Lecture 25

Monday, March 24, 2025 9:27 AM

Euler's method for the initial value problem $y' = xy$, $y(0) = y_0$
$y_n = y(x_n)$
The differential equation is approximated by
$y_{n+1} = y_n + hx_n y_n$

Python code:

```
from numpy import *
from matplotlib.pyplot import *
# solve y'=xy with initial condition y(0)=y0
y0 = 1
# with step size h
h = 0.2
# the number of steps
N = 10
# Array x = [x0,x1,...,xN]
x = linspace(0,N*h,N+1)
# Array y = [y0,y1,...,yN]
y = zeros(N+1)
y[0] = y0
for i in range(N):
    y[i+1] = y[i] + h*x[i]*y[i]
print(x)
print(y)
plot(x,y)
show()
```

This equation has an exact solution, which you can find by hand using the separation of variables method or the integrating factor method: $y = y_0 e^{\frac{x^2}{2}}$. You can draw the exact solution together with the approximate solution to compare them. Before the line show() in the code, add the following line:

```
plot(x, y0*e**(x**2/2))
```

Euler's method is an example of a more general method for numerically solving differential equations called *finite difference method*. The idea is that you approximate the derivative by the difference quotient.

Example:

$$y'(x) \approx \frac{y(x+h) - y(x)}{h}$$

$$y''(x) \approx \frac{y'(x+h) - y'(x)}{h}$$

$$\approx \frac{\frac{y(x+h) - y(x)}{h} - \frac{y(x) - y(x-h)}{h}}{h}$$

$$= \frac{y(x+h) - 2y(x) + y(x-h)}{h^2}$$

Example: $y'' + 2xyy' + y^2 = 0, y(-1) = 1, y'(-1) = 1$
$y(x_n) = y_n$
The differential equation is approximated by
$$\frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + 2x_n y_n \frac{y_n - y_{n-1}}{h} + y_n^2 = 0$$
Therefore,
$$y_{n+1} = 2y_n - y_{n-1} - 2hx_n y_n(y_n - y_{n-1}) - h^2 y_n^2$$

This initial value problem has an exact solution $y = \frac{2}{1+x^2}$.

Python code:

```
from numpy import *
from matplotlib.pyplot import *
# solve y''+2xyy'+y^2=0 with initial condition y(x0)=a and y'(x0)=b
x0 = -1
a = 1
b = 1
# with step size h
h = 0.01
# the number of steps
N = 200
# Array x = [x0,x1,...,xN]
x = linspace(x0,x0+N*h,N+1)
# Array y = [y0,y1,...,yN]
y = zeros(N+1)
y[0] = a
y[1] = a+b*h
for n in range(1,N):
    y[n+1] = 2*y[n] - y[n-1] - 2*h*x[n]*y[n]*(y[n]-y[n-1])-h**2*y[n]**2
plot(x,y)
plot(x,2/(1+x**2))
show()
```