C1

For $n=3$, the Gauss nodes are $-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}$ and the Gauss weights are $\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$. Write a Matlab code that verifies numerically that $I_3 = I$ for $f(x) = x^k$ with $0 \le k \le 5$ but $I_3 \neq I$ for $k = 6$. Specifically, print the errors $E_n = I_n - I$ for each of these cases. What happens to E_n for $k = 7, 8, 9, 10$? How do you explain this?

```
nodes=[-sqrt(3/5) 0 sqrt(3/5)];
weights=[5/9 8/9 5/9];
for k=0:20
    fprintf('E_%d = %.10f\n',k,sum(nodes.^k.*weights)-integral(@(x)x.^k,-1,1))
end
```

```
E_0 = 0.0000000000
E_1 = 0.0000000000
E_2 = 0.0000000000
E_3 = 0.0000000000
E_4 = 0.0000000000
E_5 = -0.0000000000
E_6 = -0.0457142857
E_7 = -0.0000000000
E_8 = -0.0782222222
E_9 = 0.0000000000
E_10 = -0.0954181818
E_11 = -0.0000000000
E_12 = -0.1020061538
E_13 = -0.0000000000
E_14 = -0.1022293333
E_15 = -0.0000000000
E_16 = -0.0989846588
E_17 = -0.0000000000
E_18 = -0.0940657179
E_19 = -0.0000000000
E_20 = -0.0885196312
```

When $k = 7, 9,\ f(x) = x^k$ is odd function, therefore $\int_{-1}^{1} x^k \, \mathrm{d}x = 0$, and $E_k = \frac{5}{9}\left(-\sqrt{\frac{3}{5}}\right)^k + \frac{8}{9}0^k + \frac{5}{9}\left(\sqrt{\frac{3}{5}}\right)^k = 0$, therefore $E_k = 0$;

When $k = 8, 10,\ E_k \neq 0$ and we see that $|E_k|$ increases as k increases, because the degree of $f(x)$ increases but we still use three nodes to approximate it.

C2

Let A be the $n \times n$ tridiagonal symmetric *Jacobi matrix* whose entries are all zero apart from the numbers

$$\frac{1}{\sqrt{1 \cdot 3}}, \frac{2}{\sqrt{3 \cdot 5}}, \frac{3}{\sqrt{5 \cdot 7}}, \cdots, \frac{n-1}{\sqrt{(2n-3) \cdot (2n-1)}}$$

in the upper-diagonal positions $(1,2), (2,3), \cdots, (n-1, n)$ of the matrix and also in the lower-diagonal positions $(2,1), (3,2), \cdots, (n, n-1)$. Let $\{\lambda_j\}$ and $\{v_j\}$ be the eigenvalues and eigenvectors of A as computed e.g. by the Matlab `eig` command. (In particular, each eigenvector should be normalized so that the sum of the squares of its entries equal to 1.)

Golub and Welsh showed that the node x_n is the eigenvalue λ_j, and the weight w_j is twice the square of the first component of the corresponding eigenvector v_j. Use thees properties to write a Matlab function `[x,w]=gaussq(n)` that returns vectors of the n Gauss quadrature nodes and weights for any n. Verify that for $n = 3$, you get the results given in (C1).

