

Matlab: applications en mécanique

LA207, Université Pierre et Marie Curie.

2.1 TP1: une prise en main

Dans ce TP, nous mettons en place des pratiques de base qui vont nous servir plus tard dans les TP en contexte mécanique.

Compétences techniques: Matlab comme une calculatrice; Tableaux: construction, et accès aux éléments; la fonction `plot`; boucles `for`; Créer votre compte-rendu; insérer les figures dans le compte-rendu; Rendre votre compte-rendu sur le site "Australe". Contexte pour la partie "Etude": calculer une approximation de π par des séries.

2.1.1 Manipulations

Dans les questions qui suivent, tapez à l'invite de Matlab les lignes suivantes les unes après les autres (et dans l'ordre...). Observez bien les résultats, essayez de les comprendre, et construisez ainsi une pratique intuitive de l'environnement de programmation. Les phrases que l'on vous demande de rédiger doivent être écrites dans le compte-rendu.

1. Créez un répertoire nommé "TP1" sur l'ordinateur. Avec openoffice sous Linux ou powerpoint sous Windows, créez un nouveau document que vous sauvez dans ce répertoire, avec une page de présentation avec les noms/numéros d'étudiants et le nom de votre encadrant TP.
2. Calculatrice, variables scalaires, premiers tableaux

```
>> 5
>> 2+5
(1) >> a=10
>> a
>> disp(a)
(2) >> a+5
>> b=a+5
>> b=a+5;
>> c=[1,-2,7,0,10]
>> 2*c
>> d=[1;2;3;4;5]
>> size(a)
(3) >> size(c)
>> whos
(4) >> c(3)
>> d(1)+d(2)+d(3)+d(4)+d(5)
>> c(1)=0
```

2.1. TP1: UNE PRISE EN MAIN

```
>> e=[0, 0; 0, 0]
>> pi
(5) >> cos(2*pi)
>> 2^3
```

Pour les lignes qui sont numérotées (1), (2), ... Décrire dans votre compte-rendu en une phrase ce que ces commandes font, exemple pour (1): "la variable `a` est créée, et reçoit le scalaire 10, puis la valeur de `a` est affichée à l'écran"; pour (2): "la valeur contenue dans la variable `a` est additionnée à 5 et le résultat est affiché à l'écran".

3. Messages d'erreur. Tapez l'une après l'autre les commandes suivantes, qui contiennent des erreurs, lisez les messages d'erreur.

```
(0) >> toto
>> c(6)
(1) >> c+d
(2) >> f=[0, 0 ; 0]
>> size
```

Pour les commandes (1) et (2), expliquez en une phrase l'erreur. Par exemple pour (0): "il n'existe pas de variable nommée `toto`, donc je ne peux pas afficher sa valeur à l'écran".

4. Boucle `for`. Tapez l'une après l'autre les commandes suivantes à l'invite de Matlab:

```
(0) >> for ind=1:10; disp(ind); end
>> a=0; for ind=1:10; a=a+ind; disp(a); end
(1) >> a=1; for ind=1:2:10; a=a*ind; disp(a); end
```

Décrivez en une phrase ce que fait la commande (1). Par exemple pour (0): "pour la variable `ind` allant de 1 à 10 par pas de 1, on affiche à l'écran la valeur de cette variable".

5. Nous allons maintenant voir les tableaux en plus de détails. Tapez les commandes suivantes

```
(0) >> a=[0,0; 0,0]
>> a(2,2)=1
(1) >> a(1,1)=a(2,2)
(2) >> b=[a,a]
>> c=[0*a, a, 2*a, 3*a]
```

et expliquez en une phrase ce que font les commandes (1) et (2). Par exemple pour (0): "On construit la matrice `a` avec deux colonnes et deux lignes remplies de zéros".

