Séquence : 01
Document : DS01
Lycée Dorian

Juliette Genzmer
Willie Robert
Renaud Costadoat

Avec Correction

DS01 Informatique

Référence	S01- DS01
Compétences	Alg-C6 : Justifier qu'une itération (ou boucle) produit l'effet attendu au moyen d'un invariant Alg-C7 : démontrer qu'une boucle se termine effectivement Déc-C1: Manipuler en mode utilisateur les principales fonctions d'un système d'exploitation et d'un environnement de développement Déc-C2: Appréhender les limitations intrinsèques à la manipulation informatique des nombres Déc-C3: Initier un sens critique au sujet de la qualité et de la précision des résultats de calculs numériques sur ordinateur
Description	Fait le 03/10/2020

1 Introduction

Question 1 Écrire sur le diagramme de Contexte donné en document réponse le nom des composants de l'unité centrale.

2 Analyse d'une réponse temporelle

Le tracé de la figure 1 correspond à la tension $v(t) = V_{r\ max} \cdot sin(k \cdot \theta(t)) \cdot cos(\theta(t))$ (avec $\theta(t) = 2 \cdot \pi \cdot f_r \cdot t$) induite dans l'un des enroulements fixes des deux secondaires du moteur du bras du système de pulvérisation de nacre (vu en DS de SI).

On sait que $f_r = 0,6\ \mathrm{Hz}$, mais l'objectif va être de déterminer grâce à python les valeurs des **entiers** $V_{r\ max}$ et k.

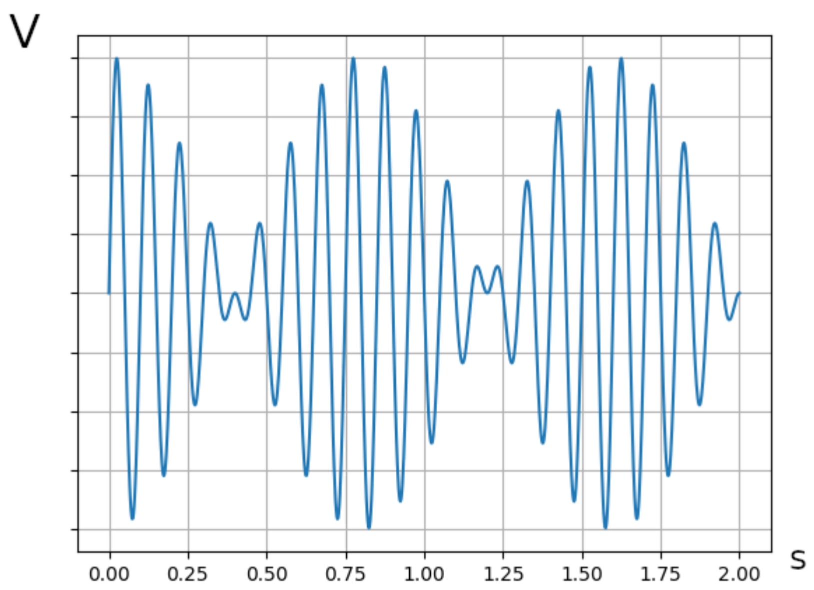


FIGURE 1 – Tracé de la réponse temporelle v(t)

Cette fonction est définie comme suit dans un script python qui va être complété :

```
def theta(t):
    return 2*np.pi*fr*t

def v(t):
    return Vr*np.sin(k*theta(t))*np.cos(theta(t))

t=np.linspace(0,2,1000)
plt.plot(t,v(t))
```

2.1 Recherche de $V_{r\ max}$

On propose deux scripts pour compléter le précédent afin de déterminer la valeur de $V_{r\ max}$.

Solution A

```python
m=v(t[i])
while v(t[i])<=m:
    if v(t[i])>m:
        m=v(t[i])
print(m)
```

Solution B

```python
m=v(0)
for ti in t:
    if v(ti)>m:
        m=v(ti)
print(m)
```

Question 2 Choisir en justifiant la solution qui permet de déterminer $V_{r\ max}$.
Le résultat affiché par le script qui convient est `9.9519`.

Question 3 En déduire en justifiant la valeur de $V_{r\ max}$.