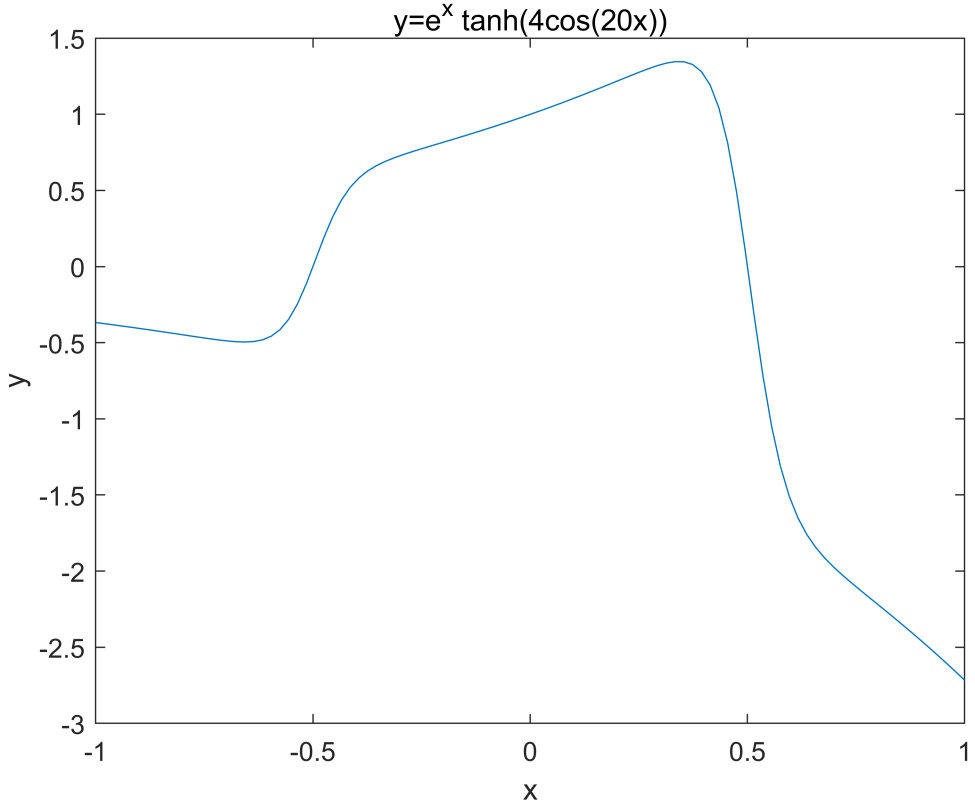
```
[nodes1,weights1]=gaussq(3)
```

```
nodes1 = 3×1
   -0.7746
         0
    0.7746
weights1 = 3×1
    0.5556
    0.8889
    0.5556
```

