Problem 1

The error in the Trapezoid Rule for \( N \) nodes separated by a distance \( \frac{2\pi}{N} \) is given by

\[
E_N = \int_0^{2\pi} e^{\cos(x)} \, dx - \frac{2\pi}{N} \left[ \frac{1}{2} e^{\cos(x_0)} + \sum_{i=1}^{N-1} e^{\cos(x_i)} + \frac{1}{2} e^{\cos(x_N)} \right]
\]

By the periodicity of the function, this is equal to

\[
E_N = \int_0^{2\pi} e^{\cos(x)} \, dx - \frac{2\pi}{N} \sum_{i=0}^{N-1} e^{\cos(x_i)}.
\]

The values \( e^{\cos(x_i)} \) can be calculated using the asymptotic series given in the problem. This yields the double-sum

\[
E_N = \int_0^{2\pi} e^{\cos(x)} \, dx - \frac{2\pi}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{\infty} a_k \cos(kx_i)
\]

Substituting in the given definition for \( a_n \),

\[
E_N = \int_0^{2\pi} e^{\cos(x)} \, dx - \frac{2\pi}{N} \sum_{i=0}^{N-1} \sum_{k=1}^{\infty} \frac{\cos(kx_i)}{2^{k-1}k!}
\]

\[
= \int_0^{2\pi} e^{\cos(x)} \, dx - \frac{2\pi}{N} \sum_{i=0}^{N-1} \frac{\cos(kx_i)}{2^{k-1}k!}
\]

This series is summable because the inner series converges (by the ratio test) and the outer sum is finite. I could sum this by hand (using the fact that \( \sum_{k=1}^{\infty} \frac{1}{2^{k-1}k!} = 2\sqrt{e} - 2 \) is closely related to the series that defines the value \( e \)), but I have an analysis exam tomorrow, so to free-up some study time I used Mathematica to find

\[
|E_N| < 10^{-60} \iff N > 40.
\]

Now let's try this with Simpson's formula. The error in Simpson's formula is

\[
E_N = \int_0^{2\pi} e^{\cos(x)} \, dx - \frac{2\pi}{3N} \left[ 2e^{\cos(x_0)} + \frac{N}{2} e^{\cos(x_{N/2})} + 4 \sum_{j=1}^{N/2-1} e^{\cos(x_{2j-1})} + \frac{1}{2} e^{\cos(x_{N/2})} \right].
\]
where I combined the $x_0$ and $x_N$ terms. Substituting the Fourier expansion of $e^{\cos(x)}$ and the given
definition for the $a_n$ terms, as before, we have the sum

$$E_N = -\frac{2\pi}{3N} \left[ 2 \sum_{k=1}^{\infty} \frac{\cos(kx_0)}{2k-1} + 4 \sum_{j=1}^{N/2} \sum_{k=1}^{\infty} \frac{1}{2k-1} \cos(kx_{2j-1}) + 2 \sum_{j=1}^{N/2-1} \sum_{k=1}^{\infty} \frac{1}{2k-1} \cos(kx_{2j}) \right].$$

Summing this formula using Mathematica, I find that

$$|E_n| < 10^{-60} \iff N > 82.$$ 

In this case, the Trapezoidal Rule gives a much smaller error.

**Problem 2**

The Mathematica code is pictured below.

```mathematica
Solve[Normal[Series[Integrate[g[x], (x, 0, 3)], {x, 0, 3}]]

a (Function[x, Evaluate[Normal[Series[D[g[x], (x, 0)], (x, x12, 3)])]])[[x12 - h/2]]

b (Function[x, Evaluate[Normal[Series[D[g[x], (x, 1)], (x, x12, 3)])]])[[x12 - h/4]]

c (Function[x, Evaluate[Normal[Series[D[g[x], (x, 1)], (x, x12, 1)])]])[[x12 - h/2]]

x12 = {h -> h/12}

Solve[Normal[Series[Integrate[g[x], (x, h, 0, 3)], {x, 0, 3}]]

a (Function[x, Evaluate[Normal[Series[D[g[x], (x, 0)], (x, x12, 3)])]])[[x12 - h/2]]

b (Function[x, Evaluate[Normal[Series[D[g[x], (x, 1)], (x, x12, 1)])]])[[x12 - h/2]]

c (Function[x, Evaluate[Normal[Series[D[g[x], (x, 1)], (x, x12, 1)])]])[[x12 - h/2]]

x12 = {h -> h/12}
```

The general algorithm can be defined inductively. If you have $n$ coefficients and you want $n+1$
coefficients, you must change the first line to

$$\text{Solve} \left[ \text{Normal} \left[ \text{Series} \left[ \int_{-h/2+x_{1/2}}^{h/2+x_{1/2}} g[x] dx, \{h, 0, 2n - 1\} \right] \right] \right],$$

and add the line

$$k \left( \text{Function} \left[ \text{Normal} \left[ \text{Series} \left[ D \left[ g(x), \{x, 2n - 3\} \right], \{x, x_{1/2}, 1\} \right] \right] \right) \right).$$

