TP 3 : Représentation des nombres en machine.

Exercice 1 : représentation des entiers naturels

Toutes les questions se traitent sur papier. Vérifier toutefois chacune des réponses grâce aux fonctions Scilab *dec2bin* et *bin2dec*.

a) Convertir en binaire les entiers naturels $(5)_{10}$, $(48)_{10}$ et $(113)_{10}$.

b) Convertir en hexadécimal les entiers naturels $(49)_{10}$ et $(161)_{10}$.

c) Convertir en décimal le nombre $(B5F)_{16}$.

d) Quel est le plus grand entier que l'on puisse coder sur 8 bits ? sur 32 bits ?

e) On suppose que les entiers sont codés sur 8 bits. Effectuer les opérations suivantes :

$$(01\ 000\ 011)_2 + (01\ 111\ 111)_2$$
$$(11\ 000\ 000)_2 + (01\ 000\ 000)_2$$

Vérifier le résultat en le reconvertissant en base 10.

f) Faire la multiplication en binaire de $(11)_{10}$ et $(7)_{10}$. Vérifier le résultat en le reconvertissant en base 10.

Exercice 2 : représentation des réels

a) Convertir en binaire le réel $(5.65)_{10}$. Vérifier le calcul en reconvertissant en base 10 le résultat obtenu.

b) Quel est le plus grand réel représentable en machine au format double ? Vérifier en affichant la valeur de la variable `realmax=number_properties("huge")`. Taper (`realmax`+0.001)-`realmax`.

c) Quel est le plus petit réel strictement positif représentable en machine au format double ? Vérifier en affichant la valeur de la variable `realmin=number_properties("tiny")`.

d) Quel est le plus petit double ϵ tel que $1 + \epsilon$ soit différent de 1 ? Vérifier en affichant la valeur de la variable `%eps` puis en tapant $(1 + \epsilon/2) - 1$.

Exercice 3 : mise en évidence des erreurs d'arrondi

a) Vous savez que l'addition est commutative. Tapez successivement :

```
>> format(20)
>> 4+0.0001-4
>> 4-4+0.0001
```

Qu'observez-vous ? Encore plus étonnant :

```
>> format(20)
>> x=1.0E+29;
>> y=1.0E-9;
>> z=((y+x)-x)/y
>> z=(y+(x-x))/y
```

Comment expliquez-vous ce que vous avez observé ?

b) A présent nous considérons le programme suivant :

```
n = 10000;
s = 1/3;
for i=1:n-1
  s = s+1/3;
end
disp(s-n/3)
```

Que fait ce programme ? Que devrait afficher l'instruction "disp(s-n/3)" ? Saisissez ce programme dans un fichier "tiers.sce" et observez le résultat. Qu'en pensez vous ?

Exercice 4 : mise en évidence des erreurs d'arrondi

Soit P_n le demi-périmètre du polygone régulier à n côtés inscrit dans un cercle de rayon 1.

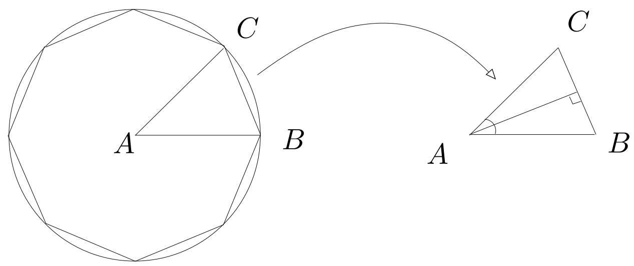

FIGURE 1 – Figure pour l'exercice 4

a) Donnez l'expression exacte de P_n.

b) Faites un développement limité de P_n pour n tendant vers l'infini.

c) On introduit x_k la suite définie par

$$x_k = P_{2^k} = 2^k \sin(\frac{\pi}{2^k}).$$

Utilisez la relation trigonométrique $\sin^2(\alpha/2) = (1 - \cos(\alpha))/2$ pour montrer que $(x_k)_k$ est définie par la récurrence suivante :

$$\begin{cases} x_1 = 2 \\ x_{k+1} = 2^k \sqrt{2(1 - \sqrt{1 - (x_k/2^k)^2})} \end{cases}$$

Quelle est la limite de cette suite ?

d) Faites un programme Scilab qui calcule les M premiers éléments de cette suite et qui les affiche sur un graphique. Le résultat correspond-il à la limite trouvée en **c)** ?

e) Nous souhaitons éviter cette difficulté. Si nous utilisons la formule :

$$\sin(\frac{\alpha}{2}) = \frac{\sin \alpha}{\sqrt{2(1 + \sqrt{1 - \sin^2 \alpha})}},$$

comment s'écrit la relation de récurrence entre x_k et x_{k+1} ?

Programmez cette suite. Converge-t-elle vers le résultat escompté ? Expliquez la différence de comportement entre les suites programmées en **c)** et en **d)**.