2.2 Identification de k

On montre que la courbe de la figure 1, coupe $2 \cdot k$ fois la droite d'équation $y = 1$ sur l'intervalle $\left[0, \frac{1}{f_r}\right]$. L'objectif de la suite est de déterminer le nombre d'intersections afin d'en déduire k.

Recherche des intervalles $[t, t+dt]$, incluant un passage par y=1

On souhaite dans cette partie créer une liste `bornes`, contenant l'ensemble des intervalles $[t, t+dt]$ tels qu'il existe un $t_p \in [t, t+dt]$ tel que $v(t_p) = 1$.

Une fois la liste créée, en tapant `print(bornes[0:last])`, on obtient le résultat suivant :

```
[[0.0, 0.002002002002002002], [0.050050050050050046, 0.052052052052052055],
[0.1041041041041041, 0.1061061061061061], [0.15415415415415415, 0.15615615615615616]]
```

Cela signifie que la courbe $v(t)$ coupe $y = 1$ entre 0 et 0.002002002002002002, etc...

Question 4 Quelle valeur de `last` permet l'affichage précédent ?

Question 5 Écrire un script python permettant de détecter puis d'écrire dans la liste `bornes` l'ensemble des intervalles définis précédemment.

Recherche des solutions par dichotomie

Voici le principe de la dichotomie :

- Au rang 0,
  - soient $a_0 = a$, $b_0 = b$. Il existe une solution x_0 de l'équation $(f(x) = 0)$ dans l'intervalle $[a_0, b_0]$.
- Au rang 1,
  - si $f(a_0).f(\dfrac{a_0+b_0}{2}) \leq 0$, alors on pose $a_1 = a_0$, $b_1 = \dfrac{a_0+b_0}{2}$,

  - sinon on pose $a_1 = \dfrac{a_0 + b_0}{2}$ et $b_1 = b$.
  - dans les deux cas, il existe une solution x_1 de l'équation $(f(x) = 0)$ dans l'intervalle $[a_1, b_1]$.
- Au rang n, supposons construit un intervalle $[a_n, b_n]$, de longueur $\dfrac{b-a}{2^n}$, et contenant une solution x_n de l'équation $(f(x) = 0)$. Alors :
  - si $f(a_n).f(\dfrac{a_n + b_n}{2}) \leq 0$, alors on pose $a_{n+1} = a_n$, $b_{n+1} = \dfrac{a_n + b_n}{2}$,
  - sinon on pose $a_{n+1} = \dfrac{a_n + b_n}{2}$ et $b_{n+1} = b_n$,
  - dans les deux cas, il existe une solution x_{n+1} de l'équation $(f(x) = 0)$ dans l'intervalle $[a_{n+1}, b_{n+1}]$.

À chaque étape, on a $a_n \leq x_n \leq b_n$, on arrête le processus dès que $|f(\frac{a_n+b_n}{2})|$ est inférieure à la précision souhaitée.

Question 6 Écrire un script python permettant de rechercher par dichotomie la solution de l'équation `f(x)=0` entre `a` et `b` avec une précision `p`.

Question 7 Créer une fonction `dichotomie(f,a,b,p)` à partir de ce script. (si vous n'avez pas réussi la question précédente, créer une fonction qui permet de calculer $f(a+b+p)$.

Le script suivant utilise la fonction `dichotomie(f,a,b,p)` précédente.

```python
def g(t):
    return v(t)-1

p=10**(-6)
l1=[]
l2=[]
for borne in bornes:
    x=dichotomie(g,borne[0],borne[1],p)
    if x<1/fr:
        l1.append(x)
        l2.append(v(x))
```

Question 8 Expliquer l'intérêt de la fonction `g(t)`.

Question 9 Expliquer à quels types de variables appartiennent `l1` et `l2` et ce qu'elles contiennent.

Question 10 Expliquer l'intérêt du test `if x<1/fr` dans ce script.

La fonction `len(list)` renvoie le nombre d'éléments de la liste `list`.

Question 11 Proposer une solution pour déterminer k.

3 Valeur approchée de ξ

On souhaite modifier la valeur de f_r et choisir maintenant $f_r = 0,7$ Hz.

Question 12 Écrire sous la forme d'un mot de 32 bits respectant la norme IEEE 754 (signe, exposant, mantisse) le float $0,7$.