Solving for $k$ yields the $(n+1)^{th}$ coefficient.
Problem 3

For this problem I will let \( \Pi_n \) be the set of all polynomials of degree less than or equal to \( n \), as is standard in polynomial theory. Fix \( w_j \). Our goal is to conclude that \( w_j > 0 \), which will suffice to show that all weights must be positive. Since it is assumed that \( w_j \) is the same for any function, we have a choice in which function we want to use. Consider

\[
\Psi(x) = \prod_{i=1}^{n} (x - x_i) \in \Pi_n.
\]

Moreover, we see that

\[
\Psi^2(x) = \left( \prod_{i=1}^{n} (x - x_i) \right)^2 = \prod_{i=1}^{n} (x - x_i)^2 \in \Pi_{2n}.
\]

We can tweak this function to exclude the factor \( (x - x_i) \), which we denote \( \Psi_j \):

\[
\Psi_j^2(x) = \prod_{k \neq j} (x - x_k)^2 \in \Pi_{2n - 1}.
\]

Now, since \( \Psi_j^2 \geq 0 \) and is not identically zero, and since \( w(x) > 0 \) by assumption, it must be the case that

\[
\int_{\phi} \Psi_j^2(x) w(x) \, dx > 0.
\]

Moreover, since the quadrature formula is exact for all polynomials in \( \Pi_{2n - 1} \), it is exact for \( \Psi_j^2 \), and so

\[
\sum_{i=1}^{n} w_i \Psi_j^2(x_i) > 0.
\]

Now, observe that

\[
\Psi_j^2(x_k) = \begin{cases} 
0, & k \neq j \\
> 0, & k = j
\end{cases}
\]

5
which means that
\[
\sum_{i=1}^{n} w_i \Psi_j^2(x_i) = w_j \Psi_j^2(x_j) + \sum_{k \neq j} w_k \Psi_k^2(x_k) \\
= w_j \Psi_j^2(x_j) + 0 \\
> 0 \quad \text{by (1)}
\]
That is, since \( \Psi_j^2(x_j) > 0 \), we conclude that \( w_j > 0 \). \( \square \)

**Problem 4**

In this problem, let \( f_1(x) = e^x \) and \( f_2(x) = 1/(1+16x^2) \). Note that we can compute the exact integral for both functions:

\[
\int_{-1}^{1} e^x = e - \frac{1}{e}, \quad \int_{-1}^{1} \frac{1}{1+16x^2} = \frac{\arctan(4)}{2}
\]

The plot of the absolute errors for \( f_1 \) is below:

The plot of the absolute errors for \( f_2 \) is below:
The first thing to note is that both methods do a better job for $f_1$, although GQ converges almost instantly whereas CC needs $n = 3$ to start getting a low error. CC also does a better job for $f_2$, but the difference is not as exaggerated. Interestingly, the errors for $f_2$ oscillate slightly until about $n = 6$. The source code for this problem is included below.

```matlab
function problem4()
    clc;
    close all;
    nstart = 2;
    nend = 12;
    x = linspace(nstart, nend, nend - nstart + 1);
    ccw = [];
    ccw2 = [];
    ccw3 = [];
    ccw4 = [];
    for i = 1:length(x)
        n = x(i);
        [xs, ws] = GQ(n-1);
        f1 = [];
        f2 = [];
        for j = 1:n
            f1 = [f1 f1(ws(i+j))];
            f2 = [f2 f2(ws(j+j))];
        end
        q1 = dot(f1, ws);
        q2 = dot(f2, ws);
        ccw = [ccw abs(integral1() * q1)];
        ccw2 = [ccw2 abs(integral2() * q2)];
        f1_app = @(x) exp(1).*x;
        f2_app = @(x) 1./((1 + 16.*x.^2));
        ccw3 = [ccw3 abs(integral1() * CC(f1_app, n 1))];
        ccw4 = [ccw4 abs(integral2() * CC(f2_app, n 1))];
    end
```
function res = integral1()
    c = exp(1);
    res = c - 1/c;
end

function res = integral2()
    res = atan(4/2);
end

function res = f1(x)
    res = exp(x);
end

function res = f2(x)
    res = 1/(1 + 16*x^2);
end

function [x, w] = CGQ(n)
    beta = .5./sqrt(1 + (2*(1:n)).^2);
    T = diag(beta, 1) + diag(beta, -1);
    [v, D] = eig(T);
    x = diag(D); [x, i] = sort(x);
    w = 2*v([1, 1]' - 2);
end

function I = CCQ(1:n)
    x = cos(pi*(0:n)' / n);
    I = feval(I, x) / (2*n);
    y = real(ffft(x(1:n-1 - 1:2)));
    a = [y(2:n); y(2-1:n); y(1:n-2); y(n+1)];
    w = 0*a'; w(1:2:end) - 2./(1 + (0:2:n).^2);
    I = w*a;
end