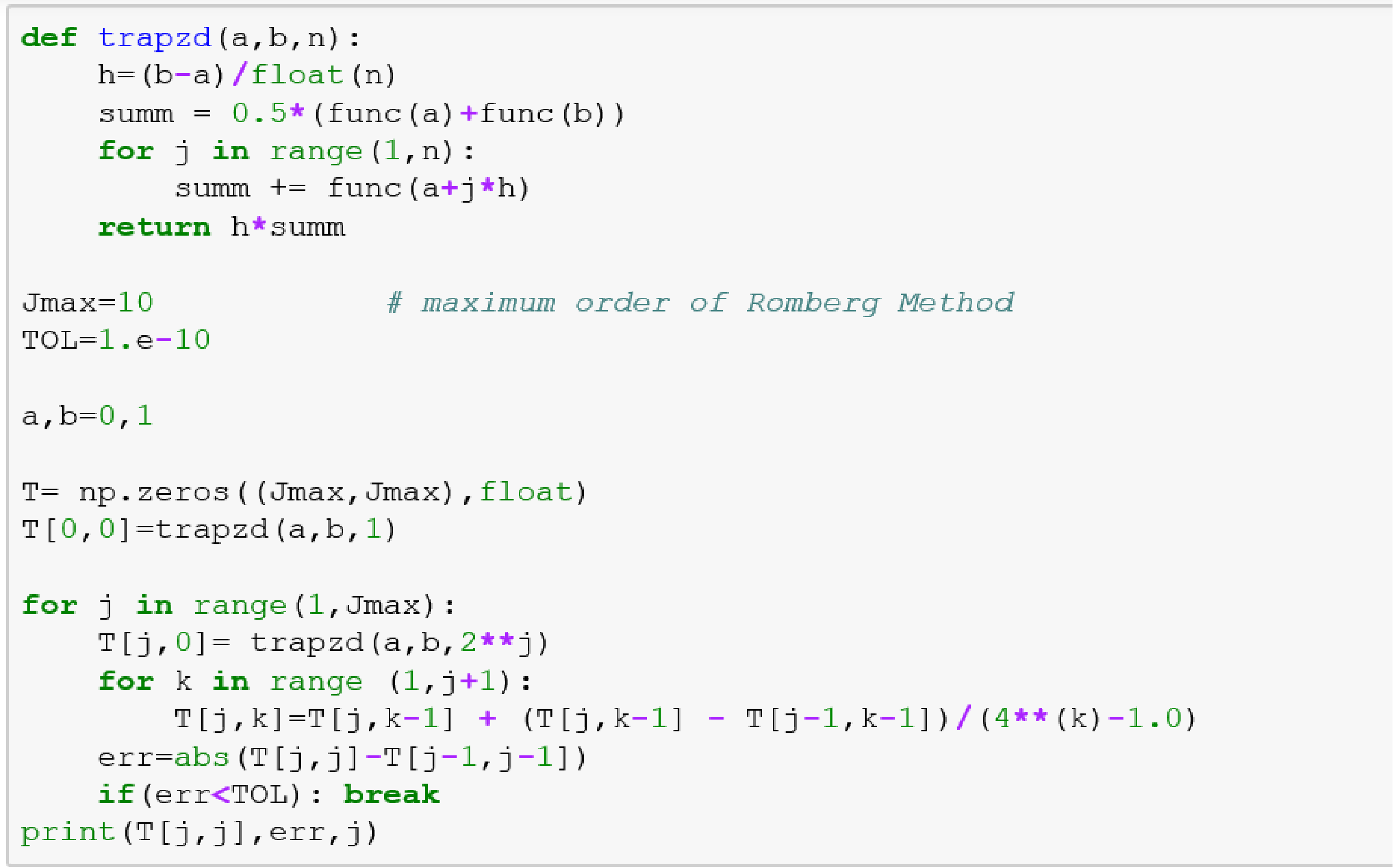

```
def trapzd(a,b,n):
    h=(b-a)/float(n)
    summ = 0.5*(func(a)+func(b))
    for j in range(1,n):
        summ += func(a+j*h)
    return h*summ

Jmax=10             # maximum order of Romberg Method
TOL=1.e-10

a,b=0,1

T= np.zeros((Jmax,Jmax),float)
T[0,0]=trapzd(a,b,1)

for j in range(1,Jmax):
    T[j,0]= trapzd(a,b,2**j)
    for k in range (1,j+1):
        T[j,k]=T[j,k-1] + (T[j,k-1] - T[j-1,k-1])/(4**(k)-1.0)
    err=abs(T[j,j]-T[j-1,j-1])
    if(err<TOL): break
print(T[j,j],err,j)
```