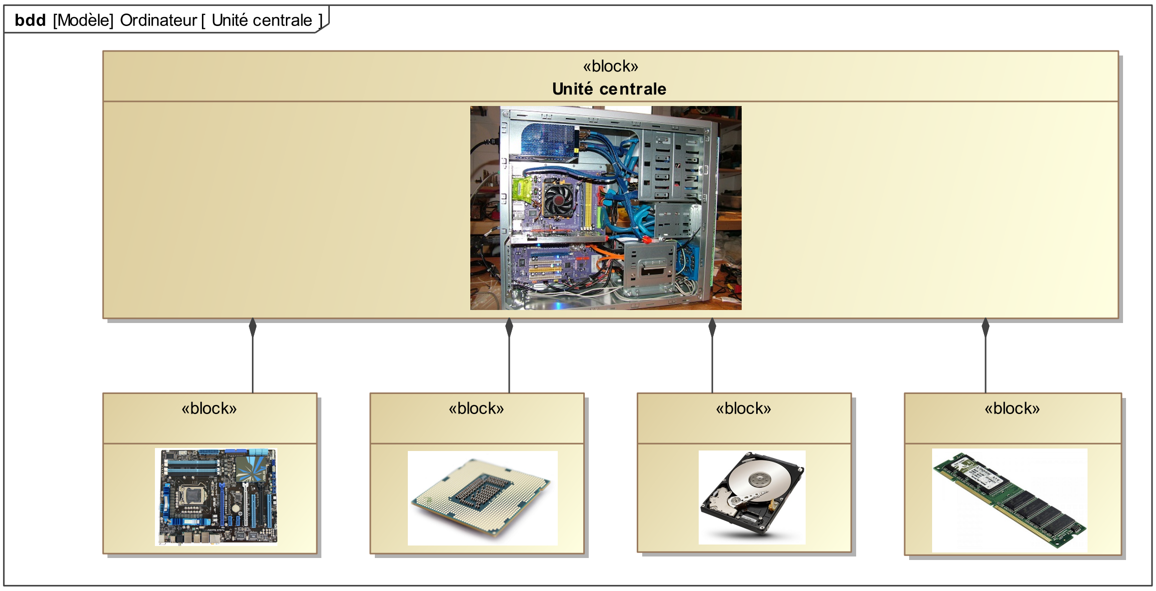
Question 13 Montrer que $00110011001100110011001 1_2 = \frac{2^{24}-1}{5}$.

On donne : $\frac{2^{-24}}{5} \approx 1.2 * 10^{-8}$.

Question 14 Déterminer l'erreur due au stockage de 0,7 à l'aide de la norme IEE74.

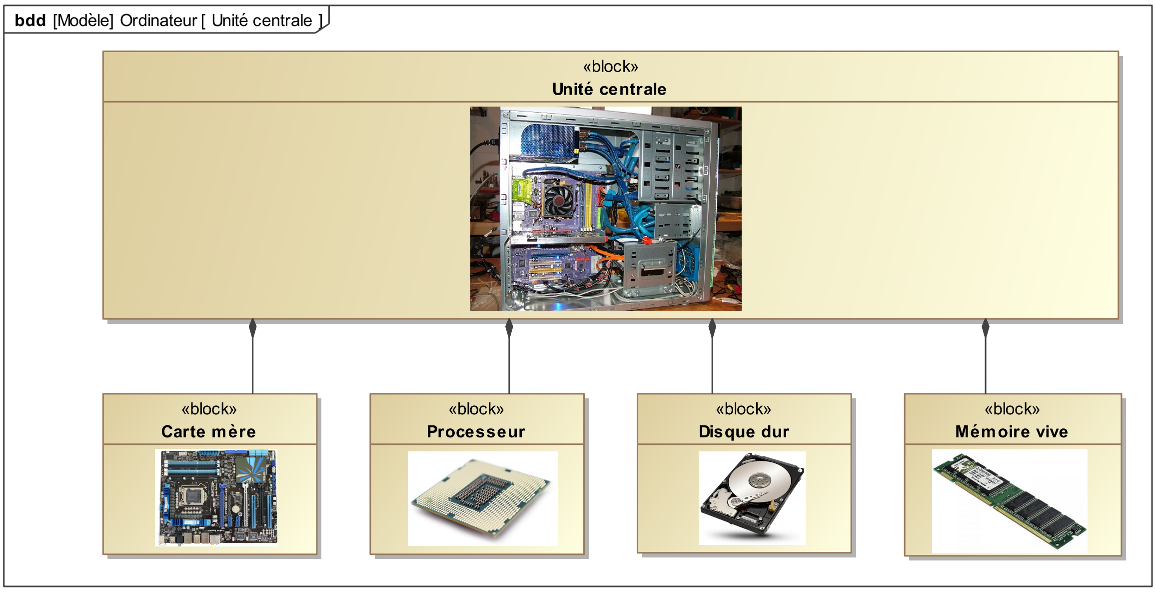
4 Document réponse

Nom :.....................
Prénom :......................



