Ordinary Differential Equations (ODEs)

Differential equations describe the evolution of systems in continous time and are widely used in science and engineering. Here, we will focus on the equations of the following form:

$$\dot{x}(t) = f\left(x(t), t\right), \tag{1}$$

where $\dot{x}(t) \equiv \frac{dx}{dt}$ is the first derivative of $x(t)$ with the respect to time t. The solution to this equation is a **function** (dynamical variable) which satisfies the above relation. This equation involves only first derivative with respect to a single independent variable and therefore it is called first-order ordinary differential equation. Here, you will revise how to find analytical solutions to such equations and learn solving them numerically.

Exercises

1. **Analytical solution of ODEs**

   Solve the following differential equations with the initial condition $x(0) = x_0$ (without using the computer).

   (a) $\frac{dx}{dt} = -x$; $x_0 = 1$

   (b) $\frac{dx}{dt} = x^{-1}$; $x_0 = 1$

   (c) $\frac{dx}{dt} = 1 - x$; $x_0 = 0$

   (d) (optional) $\frac{dx}{dt} = x(1 - x)$; $x_0 = \frac{1}{2}$

   Plot the results.

2. **Euler method**

   The simplest numerical method of solving the equation (**??**) is by discretization. Lets assume that the function $x(t)$ is constant over short time interval Δt. We can then rewrite the equation using finite steps:

$$x_{i+1} = x_i + f(x_i, t_i)\Delta t \tag{2}$$

In order to find the solution, we start with the value x_0 given by initial condition and then proceed recursively by adding small increments to the function according to the equation (**??**). This algorithm is called Euler method.

(a) Solve one of the problems from Exercise 1 with Euler method (using Python). Compare the analytical and numerical solutions.

(b) (Optional) Define a Python function which takes as an argument the function $f(x,t)$, initial condition, stop time and the intergration step Δt:

```
def logistic(x,t):
    return x*(x-1);
def euler(f_func,x_0, t_max,dt):
    ...

#Solve logistic ODE
x_t=euler(logistic, 0.5, 5., 0.01)
```

3. (*Advanced*) **Improved Euler method**

In practice we can improve the precision of numerical solutions to the ODEs using higher-order methods. The second-order method is:

$$\tilde{x}_{i+1} = x_i + f(x_i)\Delta t \tag{3}$$

$$x_{i+1} = x_i + \frac{1}{2}\left[f(x_i) + f(\tilde{x}_{i+1})\right]\Delta t \tag{4}$$

Show that the second-order method tends to make smaller error $E = |x(t_n) - x_n|$ for a given stepsize Δt.

(a) Expand $x(t_1) \equiv x(t_0 + \Delta t)$ as a Taylor series in Δt, through terms of $O(\Delta t^2)$.

(b) Show that the local error between the exact method and Euler approximation $|x(t_1) - x_1| \sim \Delta t^2$.

(c) Using the same arguments show that the second-order method is $O(\Delta t^3)$.

4. **Runge-Kutta method**

Fourth-order Runge-Kutta provides a good trade-off between computational cost and accuracy. It is also commonly used and is included in many packages for numerical methods. In SciPy you will find an advanced ODE solver which asks the user for the right side of the differential equation and detach him from the implementation details. The standard usage case is given by the following example:

```
from scipy import *
def f(x,t): return -x
x_0 = 0; t = arange(0, 5, 0.01)
x = integrate.odeint(f,x_0,t)
```

(a) Modify the above example to solve the differential equation given in 1(d)

(b) Compare the solution with: the exact solution and solution obtained with Euler method for several values of the integration step Δt.

Contact

Jan Benda (ITB, r. 1301) Phone: 2093-8652 Email: j.benda@biologie.hu-berlin.de

Robert Schmidt (ITB, r. 2316) Phone: 2093-8926 Email: r.schmidt@biologie.hu-berlin.de

Bartosz Telenczuk (ITB, r. 1309) Phone: 2093-8838 Email: b.telenczuk@biologie.hu-berlin.de