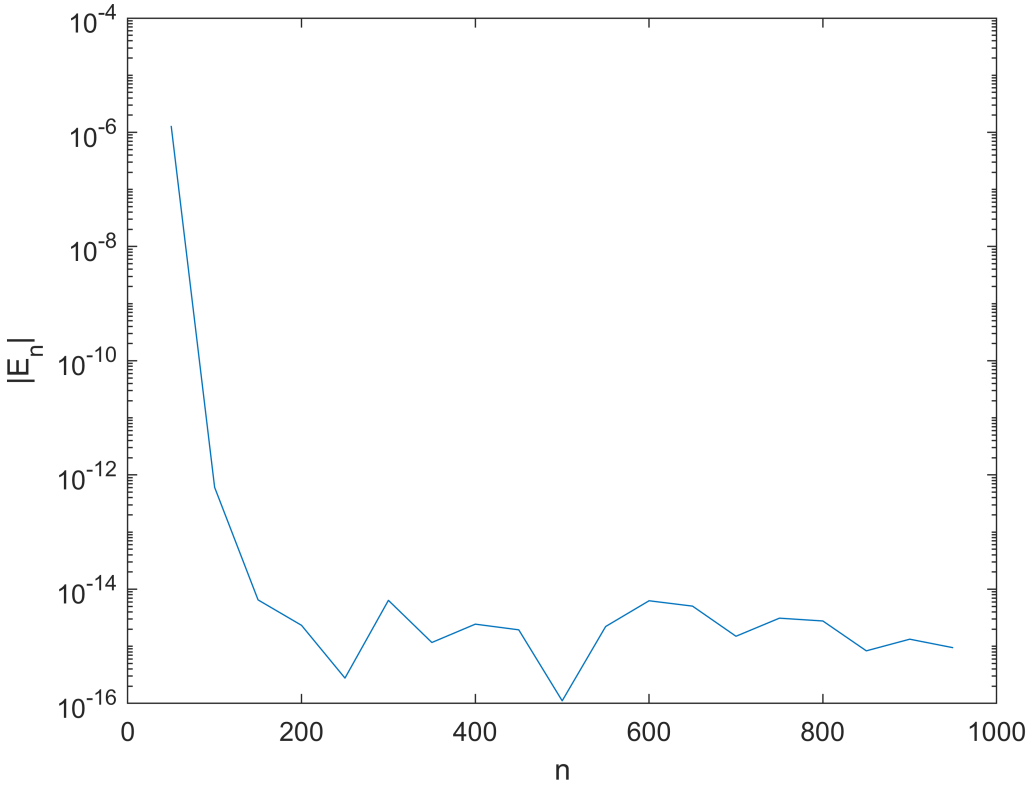
C3

Consider the function $f(x) = e^x \tanh(4\cos(20x))$ for $x \in [-1, 1]$, where tanh is as usual the hyperbolic tangent. Make a plot of f and estimate its integral I by Gauss quadrature with $n = 1000$. Let's call this estimate I even though of course it is not quite exact. Now make a semilogy plot of $|E_n|$ against n for $n = 50, 100, 150, \cdots, 950,$ making sure that the plot shows clearly where the data points lie. Comment on the shape of this curve, both its early stages and its later stages. (As it happens, the rate of convergence is determined by the locations of the singularities of $f(x)$ in the complex plane.)

```
f=@(t)exp(t).*tanh(4*cos(pi*t));
plot(linspace(-1,1),f(linspace(-1,1)))
title('y=e^x tanh(4cos(20x))');xlabel('x');ylabel('y');
```




```
[x,w]=gaussq(1000);
I=sum(f(x).*w);
E=zeros(19,1);
for i=1:19
    n=50*i;
    [x,w]=gaussq(n);
    E(i)=abs(sum(f(x).*w)-I);
end
semilogy(linspace(50,950,19),E,'-')
xlabel('n');ylabel('|E_n|');
```



In its early stages, the error decreases as n increases. This is because the approximation is better with more nodes. The error is smallest ($\approx 10^{-16}$) when $n = 800$, because the distance between nodes are closest to the distance between singularities. When $n > 800$, the error increases.

C4

Determine analytically the exact integrals over [-1,1] of the functions

(a) $f(x) = \dfrac{1}{1+25x^3}$

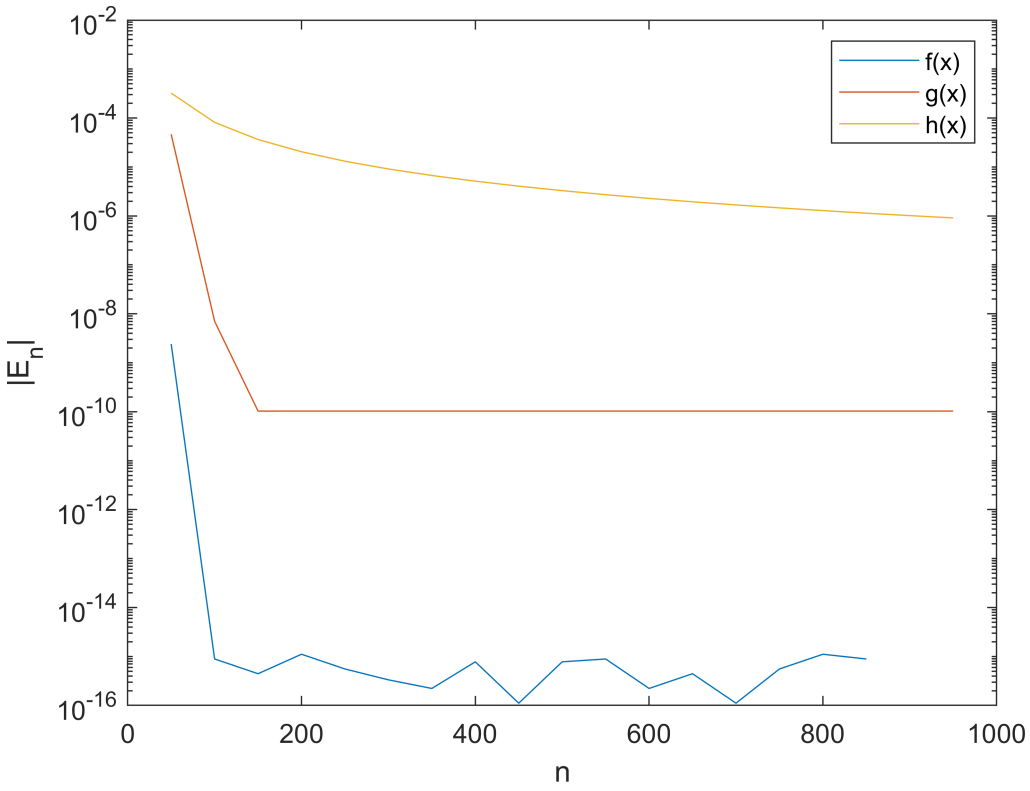
(b) $g(x) = \tanh\left(20\left(x+\dfrac{1}{2}\right)\right)$