6. Fonctions mathématiques: tapez ces commandes et observez le résultat.

```
>> sin(2.5)^2+cos(2.5)^2
>> exp(log(3))
>> sqrt(-1)
```

7. Quelques fonctions associées aux tableaux

```
(0) >> x=linspace(0,2*pi,10)
(1) >> y=0:0.5:10
>> max(x)
>> min(x)
>> sum(x)
(2) >> prod(x)
>> sin(x)
>> x=logspace(-2,2,5)
```

expliquez ce que font les commandes (1) et (2). Par exemple pour (0): "On crée le vecteur x de taille une ligne et 10 colonnes, qui contient 10 valeurs équiréparties entre 0 et 2π ".

8. Graphiques avec la fonction `plot`. Tapez les commandes suivantes

```
(0) >> plot([0,1],[1,0], 'k*-' )
>> xlabel('x'); ylabel('y'), title('joli graph')
>> xlim([-1,2]); ylim([-1,2])
(1) >> x=linspace(0,4*pi,30); plot(x,sin(x), 'r*--')
```

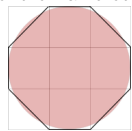
Expliquez en une phrase la commande (1). Par exemple pour (0): "On trace une ligne continue de couleur noire entre les points de coordonnées (0, 1) et (1, 0), on visualise chacun des deux points avec une astérisque".

9. Enregistrer votre figure comme une image: A l'aide du menu déroulant "Fichier", sauver votre figure sur le disque sous un format d'image, puis insérez cette image dans votre compte-rendu.

10. Chercher dans l'aide matlab: expliquer ce que fait la fonction `factorial`. Chercher dans la documentation le nom de la fonction qui arrondi un nombre réel à l'entier inférieur. (arrondi π à 3 par exemple).

2.1.2 Etude

Pour tous les cercles, il existe un rapport constant entre le périmètre et le diamètre, et entre l'aire et le carré du rayon. Cette constante a été nommée π .



Dans l'illustration, le disque a pour diamètre 9. L'aire du disque est légèrement supérieure à l'aire de l'octogone irrégulier obtenu en rognant les coins

2.1. TP1: UNE PRISE EN MAIN

du carré de côté 9. Cet octogone a pour aire 63, l'aire du disque est alors évaluée à 64 soit l'aire d'un carré de côté 8. On en tire:

$$\pi \approx \frac{A}{r^2} = \frac{64}{(9/2)^2} = \frac{256}{81} \approx 3.1601$$

Cette approximation nous vient d'un ancien papyrus égyptien (cf. Wikipedia).

Les développements limités nous offrent une méthode itérative pour calculer les décimales de π . Par exemple:

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} \dots = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{2k+1}$$

1. Nous allons utiliser ce développement limité pour calculer une approximation de π , Sachant que $\arctan(1) = \pi/4$, écrivez une boucle `for` qui calcule la valeur de

$$P(n) = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + \frac{(-1)^n}{2n+1} = \sum_{k=0}^n \frac{(-1)^k}{2k+1}$$

pour un entier n donné.

2. Insérez votre script avec des commentaires dans votre compte-rendu. Vous donnerez également les valeurs numériques de $p(5)$, $p(10)$, $p(100)$ ainsi que les cinq premières décimales de la constante π .
3. Remettez votre compte-rendu sur Australe: chaque étudiant du groupe doit remettre le compte-rendu. Vous signerez la feuille de présence une fois confirmé auprès de votre encadrant TP que votre compte-rendu est remis. Si vous n'avez pas accès à l'onglet LA207 ou bien à "Mon UPMC", manifestez vous auprès de votre encadrant de TP.

2.1.3 Pour aller plus loin

Voici une autre formule:

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

Attention! Cette formule, qui est due au mathématicien indien Ramanujan, converge extrêmement rapidement...

