Practice problems for chapter 7

1. **Exercise 7.1.** We can rewrite the formula as

\[
\frac{1 - \cos x}{\sin x} = \frac{(1 - \cos x)(1 + \cos x)}{\sin x (1 + \cos x)} = \frac{\sin x}{1 + \cos x}.
\]

Evaluating this expression yields

```matlab
>> format long e
>> chop(sin(1e-2),4)/(1+chop(cos(1e-2),4))
ans =
     5.000000000000000e-003
```

which is much more accurate, if we compare with the result in the full Matlab precision

```matlab
>> format long e
>> (sin(1e-2))/(1+cos(1e-2))
ans =
     5.000041667083338-003
```

2. **Exercise 7.3.** If you display the intermediate results in the first loop, you’ll notice that the variable `sum` reaches the value 1.6240 at \( k = 44 \), and remains constant after that. The reason is simple: \( \frac{1}{45^2} = 4.938 \cdot 10^{-4} \), so

\[
1.6240 + 4.938 \cdot 10^{-4} = 1.62449\ldots,
\]

and rounding to four significant digits yields 1.6240.

The second implementation is much more accurate, because we add the smallest terms \( 1/k^2 \) first, while the sum is still small, and the largest terms are added at the end of the iteration.

3. **Exercise 7.4.** Matlab returns the following numbers

(a) 0
(b) \( 1.1102 \cdot 10^{-16} \)
(c) \( -1.1102 \cdot 10^{-16} \)
(d) 0
(e) \( -2.2204 \cdot 10^{-16} \)
To explain the first three values, we have to determine the floating-point numbers closest to 1. The representation of 1 as a double precision floating-point number is

\[
1 = (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + \cdots + 0 \cdot 2^{-n}) 2^1
\]

where \( n = 53 \). The smallest floating-point number greater than 1 is

\[
(.10 \cdots 01)_2 2^1 = (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + \cdots + 0 \cdot 2^{-n-1} + 1 \cdot 2^{-n}) 2^1
\]

\[
= 1 + 2^{-n+1}
\]

\[
= 1 + 2.2204 \cdot 10^{-16}
\]

\[
= 1 + 2\epsilon_M.
\]

The largest floating-point number less than 1 is

\[
(.11 \cdots 11)_2 2^0 = (1 \cdot 2^{-1} + 1 \cdot 2^{-2} + \cdots + 1 \cdot 2^{-n-1} + 1 \cdot 2^{-n}) 2^0
\]

\[
= 1 - 2^{-n}
\]

\[
= 1 - 1.1102 \cdot 10^{-16}
\]

\[
= 1 - \epsilon_M.
\]

This is summarized in the figure below (with \( \epsilon_M = 2^{-53} \approx 1.11 \cdot 10^{-16} \)).

<table>
<thead>
<tr>
<th></th>
<th>1 - 3\epsilon_M</th>
<th>1 - 2\epsilon_M</th>
<th>1 - \epsilon_M</th>
<th>1</th>
<th>1 + 2\epsilon_M</th>
<th>1 + 4\epsilon_M</th>
</tr>
</thead>
</table>

The situation around the number −1 is symmetric: the smallest floating-point number greater than −1 is \(-1 + \epsilon_M\); the largest floating-point number less than −1 is \(-1 - 2\epsilon_M\).

It is now easy to explain the first three results.

(a) \( 1 + 10^{-16} \) lies between 1 and \( 1 + \epsilon_M \), so it is rounded to 1, and subtracting 1 yields zero.

(b) \( 10^{-16} - 1 \) lies between \(-1 + \epsilon_M/2\) and \(-1 + \epsilon_M\), so it is rounded to \(-1 + \epsilon_M\), and adding 1 yields \( \epsilon_M \).

(c) \( 1 - 10^{-16} \) lies between \( 1 - \epsilon_M \) and \( 1 - \epsilon_M/2 \), so it is rounded to \( 1 - \epsilon_M \), and subtracting 1 yields \( -\epsilon_M \).
To explain the next four values, we have to determine the floating-point numbers closest to 2. The representation of the number 2 as a double precision floating-point number is

\[ 2 = (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + \cdots + 0 \cdot 2^{-n}) 2^2 \]

\[ = (10 \cdots 00)_2 2^2 \]

where \( n = 53 \). The smallest floating-point number greater than 2 is

\[ (10 \cdots 01)_2 2^2 = (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + \cdots + 0 \cdot 2^{-n-1} + 1 \cdot 2^{-n}) 2^2 \]

\[ = 2 + 2^{-n+2} \]

\[ = 2 + 4.409 \cdot 10^{-16} \]

\[ = 2 + 4\epsilon_M. \]

The largest floating-point number less than 2 is

\[ (11 \cdots 11)_2 2^1 = (1 \cdot 2^{-1} + 1 \cdot 2^{-2} + \cdots + 1 \cdot 2^{-n-1} + 1 \cdot 2^{-n}) 2^1 \]

\[ = 2 - 2^{-n+1} \]

\[ = 2 - 2.2024 \cdot 10^{-16} \]

\[ = 1 - 2\epsilon_M. \]

This is summarized below.

\[
\begin{array}{cccccc}
2 - 6\epsilon_M & 2 - 4\epsilon_M & 2 - 2\epsilon_M & 2 & 2 + 4\epsilon_M & 2 + 8\epsilon_M \\
\end{array}
\]

With this in mind we can explain the answers.

