Cancellation Errors

When one adds two numbers with opposite sign but with similar absolute values, the result may be quite inexact and the situation is referred to as *loss*, or *cancellation, of significant digits*.
We encounter this type of error in polynomial evaluation:

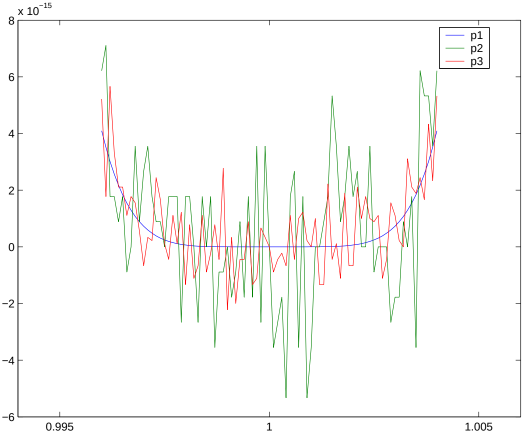
$$\begin{aligned} p(x) &= (x-1)^6 \\ &= x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1 \\ &= 1 + x(-6 + x(15 + x(-20 + x(15 + x(-6 + x))))) \end{aligned}$$

These 3 forms are equivalent at the theoretical level, but they yield very different results when they are implemented.

Let us evaluate the polynomial $p(x)$ by using the three different forms at 81 equispaced points in $[0.996, 1.004]$:

```
x=linspace(0.996,1.004,81);
p1=(x-1).^6;
p2=x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1;
a=[1,-6,15,-20,15,-6,1];
p3=horner2(a,x); % download horner2.m  from
                 % dm.ing.unibs.it/gervasio/Nummeth/matlab
plot(x,p1,x,p2,x,p3)
legend('first form','second form','third form')
```

Notice that the most correct form is the first one. The graphs associated to $p_2(x)$ and $p_3(x)$ exhibit a lot of oscillations due to cancellation errors.



x 10^−15
8
6
4
2
0
−2
−4
−6
0.995
1
1.005
p1
p2
p3

Rounding errors propagation

Exercise 1: Write a matlab function to compute the following sequence:

$$a_0 = \frac{1}{e}(e-1),$$
$$a_{n+1} = 1 - (n+1)a_n, \quad \text{for } n = 0, 1, \ldots, N,$$

where N is assigned.
Compare the numerical result with the exact limit $a_n \to 0$ for $n \to \infty$.

Solution.

The function has one input: N
and one output: the vector `an` holding the values $\{a_0, a_1, \dots, a_N\}$.
It is possible to prove that a_n is a positive, decreasing and infitesimal sequence.

Remark Indices of arrays must be positive integers in matlab, thus
a_0 is stored in `an(1)`
a_1 is stored in `an(2)`, and so on.

The sequence computed numerically is oscillating with diverging absolute values. Why?

At each step we introduce a small rounding error $\leq \mathtt{u} \simeq 1.1102e-16$, that is amplified by the recursive formula.
When $n = 17$, the amplified error is $\simeq 1$, when $n = 20$ the amplified error is $\simeq 30$, and so on.

Rounding errors propagation

Exercise 2: Write a matlab function to compute the following sequence:

$$z_2 = 2, \quad z_{n+1} = 2^{n-1/2}\sqrt{1-\sqrt{1-4^{1-n}z_n^2}}, \qquad n = 2, 3, \ldots. \tag{1}$$

Compare the numerical result with the exact limit $z_n \longrightarrow \pi$ for $n \longrightarrow \infty$.

Solution The function has one input: N
and one output: the vector $\mathbf{z}$ holding the values $\{z_2, z_3, \ldots, z_N\}$.
The anomalous behavior of the computed sequence is due to the propagation of roundoff errors from the innermost operation. In particular, when $4^{1-n}z_n^2$ is less than $\epsilon_M/2$, the subsequent element z_{n+1} of the sequence is equal to 0. This happens for $n \geq 30$.