Problem 1.
Let \( f(x) = xe^{-x^2} \).

a) Find the degree \( 2n + 1 \) Taylor polynomial for \( f(x) \), about the point \( x_0 = 0 \).

Solution
First note that
\[
e^t = \sum_{k=0}^{\infty} \frac{t^k}{k!}
\]
We could substitute and apply a derivative, or substitute and construct the desired sequence. We choose the latter approach for brevity. By substitution we obtain
\[
e^{-x^2} = \sum_{k=0}^{\infty} \frac{(-x^2)^k}{k!} = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{k!}
\]
Then multiply by \( x \) to obtain
\[
x e^{-x^2} = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{k!}
\]
We can truncate the infinite series to obtain a Taylor approximation of degree \( 2n + 1 \) of function \( f \) as
\[
q_{2n+1}(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{k!}
\]

b) Bound the error in degree \( 2n + 1 \) approximation for \( |x| \leq 2 \).

Solution
Note that
\[
e^t = p_n(t) + R_n(t) \implies e^{-x^2} = p_n(-x^2) + R_n(-x^2)
\]
Which gives
\[
x e^{-x^2} = \left( x p_n(-x^2) + x R_n(-x^2) \right)
\]
Taylor poly. \( q_{2n+1} \) error term \( E_{2n+1} \)

Then
\[
|f(x) - q_{2n+1}(x)| = |x R_n(-x^2)|
\]
The left term in the sum is already known. The error term is therefore \( x R_n(-x^2) \), which we can bound over \([-2, 2]\). Indeed, put \( t = -x^2 \). Since \( x \) varies between -2 and 2, \( t \) varies between -4 and 0. We apply Lagrange’s theorem for the function \( g(t) = e^t \). There exists \( c \) between 0 and \( t \) such that
\[
R_n(t) = \frac{g^{(n+1)}(c)}{(n+1)!} t^n = \frac{e^c}{(n+1)!} t^n.
\]
Then
\[
|R_n(t)| \leq \frac{e^0}{(n+1)!} |t|^n \leq \frac{4^n}{(n+1)!}.
\]
Therefore, the error term is estimated as follows:
\[
|E_{2n+1}(x)| = |x R_n(-x^2)| = |x||R_n(-x^2)| \leq \frac{2 \cdot 4^n}{(n+1)!}.
\]
c) Find $n$ so as to have $2n + 1$th degree Taylor approximation with error of at most $10^{-9}$ on $[-2, 2]$.

**Solution**

To make sure that the size of error term $E_{2n+1}(x)$ is under $\epsilon = 10^{-9}$, we only need to find $n$ such that

$$\frac{2 \cdot 4^n}{(n+1)!} < \epsilon.$$ 

And we find that $n = 23$ is the smallest $n$ for this inequality to be satisfied.

**Problem 2.**

Convert the number $(101.011)_2$ from binary to base 10.

**Solution**

$$(101.011)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 5.375$$

**Problem 3.**

Convert the number $3.7$ from decimal to binary system.

**Solution**

$$3.7 = 2 \times 2^1 + 1 \times 2^0 + 0.7$$

We need a base 2 expansion for 0.7. Note that $0.7 \times 2 = 1.4$, so we record a 1. Then $0.4 \times 2 = 0.8$, so we record a 0. Then $0.8 \times 2 = 1.6$, so we record a 1. Then $0.6 \times 2 = 1.2$, so we record a 1. Then $0.2 \times 2 = 0.4$, so we record a 0. And finally $0.4 \times 2 = 0.8$, the second term in the sequence. This describes a repeating base 2 expansion. Thus

$$3.7 = 11.101\overline{10}_2$$

**Problem 4.**

Do the following operations

a) $(1.001)_2 \times 2^2 + (1.101)_2 \times 2^4$

b) $(1.001)_2 \times 2^1 - (1.101)_2 \times 2^3$

c) $(1.001)_2 \times 2^7 + (1.101)_2 \times 2^7$

d) $(1.001)_2 \times 2^6 + (1.100)_2 \times 2^{-2}$

Write your results in both floating-point and decimal format. Make sure to show all your calculations, not just the final result. What do you notice when adding these two numbers of quite different size?

**Solution**

One needs to make sure that the result of each operation stays in the given floating-point format.
a) 

\[(1.001)_2 \times 2^2 + (1.101)_2 \times 2^4 = (0.01001)_2 \times 2^4 + (1.101)_2 \times 2^4 \quad \text{(matching exponents)}
\]

\[= (1.111001)_2 \times 2^4 \quad \text{(summing)}
\]

\[\approx (1.111)_2 \times 2^4 \quad \text{(rounding)}
\]

And \((1.111)_2 \times 2^4 = 30_{10}\).

b) 

\[(1.001)_2 \times 2^1 - (1.101)_2 \times 2^3 = (0.01001)_2 \times 2^3 - (1.101)_2 \times 2^3 \quad \text{(matching exponents)}
\]

\[= -((1.101)_2 - (0.01001)_2) \times 2^3 \quad \text{(subtracting)}
\]

\[= (1.01011)_2 \times 2^3 \quad \text{(subtracting)}
\]

\[\approx (1.011)_2 \times 2^3 \quad \text{(rounding)}
\]

c) 

\[(1.001)_2 \times 2^7 + (1.101)_2 \times 2^7 = ((1.001)_2 + (1.101)_2) \times 10^7 = (1.011)_2 \times 2^8 \approx \infty
\]

because \(e = 8\) corresponds to \(E = 15\).

d) 

\[(1.001)_2 \times 2^6 + (1.101)_2 \times 2^{-2} = (1.001000011)_2 \times 2^6 \approx (1.001)_2 \times 2^6
\]

Adding two numbers of too different sizes causes the smaller number to be completely ignored. This results in arithmetic error \(x + y = x\) when \(x >> y\).

**Problem 5.**

What number does the bit sequence 1 0 0 1 1 0 1 1 represent?

**Solution**

Note: See worksheet 10/7/19 for the structure of the 8 bit sequence.

- The number in the first position is 1, therefor the sign is negative.

- The mantissa is 1.0112 (1.a1a2a3)

- The exponent is 00112 - 7 = 3 - 7 = -4

We can then compute the value of the bit sequence (denoted \(x\)) as

\[x = -1.0112 \times 2^{-4} = -0.00010112 = -0.0859375
\]

**Problem 6.**

What is the smallest number greater than 1 that can be represented by floating-point format? Call this number \(b\). The difference \(\epsilon = b - 1\) is called the *machine epsilon* of this number format. Find \(\epsilon\).
Solution

We have 7 digits to allocate, 4 to describe the exponent and 3 to describe the mantissa.

\[ b = 1.001 \times 2^0 \]

is the smallest number greater than 1 accessible with 3 digits that we can store in the mantissa. \( b \) can be represented with the bit sequence 0 0111 001 (spaces added for emphasis). Then

\[ b - 1 = 1.001_2 - 1_2 = 0.001_2 = 2^{-3} = \frac{1}{2^3} = \frac{1}{8} = 0.125_{10} \]

Thus the machine epsilon of this floating point format is 0.125 base 10, or 0.001_2.