

```
In [ ]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

```
In [ ]: def func(x):
        return np.exp(-2*x)*np.cos(10*x)
```

```
In [ ]: def func_integral(x):
        return (5*np.exp(-2*x)*np.sin(10*x))/52. - (np.exp(-2*x)*np.cos(10*x))/52.
```

```
In [ ]: def trapezoid_core(f,x,h):
        return 0.5*h*(f(x+h) + f(x))
```

```
In [ ]: def trapezoid_method(f,a,b,N):
        x = np.linspace(a,b,N)
        h = x[1]-x[0]

        Fint = 0.0

        for i in range(0,len(x)-1,1):
            Fint += trapezoid_core(f,x[i],h)
        return Fint
```

```
In [ ]: def simpson_core(f,x,h):
        return h*(f(x) + 4*f(x+h) + f(x+2*h))/3
```

```
In [ ]: def simpsons_method(f,a,b,N):
        x = np.linspace(a,b,N)
        h = x[1]-x[0]

        Fint = 0.0

        for i in range(0,len(x)-2,2):
            Fint += simpson_core(f,x[i],h)
        if((N%2)==0):
            Fint += simpson_core(f,x[-2],0.5*h)
        return Fint
```

```
In [ ]: def romberg_core(f,a,b,i):
        h = b-a
        dh = h/2.**i
        K = h/2.**i+1
        M = 0.0
        for j in range(2**i):
            M += f(a + 0.5*dh + j*dh)
        return K*M
```

```
In [ ]: def romberg_integration(f,a,b,tol):
        i = 0
        imax = 1000
        delta = 100.0*np.fabs(tol)

        I = np.zeros(imax,dtype=float)
        I[0] = 0.5*(b-a)*(f(a) + f(b))
        i += 1

        while(delta>tol):
            I[i] = 0.5*I[i-1] + romberg_core(f,a,b,i)
            delta =np.fabs( (I[i]-I[i-1])/I[i])
            print(i,I[i],I[i-1],delta)

            if(delta>tol):
                i+=1
                if(i>imax):
                    print("Max iterations reached.")
                    raise StopIteration('Stopping iterations after ',i)
        return I[i]
```

```
In [65]: Answer = func_integral(np.pi)-func_integral(0)
print(Answer)
print("Trapezoid")
print(trapezoid_method(func,0,np.pi,10))
print("Simpson's Method")
print(simpsons_method(func,0,np.pi,10))
print("Romberg")
tolerance = 1.0e-6
RI = romberg_integration(func,0,1,tolerance)
print(RI-Answer)/Answer, tolerance
```

```
23 0.014335207263496661 0.014335227461638966 1.408988508729909e-06
24 0.014335197164426737 0.014335207263496661 7.044946650012679e-07
-0.0048596597061173405
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-65-59f29568c514> in <module>
      8 tolerance = 1.0e-6
      9 RI = romberg_integration(func,0,1,tolerance)
----> 10 print(RI-Answer)/Answer, tolerance

TypeError: unsupported operand type(s) for /: 'NoneType' and 'float'
```

```
In [ ]:
```