1 Correction

Question 1 :



Question 2 : La solution à choisir est la B, la A ne fonctionne pas dès la première ligne car `i` n'est pas défini. De plus, il faut ici parcourir toute la liste pour chercher un maximum, il faut donc une boucle `for` et non `while`.

Question 3 : La valeur de V_{max} est un entier, vu la valeur obtenue on suppose que le premier entier supérieur doit être la bonne réponse, donc $V_{max} = 10$ V.

Question 4 : Il y a 4 éléments dans la liste, ce sont donc les 0, 1, 2 et 3ème, donc `last=4`.

Question 5 :

```
bornes=[]
for i in range(1,len(t)):
#    if v(t[i-1])>1 and v(t[i])<1 or v(t[i-1])<1 and v(t[i])>1: (autre solution)
    if (v(t[i-1])-1.)*(v(t[i])-1.) < 0:
        bornes.append([t[i-1],t[i]])
```

Question 6 :

```
m=(a+b)/2.
while np.abs(v(m)) >  p:
    m=(b+a)/2.
    if v(a)*v(m) > 0:
        a=m
    else:
        b=m
```

Question 7 :

```python
def dichotomie(f,a,b,p):
    m=(a+b)/2.
    while np.abs(f(m)) >  p:
        m=(b+a)/2.
        if f(a)*f(m) > 0:
            a=m
        else:
            b=m
    return m
```

Si pas de question 6

```python
def dichotomie(f,a,b,p):
    return f(a+b+p)
```

Question 8 : La fonction $g(t) = v(t) - 1$ permet de rechercher la solution $v(t) = 1$ et non pas $v(t) = 0$.

Question 9 : `l1` et `l2` sont des listes qui contiennent :

— `l1` : la liste des instants t où $v(t) = 1$ (abscisses),

— `l2` : la liste des $v(t)$ à ces instants là (ce sont des valeurs proches de 1) (ordonnées).

Question 10 : Ce test permet de ne prendre en compte que les solutions dans l'intervalle $\left[0, \frac{1}{f_r}\right]$ comme demandé dans l'énoncé.

Question 11 : La courbe passe 2 fois par 1 à chaque période, on a donc `k=len(l1)/2.`

Question 12 : Le nombre à traduire est $0,7_{10}$.

0,7	x	2	=	1,4	=	1	+	0,4
0,4	x	2	=	0,8	=	0	+	0,8
0,8	x	2	=	1,6	=	1	+	0,6
0,6	x	2	=	1,2	=	1	+	0,2
0,2	x	2	=	0,4	=	0	+	0,4
0,4	x	2	=	0,8	=	0	+	0,8
0,8	x	2	=	1,6	=	1	+	0,6

...

On remarque un récurrence dans l'écriture du $0,7_{10}$ en binaire : $0,7_{10} = 0,1011001100.._2$

Le nombre stocké est alors : $1,\underbrace{01100110011001100110011_2}_{23bits} *2^{-1}$

— Signe = 0,

— Mantisse :$\underbrace{01100110011001100110011_2}_{23bits}$,

— Exposant :$127 - 1 = 126_{10} = 01111110_2$

Question 13 :

$a = \underbrace{001100110011001100110011}_{24 bits} = \underbrace{111111111111111111111111}_{24 bits} - \underbrace{110011001100110011001100}_{24 bits} = (2^{24} - 1) - 4 * a$, donc $a = \frac{2^{24}-1}{5}$.

Question 14 : Le nombre stocké est donc : $101100110011001100110011.2^{-24} = (2^{23} + \frac{2^{24}-1}{5}).2^{-24} = (\frac{1}{2} + \frac{1}{5} - \frac{2^{-24}}{5})$.

L'erreur est donc de $-\frac{2^{-24}}{5} = -1.2 * 10^{-8}$