1. Tutorial

Exercise 1 (Summation)**.** Let $p_i \in \mathbb{F}, 1 \le i \le n$ be a sequence of n floating-point numbers.

1. Show that the condition number of the computation of the summation satisfies

$$\operatorname{cond}(\sum_{i=1}^{n} p_i) = \frac{\sum_{i=1}^{n} |p_i|}{|\sum_{i=1}^{n} p_i|}.$$

We recall that by definition

$$\operatorname{cond}(\sum_{i=1}^{n} p_i) := \lim_{\varepsilon \to 0} \sup \left\{ \left| \frac{\sum_{i=1}^{n} \widetilde{p}_i - \sum_{i=1}^{n} p_i}{\varepsilon \sum_{i=1}^{n} p_i} \right| : |\widetilde{p}_i - p_i| \le \varepsilon |p_i| \text{ for } i = 1, \dots, n \right\}.$$

2. Show that the recursive summation algorithm is *backward-stable*.
3. Derive a bound on the relative error for the summation.
4. Redo all the questions for the dot product.

2. Practical

Exercise 2 (Summation algorithms)**.** The purpose is to compare the accuracy of different algorithms for summation.

1. Implement the Error-Free Transformations (EFT).

Algorithm 1. EFT for the summation of two floating-point numbers with $|a| \ge |b|$

function $[x, y] = \texttt{FastTwoSum}(a, b)$
  $x = \mathrm{fl}(a + b)$
  $y = \mathrm{fl}((a - x) + b)$

Algorithm 2. EFT for the summation of two floating-point numbers

function $[x, y] = \texttt{TwoSum}(a, b)$
  $x = \mathrm{fl}(a + b)$
  $z = \mathrm{fl}(x - a)$
  $y = \mathrm{fl}((a - (x - z)) + (b - z))$

2. Implement the following summation algorithms:

Algorithm 3. Classic recursive summation algorithm

function $\texttt{res} = \texttt{Sum}(p)$
  $\sigma = 0;$
  for $i = 1 : n$
    $\sigma = \mathrm{fl}(\sigma + p_i)$
  $\texttt{res} = \sigma$

Algorithm 4. Kahan's summation algorithm

function `res` = `SCompSum`(p)
  $\sigma = 0$
  $e = 0$
  for $i = 1 : n$
    $y = p_i + e$
    $[\sigma, e]$ = `FastTwoSum`(σ, y)
  `res` = σ

Algorithm 5. Priest's doubly compensated summation algorithm

function `res` = `DCompSum`(p)
  we sort the p_i such that $|p_1| \geq |p_2| \geq \cdots \geq |p_n|$
  $s = 0$
  $c = 0$
  for $i = 1 : n$
    $[y, u]$ = `FastTwoSum`(c, p_i)
    $[t, v]$ = `FastTwoSum`(y, s)
    $z = u + v$
    $[s, c]$ = `FastTwoSum`(t, z)
  `res` = s

Algorithm 6. Rump's compensated summation algorithm

function `res` = `CompSum`(p)
  $\pi_1 = p_1$; $\sigma_1 = 0$;
  for $i = 2 : n$
    $[\pi_i, q_i]$ = `TwoSum`(π_{i-1}, p_i)
    $\sigma_i = \mathrm{fl}(\sigma_{i-1} + q_i)$
  `res` = $\mathrm{fl}(\pi_n + \sigma_n)$

3. Study the accuracy of those different algorithms in function of the condition number of the sum[1].

[1]A MATLAB generator of ill-conditioned sum can be found here:
`http://www-pequan.lip6.fr/~graillat/gensum.zip`