Sauriez-vous coder celle ci?

$$\frac{\pi}{2} = \frac{2}{\sqrt{2}} \times \frac{2}{\sqrt{2+\sqrt{2}}} \times \frac{2}{\sqrt{2+\sqrt{2+\sqrt{2}}}} \times \frac{2}{\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}} \times \dots$$

Compte-rendu modèle pour le TPI (2011)

manipulations

Q2)

(3): On fait appel à la fonction “size”, qui affiche à l’écran le nombre de lignes et le nombre de colonnes du tableau c

(4): On affiche à l’écran la valeur numérique mémorisée dans la troisième case du tableau c. (S’il y a une seule ligne ou bien une seule colonne, on n’a pas besoin de mettre les deux indices c(3,1))

(5) On affiche à l’écran la valeur du cosinus de 2 pi, qui est égale à 1.

Q3)

(1): **??? Error using ==> plus, Matrix dimensions must agree.**

On a essayé d’additionner deux tableaux qui n’ont pas la même taille.

(2) **??? Error using ==> vertcat, CAT arguments dimensions are not consistent.**

On a essayé de construire un tableau qui n’a pas le même nombre de colonne pour chaque ligne, ce qui est impossible puisqu’un tableau doit être rectangulaire. Vertcat signifie “concaténation verticale”.

Q4)

(1) On initialise la variable “a” avec la valeur 2, puis on fait une boucle de 1 à 10 par pas de 2. A chaque itération, on multiplie “a” par l’indice “ind” de la boucle et on affiche la valeur de “a” à l’écran.

Q5)

(1) On remplace la valeur mémorisée dans la case de la première ligne première colonne par la valeur mémorisée dans la case de la seconde ligne seconde colonne.

(2) On construit le tableau “b” en concaténant “a” côte à côte avec lui même.

Q7)

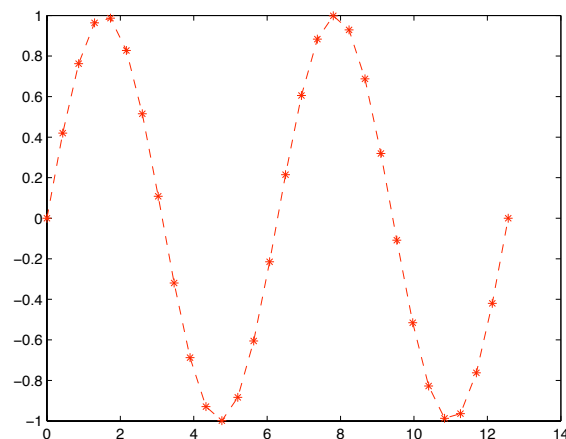
(1) On construit un tableau à une ligne d’éléments linéairement répartis entre 0 et 10, avec un pas de 0.5 (c’est très similaire à la fonction linspace, sauf que ici on donne le pas, plutôt que le nombre d’éléments)

(2) On affiche à l’écran le produit de toutes les cases du tableau “x”.

Q8)

On construit un tableau de 30 éléments linéairement répartis entre 0 et 4 pi, puis on trace la courbe des points qui ont les valeurs des cases de x en abscisse et les valeurs des cases de sin(s) en ordonnée. La courbe est tracée en rouge avec une ligne hachurée, et des marqueurs en astérisque.

La courbe de la question 8



Voici ce que nous dit l'aide en ligne de Matlab pour l'utilisation de la fonction factorial:

```
>> help factorial
```

FACTORIAL Factorial function.

FACTORIAL(N) for scalar N, is the product of all the integers from 1 to N, i.e. $\text{prod}(1:N)$. When N is an N-D matrix, **FACTORIAL(N)** is the factorial for each element of N. Since double precision numbers only have about 15 digits, the answer is only accurate for $N \leq 21$. For larger N, the answer will have the correct order of magnitude, and is accurate for the first 15 digits.

See also `prod`.

Reference page in Help browser
`doc factorial`

Script

Approximation de pi avec une série convergente:

Dans ce script, j'en fait plus que ce qui est demandé dans l'énoncé: je mémorise dans le tableau pvec au cours des itérations l'évolution de la somme partielle de sorte à pouvoir en tracer le graphique, ainsi que pour tracer l'évolution de l'erreur d'approximation au cours des itérations.

