

Exercices programmation – Matlab 1 + Corr

Rappel matlab

- les indices des tableaux commencent à 1
- vecteur ligne [1 2 3] ou [1, 2, 3]
- vecteur colonne [1 ; 2 ; 3]
- 1^e ligne d'une matrice M(1, :)
- 1^e colonne d'une matrice M(:, 1)
- les fonctions matlab sont polymorphiques
- le help contient l'entier de la documentation

Commandes

Testez les commandes suivantes dans la fenêtre de commande de matlab et observez ce qui se passe dans le *workspace*.

```
sin(pi)  
>> ans = 1.2246e-16
```

```
sin=45  
>> sin = 45 % redéfinition de la fonction sin() ☹, Matlab l'accepte mais il y aura des  
conséquences...
```

```
sin(2*pi)  
>> Subscript indices must either be real positive integers or logicals.
```

```
who  
>> Your variables are: ans sin % sin est une variable que nous avons reféfini
```

```
clear sin  
% efface la variable 'sin', la fonction sin() est de nouveau accessible
```

```
sin(4*pi)  
>> ans = -4.8986e-16
```

```
R = [1 2 3 4]  
>> R = [1 2 3 4]
```

```
S = [1, 2, 3, 4]  
>> S = [1, 2, 3, 4]
```

```
R==S  
>> ans = 1 1 1 1
```

```
S(0)  
>> Subscript indices must either be real positive integers or logicals.
```

```
S(6)  
>> Index exceeds matrix dimensions.
```

S([1,3])

```
>> ans = 1 3 % les valeurs aux index 1 et 3
```

S(1,3)

```
>> ans = 3 % la valeur à la position 1,2 (1ere ligne, 3e colonne)
```

S(2,3)

```
>> Index exceeds matrix dimensions. % S est un vecteur !
```

S([2,3])

```
>> ans = 2 3 % les valeurs aux index 2 et 3
```

S(6)=9

```
>> S = 1 2 3 4 0 9 % les valeurs manquantes sont mises à 0
```

S(end)

```
>> ans = 9 % = S(6), end dernière valeur du vecteur S
```

C=[2; 4; 6; 8]

```
>> C = 2
      4
      6
      8
```

D=2:2:8

```
>> D = 2 4 6 8
```

C==D'

```
>> ans = 1
      1
      1
      1
```

F=[C' max(C)-C']

```
>> F = 2 4 6 8 6 4 2 0 % max(C)=8 => 8 - C' => (8-2) (8-4) (8-6) (8-8) = 6 4 2 0
```

min(F)

```
>> ans = 0
```

[m p] = min(F)

```
>> m = 0
    p = 8
```

Comment créer le vecteur suivant à l'aide des ':'

-1.00 -0.50 0 0.50 1.00

```
>> -1:0.5:1
```

Comment créer le vecteur suivant à l'aide de linspace()

0 2 4 6 8

```
>> linspace(0,8,5)
```

```
A = 1:3  
>> A = 1 2 3
```

```
B = 4:6  
>> B = 4 5 6
```

```
A*B  
Error using *  
Inner matrix dimensions must agree.
```

```
A*B'  
ans = 32
```

```
A.*B  
>> ans = 4 10 18 % '.*' => multiplication éléments par élément
```

```
m = [1 2 3; 4 5 6]  
>> m = 1 2 3  
      4 5 6
```

```
m(3)  
>> ans = 2 % valeur à la position 2
```

```
m(1,3)  
>> ans = 3 % valeur de la 1ère ligne, 3e colonne
```

Comment faire pour inverser la première et la dernière colonne de **m** ?

```
>> m(:,[1 3])=m(:,[3 1]) % ';,' -> toutes les colonnes
```

Calcul de $\sin(x)$ -- Tiré du test final 2012

La fonction $\sin(x)$ peut être calculée à l'aide de la série suivante :

$$\sin(x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^{(2i+1)}}{(2i+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

Calculez $\sin(x)$ en employant le développement en série ci-dessus. Votre code calcule les n premiers éléments de la série, avec $n = 100$. **Votre code ne doit pas contenir de boucle.** Décomposez votre code en plusieurs séries qui seront combinées pour obtenir le résultat final. Le calcul de factorielle se fait à l'aide de la fonction matlab $\text{factorial}(x)$.

Avec :

```
n=100
x= pi/2
```

Définissez les séries suivantes :

```
0 à n pour les i
(-1)^i
(2*i+1)
etc.
```

Combinez vos séries et vérifiez votre résultat.

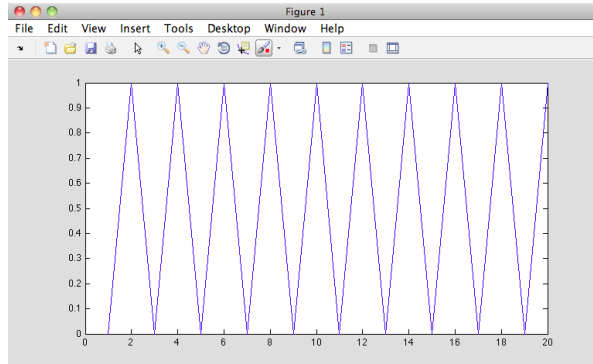
Rappel: mettre un '.' avant un opérateur pour faire une opération élément par élément.

```
n=100;
x=pi/2;

idx = 0:n;           % index of the sum coeff., 0..inf -> 0..n
s = (-1).^idx;      % sign for the sum coeff. -> 1 -1 1 -1 1 ...
p = 2 * idx + 1;    % for ()^2i and (2i)! -> 1 3 5 7 9 ...
num =s.* x.^p       % numerator: x.^p
den = factorial(p); % den: factorial
mySIN = sum(num./den) % sin = sum(num./den)
```

Signal binaire

A l'aide des commandes vues dans le 1^{er} module Matlab, générer un signal binaire qui oscille entre 0 et 1.

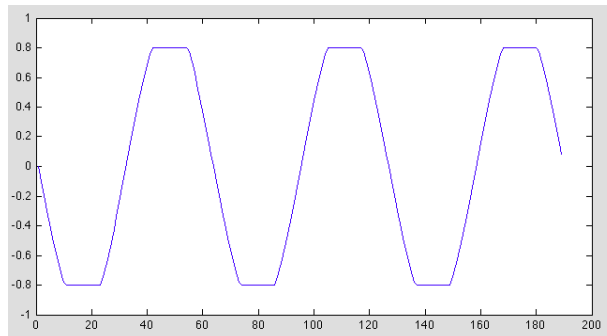


Hints : Help mod()

```
>> plot(mod(0:19,2)) % mod(0:19,2) => mod([0 1 2 3 ..19 ],2) -> 0 1 0 1 0 ...
```

Sinusoïdales tronquées

A l'aide des commandes vues dans le 1^{er} module Matlab, créer un signal formé de 3 sinus bornés entre -0.8 et + 0.8.



Hints : Help min() et max()

```
>> plot(max(min(sin(0:0.1:20),0.8),-0.8)) % décomposer l'opération en plusieurs étapes  
% sin (0 à 20 avec un pas de 0.1)  
% min(sin(), 0.8) ->sature à 0.8  
% max(sin(), -0.8) ->sature à -0.8
```

funtool

Taper « funtool » dans la fenêtre de commande de matlab et explorer la puissance du calcul symbolique de matlab. Ex. $f = (\cos(x)^2 + \sin(x)^2)$ et appuyer sur le bouton **simplify f**.