(c) $h(x) = |x|$

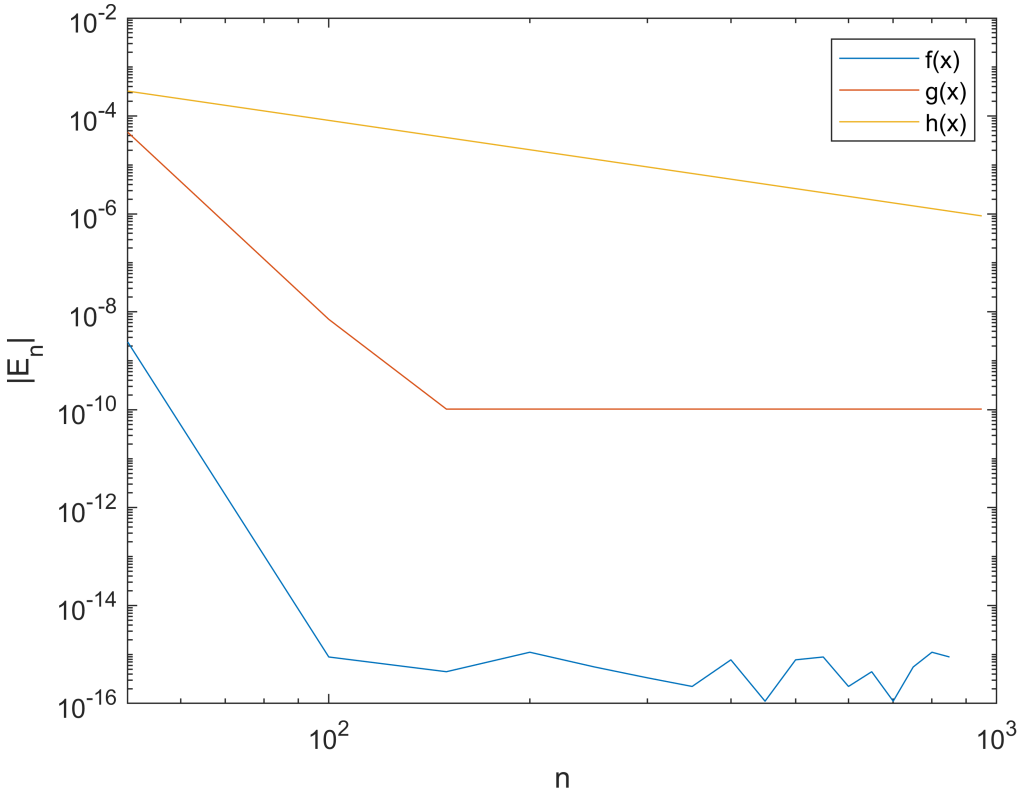
(You do not have to show your working.) Then make two plots with clearly labeled curves indicating convergence of n-point Gauss quadrature for these three integrands as a function of n. One plot should be on semilogy axes, and the other on loglog axes. Approximately how large must n be for 6-digit accuracy in each of the three cases? What do the shapes of the curves on the two axes indicate about convergence rates?

```
f=@(t)1./(1+25*t.^2);If=(2*atan(5))/5;
g=@(t)tanh(20.*(t+1/2));Ig=1;
h=@(t)abs(t);Ih=1;
Ef=zeros(19,1);Eg=Ef;Eh=Ef;
for i=1:19
    n=50*i;
```

```
    [x,w]=gaussq(n);
    Ef(i)=abs(sum(f(x).*w)-If);
    Eg(i)=abs(sum(g(x).*w)-Ig);
    Eh(i)=abs(sum(h(x).*w)-Ih);
end
semilogy(linspace(50,950,19),Ef,linspace(50,950,19),Eg,linspace(50,950,19),Eh)
xlabel('n');ylabel('|E_n|');legend('f(x)','g(x)','h(x)')
```




```
loglog(linspace(50,950,19),Ef,linspace(50,950,19),Eg,linspace(50,950,19),Eh)
xlabel('n');ylabel('|E_n|');legend('f(x)','g(x)','h(x)')
```



Approximately, for 6-digit accuracy, $n > 50$ is sufficient for f, $n > 100$ is sufficient for g, $n > 900$ is sufficient for h.

In the second axes, the graph is approximately linear, and from the slope we can see that f converges faster than g and g converges faster than h.

```
function [x,w]=gaussq(n)
    array=2:n;
    array=(array-1)./sqrt((2*array-3).*(2*array-1));
    [V,D]=eig(diag(array,1)+diag(array,-1));
    x=diag(D);
    w=(2*V(1,:).^2)';
end
```