Semilogy est une fonction très similaire à plot, mais qui utilise un axe des ordonnées logarithmique.

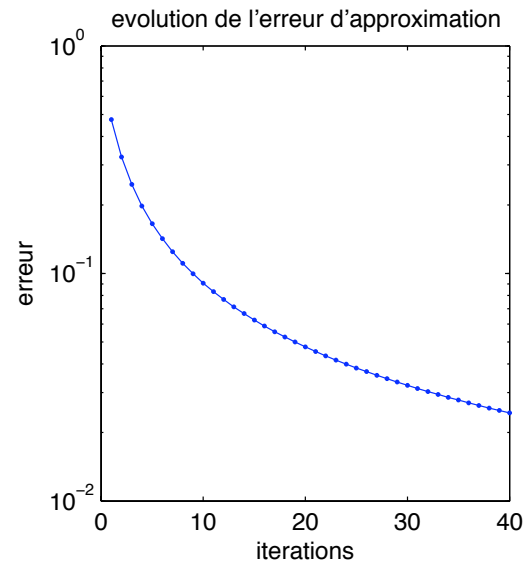
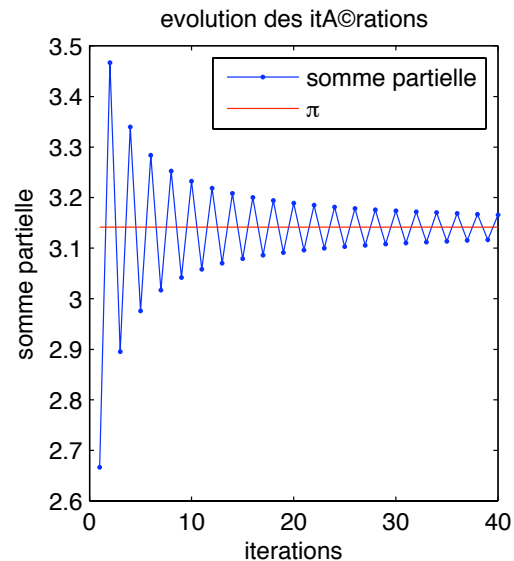
A ne pas oublier: initialisation de la valeur de la somme partielle avant de commencer la boucle.

```
% un script pour calculer pi
clear all
clf
n=40 % nombre d'iteration
kvec=1:n; % les indices d'itération

pp=1; % Initialisation de la somme partielle
pvec=0*kvec; % Tableau pour mémoriser l'évolution de la somme partielle

% La boucle
for k=kvec;
pp=pp+(-1)^k/(2*k+1); % Voici la commande qui est importante
pvec(k)=4*pp; % Mémorisation de la somme partielle
end

% Tracé des graphiques
subplot(1,2,1); plot(kvec,pvec,'b.-'); hold on
plot(kvec,kvec*0+pi,'r');
subplot(1,2,2); semilogy(kvec,abs(pvec-pi),'b.-'); hold on
```



Script

```
% la série en racine de 2
kvec=1:n;
aa=0;
pp=1;
for k=kvec;
aa=sqrt(2+aa);
pp=pp*2/aa;
pvec(k)=2*pp;
end

subplot(1,2,1); plot(kvec,pvec,'c.-');
subplot(1,2,2); semilogy(kvec,abs(pvec-pi),'c.-')

% ramanujan: 10^-7 de précision à la première itération!
pvec=0*kvec;
pp=0;
for k=kvec-1;
pp=pp+factorial(4*k)*(1103+26390*k)/(factorial(k)^4*396^(4*k));
pvec(k+1)=9801/(2*sqrt(2)*pp);
end
subplot(1,2,1); plot(kvec,pvec,'k.-');
subplot(1,2,2); semilogy(kvec,abs(pvec-pi),'k.-')
```

Pour aller plus loin:

On a maintenant deux séries qui convergent plus rapidement. La méthode de Ramanujan converge extrêmement vite: l'erreur entre la somme partielle et la valeur exacte de pi est plus petite que 10^{-15} en trois itérations.