(d) \( 2 + 2 \cdot 10^{-16} \) lies between 2 and \( 2 + 2\epsilon_M \), so it is rounded to 2, and subtracting 2 yields zero.

(e) \( 2 - 2 \cdot 10^{-16} \) lies between \( 2 - 2\epsilon_M \) and \( 2 - \epsilon_M \), so it is rounded to \( 2 - 2\epsilon_M \), and subtracting 2 yields \(-2\epsilon_M\).

(f) \( 2 + 3 \cdot 10^{-16} \) lies between \( 2 + 2\epsilon_M \) and \( 2 + 4\epsilon_M \), so it is rounded to \( 2 + 4\epsilon_M \), and subtracting 2 yields \( 4\epsilon_M \).

(g) \( 2 - 3 \cdot 10^{-16} \) lies between \( 2 - 3\epsilon_M \) and \( 2 - 2\epsilon_M \), so it is rounded to \( 2 - 2\epsilon_M \), and subtracting 2 yields \(-2\epsilon_M\).

4. Exercise 7.6. The final value is \( x = 1 \).

Using the hint we can say that after one pass through the first for-loop we have \( 1 < x < 1 + 1/2 \). After the second pass, \( 1 < x < 1 + 1/4 \). After \( k \) passes, \( 1 < x < 1 + 1/2^k \), and after finishing the for-loop we have

\[ 1 < x < 1 + 2^{-54}. \]
This means \( x \) lies between 1 and \( 1 + \epsilon_M \). (Recall that \( \epsilon_M = 2^{-53} \).) Therefore we can expect that in double-precision arithmetic, the value after the first for-loop will be \( x = 1 \), and squaring 54 times still yields \( x = 1 \).

5. **Exercise 7.9.** Matlab returns the following values:

- for \( n = 10^{-4} \): \((1 + 1/n)^n = 2.718145926 \) (five correct significant digits)
- for \( n = 10^{-8} \): \((1 + 1/n)^n = 2.718281798 \) (eight correct significant digits)
- for \( n = 10^{-12} \): \((1 + 1/n)^n = 2.718523496 \) (four correct significant digits)
- for \( n = 10^{-16} \): \((1 + 1/n)^n = 1 \) (zero correct significant digits)

The last result is easiest to explain. \( 1 + 1/n = 1 + 10^{16} \) is rounded to one because \( 10^{-16} \) is less than the machine precision.

For the other values we observe that the accuracy first improves as expected, and then gets worse for very large values of \( n \). We can explain this as follows.

When adding \( 1/n \) to 1, we make a very small roundoff error \( \Delta \) (of the order of \( 10^{-16} \)), so in the next step we really calculate \((1 + 1/n + \Delta)^n\) instead of \((1 + 1/n)^n\). Taking the Taylor series expansion of \((1 + 1/n + \Delta)^n\) around \( \Delta = 0 \), we obtain

\[
(1 + 1/n + \Delta)^n \approx (1 + 1/n)^n + n(1 + 1/n)^{n-1}\Delta \\
= (1 + 1/n)^n(1 + \frac{n}{1 + 1/n}\Delta) \\
\approx (1 + 1/n)^n(1 + n\Delta).
\]

We see that the relative error in the result is

\[
\frac{|(1 + 1/n + \Delta)^n - (1 + 1/n)^n|}{(1 + 1/n)^n} \approx |n\Delta|,
\]

i.e., the small error \( \Delta \) leads to a very large relative error \( |n\Delta| \) in the final result. Note that this error is not caused by cancellation.

6. **Exercise 7.11.**

(a) The first discontinuity is at \( x = \sqrt{\epsilon_M} \).

The number \( 1 - x^2/2 \) is indistinguishable from 1 if

\[
1 - \frac{\epsilon_M}{2} < 1 - \frac{x^2}{2} \leq 1,
\]

or \( 0 \leq x < \sqrt{\epsilon_M} \).
(b) The second discontinuity is at \( x = \sqrt{3\epsilon_M} \). If

\[
1 - \frac{3\epsilon_M}{2} < 1 - \frac{x^2}{2} < 1 - \frac{\epsilon_M}{2}
\]

then \( 1 - x^2/2 \) is rounded to \( 1 - \epsilon_M \). If

\[
1 - \frac{5\epsilon_M}{2} < 1 - \frac{x^2}{2} < 1 - \frac{3\epsilon_M}{2}
\]

then \( 1 - x^2/2 \) is rounded to \( 1 - 2\epsilon_M \). The boundary is at \( x = \sqrt{3\epsilon_M} \).

(c) The left limit is

\[
\frac{1 - (1 - \epsilon_M)}{3\epsilon_M} = \frac{1}{3}.
\]

The right limit is

\[
\frac{1 - (1 - 2\epsilon_M)}{3\epsilon_M} = \frac{2}{3}.
\]

(d) Multiply and divide by \( 1 + \cos x \) gives a formula that avoids cancellation for \( x \) around 0:

\[
f(x) = \frac{(1 - \cos x)(1 + \cos x)}{x^2(1 + \cos x)} = \frac{\sin(x)^2}{x^2(1 + \cos x)}.
\]