In [18]:
```
import math as math

#Simpson 1/3 rule
def simpson(f,a,fa,b,fb):
    m=0.5*(a+b)
    fm=f(m)
    return (m,fm,(b-a)*(fa+4*fm+fb)/6.0)

#adaptive Simpson rule: split the interval a,b and apply Simpson's
rule to left and right half
#avoid multiple evaluations of f
def adaptive_simpson(f,a,fa,b,fb,eps,lastint,m,fm):
    lm,flm,leftint=simpson(f,a,fa,m,fm)
    rm,frm,rightint=simpson(f,m,fm,b,fb)
    deltaint=leftint+rightint-lastint
    if abs(deltaint)<=15*eps:
        return leftint+rightint+deltaint/15.0
    return adaptive_simpson(f,a,fa,m,fm,0.5*eps,leftint,lm,flm)+ad
aptive_simpson(f,m,fm,b,fb,0.5*eps,rightint,rm,frm)

def AdapSimpson_integrate(f,a,b,eps):
    fa,fb=f(a),f(b)
    m,fm,inint=simpson(f,a,fa,b,fb)
    return adaptive_simpson(f,a,fa,b,fb,eps,inint,m,fm)

#test: Integrate[Sin[x], {x, 0, 1}]
I=1-math.cos(1)
IS=AdapSimpson_integrate(math.sin,0,1,1e-9)
print("Correct result: ",I)
print("Adap. Simpson   ",IS)
print("absolute error  ",abs((I-IS)))
print("relative error  ",abs((I-IS)/I))
```

```
Correct result:  0.45969769413186023
Adap. Simpson    0.4596976941317858
absolute error   7.444045380111675e-14
relative error   1.6193349401436013e-13
```

In [ ]:
```

```