

## Algorithmes et calcul scientifique

Examen de mai 2020

*Durée : 1 heure. Tous moyens autorisés sous contrainte de travail personnel.*

**Votre rendu DOIT ETRE un PDF** obtenu à partir d'un document au format de votre choix : openoffice,  $\text{\LaTeX}$ , md, texte, manuscrit, ... Aucun document autre qu'un pdf ne sera pris en considération pour cette épreuve.

Barème indicatif sur 20 : Ex. 1 : 16 (1+4+11); Ex. 2 : 4 (1+1+1+1); qualité de la rédaction =  $\pm 1$ .

### Exercice 1.

Dans cet exercice, on considère le format `binary32` des flottants binaires de l'IEEE-754.

1. Rappeler brièvement les différents modes d'arrondi de l'arithmétique IEEE-754.
2. Soit  $x = (0.1)_{10}$ . On rappelle que  $(0.1)_{10} = (0.0\ 0011\ 0011\ 0011\ \dots)_2$ .
  - (a) Déterminer les valeurs entières décimale et binaire de l'exposant de la représentation de  $x$  dans le format considéré.
  - (b) Déterminer les mantisses binaires de la représentation de  $x$  selon les différents modes d'arrondi de l'arithmétique IEEE-754.
  - (c) En déduire les représentations complètes de  $x$  dans le format et pour les arrondis considérés.
3. On note  $\epsilon$  la précision du format `binary32` et on considère les sept valeurs suivantes :
  - $x_1 = 4 + 2^{-21}$
  - $x_2 = 4 + 2^{-22}$
  - $x_3 = 4 + 2^{-23}$
  - $y_1 = 4 - 2^{-21}$
  - $y_2 = 4 - 2^{-22}$
  - $y_3 = 4 - 2^{-23}$
  - $y_4 = 4 - 2^{-24}$ .

**En utilisant chaque fois que possible la notation  $\epsilon$**  (et ainsi ne plus avoir aucune puissance "compliquée"), explicitez :

- (a) les différents arrondis de chacune de ces 7 valeurs et
- (b) les erreurs absolues d'arrondi associées.

Vous présenterez ces résultats avec un tableau de forme pertinente.

### Exercice 2.

Toujours en `binary32` et en arrondi au plus près, on considère l'opération  $x + y$  avec  $x = 1.0$  et  $y = (1.000\dots 01)_2 \times 2^{-24}$  —  $y$  a 22 bits nuls entre le point et le 1 du bit final.

1. Quel est le résultat correctement arrondi ?
2. Quel est le résultat calculé avec un seul bit de garde ?
3. Quel est le résultat calculé avec deux bits de garde, *i.e.* un bit de garde et un bit d'arrondi ?
4. Quel est le résultat calculé avec deux bits de garde et un *sticky* bit ?