Corrigé
Représentations et Méthodes Numériques
13 janvier 2020

Durée : 2h.
SANS DOCUMENT.
Tout objet connecté doit être éteint et rangé.
Les vraies calculatrices non connectées sont autorisées.

Exercice 1 – Représentations

1. (**1 point**) Quelle est la représentation en tant qu'entier signé sur 16 bits de 588 ?

Solution :

$$588 = 0000001001001100$$

2. (**1 point**) Quelle est la représentation en tant qu'entier signé sur 16 bits de -588 ?

Solution :

$$-588 = 1111110110110100$$

3. (**1 point**) Quel est la valeur en base 10 de 1111111111110010 en tant qu'entier signés codés sur 16 bits ?

Solution :

$$1111111111110010 = -1110 = -14$$

4. (**2 points**) Calculer le $pgcd$ et les coefficients de Bézout de $a = 264$ et $b = 198$.

Solution :

$$u = 1, v = -1, pgcd = 66$$

5. (**2 points**) Quel est le codage de $13,7$ en simple précision avec un arrondi vers $+\infty$?

Solution :

$$13,7 = 0\ 10000010\ 10110110011001100110100$$

Exercice 2 – Algèbre linéaire

1. (**2 points**) La fonction ci-dessous calcule la valeur d'un polynôme de degré n d'un réel x en double précision par le scéma de Hornër. les coefficients du polynôme sont stockés dans le tableau a.

```
double horner(double *a, double x, int n)
{
  double y;
  int i;
  y = a[n];
  for(i=n-1; i>=0; i--)
    y = y*x + a[i];
  return y;
}
```

Réécrire cette fonction sans utiliser la variable entière i.

Solution :

```
double horner(double *a, double x, int n)
{
  double y, *pa;
  y = *(a+n);
  for(pa = a + n-1; pa>=a; pa--)
    y = y*x + *pa;
  return y;
}
```

2. (**3 points**) Soit $A = \begin{pmatrix} -6 & 1 & -7 \\ 9 & 3 & 5 \\ 2 & 0 & 2 \end{pmatrix}$ et $B = \begin{pmatrix} -14 \\ 11 \\ 4 \end{pmatrix}$
   - Quelle est la solution exacte du système ?
   - Résoudre le système $A.X = B$ par la méthode de Gauss avec recherche partielle de pivot maximum en simulant un ordinateur avec une arthmétique virgule flottante en base 10 avec 3 chiffres de mantisse avec arrondi vers zéro.

Solution :

$$X_{sol} = (1, -1, 1)$$

$$AB1 = \begin{pmatrix} 1 & 0.333 & 0.555 & 1.22 \\ 0 & 2.99 & -3.67 & -6.68 \\ 0 & -0.666 & 0.89 & 1.56 \end{pmatrix}$$

$$AB2 = \begin{pmatrix} 1 & 0.333 & 0.555 & 1.22 \\ 0 & 1 & -1.22 & -2.23 \\ 0 & 0 & 0.0774 & 0.08 \end{pmatrix}$$

$$x_{sol} = (1.03, -0.98, 0.969)$$

3. (**2 point**) Calculer la décomposition LU de

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

Solution :

$$L = \begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & 3/2 & 0 & 0 \\ 0 & -1 & 4/3 & 0 \\ 0 & 0 & -1 & 5/4 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & -1/2 & 0 & 0 \\ 0 & 1 & -2/3 & 0 \\ 0 & 0 & 1 & -3/4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Exercice 3 – Etude de suites

1. (**2 points**) Sachant que les erreurs d'arrondi sont négligeables, un calcul des éléments d'une suite sur ordinateur donne

$$\begin{aligned} x_{10} &= 2.00933613521090 \\ x_{13} &= 2.00276928362265 \\ x_{16} &= 2.00082079478201 \\ x_{19} &= 2.00024322186046 \end{aligned}$$

Dévrivez avec le plus de précision possible la vitesse de convergence.

Solution :
La convergence est linéaire en $\left(\frac{2}{3}\right)^n$

2. (**4 points**) Soit le suite récurrente d'ordre 1

$$x_{n+1} = \phi(x) = \frac{x_n^3 + x_n}{2.x_n^2 + x_n + 1}$$

— Déterminer les points fixes de la suite.
— Déterminer les limites possibles de la suite.
— Préciser la vitesse de convergence pour chaque limite possible.

Solution :
Les points fixes sont 0 et -1.
-1 est limite possible avec une convergence linéaire en $\left(\frac{1}{2}\right)^n$

3. (**4 points**) On s'intéresse à l'équation $f(x) = \frac{1}{x^2} - y = 0$ avec $y > 0$.
— Quelle est l'itération de Newton correspondante ?
— Quelles sont les limites possibles de l'itération de Newton ?
— Si l'on suppose que le point initial a 4 bits de précision avec la limite que l'on cherche à approcher, combien d'itérations sont nécessaires pour obtenir 53 bits de précision ?

Solution :
$$x_{n+1} = \frac{x_n.(3 - x_n^2.y)}{2}$$
Les limites possibles sont $\pm\sqrt{1/y}$.
La convergence de la méthode de Newton est quadratique au voisinage d'une racine simple ce qui est le cas des deux limites possibles.
Donc à partir d'une approximation à 4 bits de précision, après 1 itération on a 8 bits de précision, après 2 itérations on a 16 bits de précision, après 3 itérations on a 32 bits de précision et après 4 itérations on a 64 bits de précision.