```
syms x;
y1 = inline('8 + 4*cos(x)')

y =@(x) 8 + 4*cos(x);

%int(y(x))

a = 0;
b = pi/2;
analytical_sol = double(int(y1(x),a,b))

%trapezoidal rule

c = (b-a)/2;

integ_trape = c * (y(a) + y(b))

x = linspace(a,b,3);
x1 = linspace(a,b,5);
h = (b-a)/2;

integ_trap_2seg = h/2 * ( y(x(1)) + 2*y(x(2)) + y(x(3)) )

integ_trap_4seg = (h/4)*( y(x1(1)) + 2*(y(x1(2)) + y(x1(3)) +
 y(x1(4))) + y(x1(5)) )

integ_sim13rd = (h/3) * (y(a) + 4*y(a+b/2) + y(b) )

x2 = linspace(a,b,7);  %% in Simpsons 3/8th rule we generally divide
 the interval into n segments where n is a multiple of 3 I take n = 6
 so 7 points

%and h will be (b-a)/6

integ_sim38 = (3*h/24)*( y(x2(1)) + y(x2(end)) + 2*(y(x2(4))) +
 3*( y(x2(2)) + y(x2(3)) + y(x2(5)) + y(x2(6)) ) )


y1 =

     Inline function:
     y1(x) = 8 + 4*cos(x)


analytical_sol =

   16.5664


integ_trape =

   15.7080
```

```
integ_trap_2seg =

   16.3586


integ_trap_4seg =

   16.5148


integ_sim13rd =

   16.5755


integ_sim38 =

   16.5666
```

composite simpson's rule with 4 segments

```
xn = linspace(a,b,5);  %%for composite simpson's rule with 4 th
 segments;
% where h is (b-a)/4 xj = a + j*h;
hn = (b-a)/4;
integ_compsimp_rule = (hn/3).*( y(xn(1)) + y(xn(5)) + 2*y(xn(3)) +
 4*(y(xn(2))+y(xn(4))) )
```

```
integ_compsimp_rule =

   16.5669
```

Relative error calculations

```
relaerr_integ_trap  =  ((integ_trape-analytical_sol)/
analytical_sol)*100
relaerr_integ_trap_2seg = ((integ_trap_2seg-analytical_sol)/
analytical_sol)*100
relaerr_integ_trap_4seg = ((integ_trap_4seg-analytical_sol)/
analytical_sol)*100
relaerr_integ_sim13rd =  (( integ_sim13rd- analytical_sol)/
analytical_sol)*100
relaerr_integ_sim38  = (( integ_sim38-analytical_sol )/
analytical_sol)*100
relaerr_integ_compsimp_rule = (( integ_compsimp_rule -analytical_sol)/
analytical_sol)*100
```

```
relaerr_integ_trap =

   -5.1816

relaerr_integ_trap_2seg =

   -1.2541

relaerr_integ_trap_4seg =

   -0.3111

relaerr_integ_sim13rd =

    0.0550

relaerr_integ_sim38 =

    0.0014

relaerr_integ_compsimp_rule =

    0.0032
```

Published with MATLAB® R2018b