$ Supplement note 1

You should install numpy(basic numerical arrays for faster computation), scipy(scientific packages based on c and c++ codes) and matplotlib(plotting packages) in addition to python. As a first application, let us find the root of the cubic function $f(x) = 0$, where

$$f(x) = 1 + x + x^2 + 2x^3 \tag{1}$$

Note that if the magnitude of x is large enough, the values of a cubic function f, $f(x)$ and $f(-x)$, are of opposite signs. It follows from the *Intermediate-Value Theorem* (**Theorem 1.9**) that there is a root x_0 lying in the interval $[-x, x]$. The solution may then be obtained by repeated bisecting of the interval. This is done in *method 1* of python code *t1.py*. An alternative approach is by rewriting the equation $f(x) = 0$ as

$$x + 2x^3 = -(1 + x^2) \quad \Rightarrow \quad x = -\frac{1 + x^2}{1 + 2x^2} \tag{2}$$

To avoid numerical cancellation, the terms in both numerator and denominator are of the same sign. The solution may be obtained by iterations of this function. This is done in *method 2* of *t1.py*. The last part of the program yields the graph of the function $f(x)$ using the *pyplot* program in the **matplotlib** package. During the constructing process, by using *numpy array xx*, the calculation of the values of $x * x$ is *broadcasting* for all the values of x in the array xx by a single statement $xx * xx$.

```
# t1.py :Find roots for cubic function  f(x) = 1+ x + x**2 + 2 x**3
import numpy as np
import matplotlib.pyplot as plt
#method 1
```

```
print('method 1: bisection')
xu = 10
print( 1 + xu*( 1 + xu*(1+ 2*xu)))
xd = -10
print( 1 + xd*( 1 + xd*(1+ 2*xd)))
for i in range(20):
    x = (xu+ xd)/2
    f = 1+ x*(1 + x*(1+ 2*x))
    print(x,f)
    if f<0: xd = x
    else:  xu = x
print('testing result', x,  1 + x*( 1 + x*(1+ 2*x)))
#method 2
print('method 2: iteration')
x = 0  # initial guess
for i in range(20):
    x = - ( 1 + x**2)/(1 + 2* x**2)
    print('x, f=', x, 1+ x*(1 + x*(1+ 2*x)) )
print('testing result', x,  1 + x*( 1 + x*(1+ 2*x)))
print('construct graph')
xx = np.linspace(-10, 10, 20)
xx2 = xx*xx
xx3 = xx*xx2
ff = 1+ xx + xx2 + 2*xx3
print('xx=', xx)
print('xx2=', xx2)
print('xx3=', xx3)
print('ff=', ff)
plt.plot(xx,ff,label='1+ x+ x^2+ 2x^3')
plt.title('cubic function')
plt.xlabel('x')
plt.legend()
plt.show()
```