$ Supplement Note 6: Euler's Method

The *Euler method* is the simplest approach to solve the linear differential equation

$$\frac{df}{dx} = g(x)\ . \tag{1}$$

It is based on repeated use of formula (3.2) in the lecture note:

$$f(x+\Delta x) \approx f(x) + f'(x_0)\Delta x \qquad\qquad (3.2)\ ,$$

which is a direct consequence of the definition of the derivative (3.1): *if the magnitude of Δx is small enough then the magnitude of the difference $\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} - f'(x_0)$ will also be very small.* Explicitly, this is written as

$$\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} - f'(x_0) = err \quad \Rightarrow \quad f(x_0+\Delta x) = f(x_0) + f'(x_0)\cdot\Delta x + err\cdot\Delta x$$

where $|err| < \epsilon$ with ϵ being the desired error tolerance. In particular, the neglected term in (3.2) is *much smaller than* $|\Delta x|$.

As a first application of Euler's method, let us consider the problem of a freely falling object near Earth's surface. The gravitation force on the object is given by $F = -mg$, where m is the mass of the object, and $g = 9.8\mathrm{m/s}^2$. It follows from Newton's Law $F = ma$ that

$$\frac{dv}{dt} = a = -g \quad \Rightarrow \quad v(t+\Delta t) = v(t) - g\cdot\Delta t \tag{2}$$

$$\frac{dy}{dt} = v \quad \Rightarrow \quad y(t+\Delta t) = y(t) + v(t)\cdot\Delta t \tag{3}$$

In the code *FreeFall.py* three sets of increment $\Delta t = 0.5, 0.1, 0.05$ are used and the calculated values together with the exact results $v(t) = -gt, \quad y(t) = 100 - \frac{1}{2}gt^2$ are shown in the plot. Notice that all the curves except for $y(t)$ with $\Delta = 0.01$ are almost identical! Euler approximation is exact for linear functions.

```
# FreeFall.py
# simulation of freely falling of object near Earth’s surface
import numpy as np
import matplotlib.pyplot as plt
# calculation with dt = 0.5
t , y  , v = 0 , 100, 0 #initial condition
dt = 0.5
g = 9.8    #m/sˆ2 acceleration due to gravity
T1, y1, v1 = [],[],[]
while t < 4.5 :
    T1.append(t)
    y1.append(y)
    v1.append(v)
    v = v - g*dt
    y = y + v*dt
    t = t + dt
plt.plot(T1,y1, label= ’altitude, dt = 0.5’)
plt.plot(T1, v1, label = ’velocity, dt = 0.5’)
plt.title(’Free Fall’)
plt.xlabel(’time’)
# calculation with dt = 0.05
t , y  , v = 0 , 100, 0 #initial condition
dt = 0.05
T2, y2, v2 = [],[],[]
while y > 0 :
    T2.append(t)
    y2.append(y)
    v2.append(v)
    v = v - g*dt
    y = y + v*dt
    t = t + dt
plt.plot(T2, y2, label= ’altitude, dt = 0.05’)
plt.plot(T2, v2, label = ’velocity, dt = 0.05’)
```

```
# calculation with dt = 0.01
t , y  , v = 0 , 100, 0 #initial contition
dt = 0.01
T3, y3, v3 = [],[],[]
while y > 0 :
    T3.append(t)
    y3.append(y)
    v3.append(v)
    v = v - g*dt
    y = y + v*dt
    t = t + dt
plt.plot(T3, y3, label= 'altitude, dt = 0.01')
plt.plot(T3, v3, label = 'velocity, dt = 0.01')
# exact result: v = -gt; y = y_0- g *t*t/2
T = np.linspace(0, 4.5, 100)
V = - g* T
print(T)
Y = 100 - 0.5*g *T**2
plt.plot(T, Y, label= 'altitude, exact')
plt.plot(T, V, label = 'velocity, exact')
plt.legend()
plt.show()
```