COMP 350 Solutions to Assignment 6

1. (12 points)

   (a) (6 points)
   The number of function evaluations is 17.
   The final result is 0.999975503966687, and $n = 4$ so we have $2^4 = 16$ subintervals.

```
% f.m
function y = f(x)
y=2/sqrt(pi)*exp(-power(x,2));

% trapezoid.m
function y = trapezoid(a,b,epsilon)
n=0;
count=2;
h=b-a;
I_old = 0;
I_new = h/2*(f(a)+f(b));
while abs(I_new-I_old)>epsilon
    n = n + 1;
    I_old = I_new;
    I_new = 0;
    h = h/2;
    for i=1:2^(n-1)
        I_new = I_new + f(a+(2*i-1)*h);
        count = count + 1;
    end;
    I_new = I_old/2 + h*I_new;
end;
count
n
y = I_new;


Result:
>> trapezoid(0,3,1e-5)
count = 17
n = 4
ans =  0.999975503966687
```

   (b) (6 points)
   The minimal number of function evaluations is 41.
   In each recursive call, we only need 2 function evaluations, that is $f(d)$ and $f(e)$. And we

can get $f(a), f(b), f(c)$ from previous call. We call the adapt Simpson method 19 times, so the total number of function evaluations is $3 + 2 * 19 = 41$.

Note: Here the errors of the computed integration by the two methods used in 1(a) and 1(b) are similar. But the adaptive Simpson's method needs more function evaluations than the recursive trapezoid method, i.e., the former is not as efficient as the latter. However, when epsilon is smaller than 1.e-6, we can see the former will take less number of function evaluations than the latter and furthermore the former's result is also more accurate than the latter's.

```
% adapt_simpson.m
function numI = adapt_simpson(f,a,b,epsilon,level,level_max,fa,fb,fc)
h = b-a;
I1 = h*(fa+4*fc+fb)/6;
level = level+1;
c = (a+b)/2;
d = (a+c)/2;
e = (c+b)/2;
fd = feval(f,d);
fe = feval(f,e);
I2 = h*(fa+4*fd+2*fc+4*fe+fb)/12;
global fcount;
fcount = fcount + 2;
if level>=level_max
    numI = I2;
else
    if abs(I2-I1)<=15*epsilon
        numI = I2;
    else
        numI = adapt_simpson(f,a,c,epsilon/2,level,level_max,fa,fc,fd)...
            +adapt_simpson(f,c,b,epsilon/2,level,level_max,fc,fb,fe);
    end;
end;

% simpson_test.m
global fcount;
a = 0;
b = 3;
c = (a+b)/2;
fa = feval('f',a);
fb = feval('f',b);
fc = feval('f',c);
fcount = 3;
adapt_simpson('f',a,b,1e-5,1,30,fa,fb,fc)
fcount
```

```
Result:
>> simpson_test
ans =  0.999977472878837
fcount = 41
```

2. (8 points)

(a) (4 points)
Let $x = \frac{b-a}{2}t + \frac{a+b}{2}$,

$$\begin{aligned}
&\int_a^b f(x)dx \\
= &\frac{b-a}{2}\int_{-1}^{1} f(\frac{b-a}{2}t + \frac{a+b}{2})dt \\
\approx &\frac{b-a}{2}[f(\frac{b-a}{2}\cdot(-\frac{\sqrt{3}}{3}) + \frac{a+b}{2}) + f(\frac{b-a}{2}\cdot\frac{\sqrt{3}}{3} + \frac{a+b}{2})]
\end{aligned}$$

Then divided $[a,b]$ into n intervals, and apply the formula above, we have

$$\begin{aligned}
\int_a^b f(x)dx &= \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)dx \qquad (x_i = a + ih,\ h = \frac{b-a}{n}) \\
&= \frac{h}{2}\sum_{i=0}^{n-1}[f(-\frac{h}{2\sqrt{3}} + a + ih + \frac{h}{2}) + f(\frac{h}{2\sqrt{3}} + a + ih + \frac{h}{2})] \qquad (h = \frac{b-a}{n})
\end{aligned}$$

(b) (4 points)
The result is more accurate than 1(a). In fact, the absolute difference between this result and erf(3) is about 1.04×10^{-7}, while the absolute difference between the result obtained in 1(a) and erf(3) is about 2.41×10^{-6}. The computation involves 16 function evaluations, one less than that of 1(a).

Note: When epsilon becomes smaller than 10^{-4}, the difference between the two methods becomes larger.

```
% GQ_composite.m
a=0;
b=3;
n=floor(8.5);
h=(b-a)/n;
numEval=0;
numI=0;
for i=0:n-1
    f1=feval('f',-h/2/sqrt(3)+a+i*h+h/2);
    f2=feval('f',h/2/sqrt(3)+a+i*h+h/2);
    numI=numI+f1+f2;
    numEval=numEval+2;
end;

numI=h*numI/2
numEval

Result:
>> GQ_composite

numI = 0.999978013813411
numEval = 16

>> erf(3)
ans =0.999977909503001
```