

Calcul scientifique : examen du 20 juin 2005

L1 : Licence sciences et technologies,
mention mathématiques, informatique et applications

Nombre de pages de l'énoncé : 3. Durée 1 heure 30.

NB : L'examen se compose de 10 questions indépendantes. Pour chaque question 5 affirmations sont proposées, parmi lesquelles 2 sont vraies et 3 sont fausses. Pour chaque question, indiquez sur votre copie les lettres des 2 affirmations que vous pensez vraies. Chaque question pour laquelle les 2 affirmations vraies sont données rapporte 2 points.

Tout document est interdit. Les calculatrices et les téléphones portables, même à titre d'horloge, sont également interdits.

Attention : les lignes de commande sont indépendantes les unes des autres, c'est-à-dire que l'on suppose que pour chaque affirmation proposée on vient de démarrer une nouvelle session Scilab.

Rappel : Les codes des couleurs sont 1 pour noir et 5 pour rouge.

Question 1. Les lignes proposées sont écrites dans un éditeur et sauvegardées dans un fichier intitulé `f.sci`. On utilise ensuite la commande `getf("f.sci")`. Cela permet de définir une fonction qui à deux matrices carrées de même taille $A = (a_{i,j})$ et $B = (b_{i,j})$, associe $C = (c_{i,j})$ telle que $c_{i,j} = \min(a_{i,j}, b_{i,j})$:

A : (FAUX) `C=min(A,B)`

B : (VRAI) `function C=f(A,B)`
`C=B;`
`bool=A<B;`
`C(bool)=A(bool);`
`endfunction`

C : (FAUX) `n=size(A); for i=1:n(1), for j=1:n(2), C(i,j)=min(A(i,j),B(i,j)); end; end;`

D : (VRAI) `function C=f(A,B)`
`C=min(A,B);`
`endfunction`

E : (FAUX) `function f(A,B)`
`C=min(A,B);`
`endfunction`

Question 2. La commande `plotframe([-2,-2,2,2],[2,10,2,10]);` suivie de la ligne de commande proposée affiche un rectangle.

A : (VRAI) `x=[-1,1,1,-1,-1];y=[-1,-1,1,1,-1]; plot2d(x,y);`

B : (FAUX) `x=[-1,-1,1,1];y=[-1,1,-1,1]; plot2d(x,y);`

C : (VRAI) `x=[-1,-1,1]';y=[-1,1,1]'; plot2d([x,y],[y,x],[1,1])`

D : (FAUX) `x=[-1,-1,1];y=[-1,1,1]; plot2d([x,y],[y,x],[1,1])`

E : (FAUX) `x=[-1,-1,1,1;-1,1,1,-1]; y=[-1,1,1,-1;1,1,-1,-1]; plot2d(x,y,[1,1,1]);`

Question 3. Les lignes proposées sont écrites dans un éditeur et sauvegardées dans un fichier intitulé `f.sci`. On utilise ensuite la commande `getf("f.sci")`. Cela permet de définir une fonction qui prend en entrée une matrice $A = (a_{ij})$ et retourne en sortie une matrice $B = (b_{ij})$ de mêmes dimensions telle que $b_{ij} = a_{ij}$ si i est pair et $b_{ij} = 0$ sinon.

A : (FAUX) `function B=f(A)`
`B=kron(A,[0;1]);`
`endfunction`

B : (FAUX) `function B=f(A)`
`for i=1:n,`
`B(i,:)=0;`
`end;`
`endfunction`

C : (VRAI) `function B=f(A)`
`B=A; n=size(A,"r");`
`for i=1:2:n,`
`B(i,:)=0;`
`end;`
`endfunction`

D : (VRAI) `function B=f(A)`
`B=A; n=size(A,"r");`
`indices = find((-1)^[1:n]==-1);`
`B(indices,:)=0;`
`endfunction`

E : (FAUX) `function B=f(A)`
`C=ones(size(A,"c"),1)*((-1)^[1:size(A,"r")]);`
`B=(A+A*C)/2;`
`endfunction`

Question 4. Les opérations suivantes définissent la fonction `moins` qui retourne la différence de deux matrices :

A : (FAUX) Sauver dans le fichier `f.sci` les lignes
`function C=f(A,B)`
`C=A-B;`
`endfunction`
 puis charger le fichier par `getf("f.sci")`

B : (VRAI) Sauver dans le fichier `f.sci` les lignes
`function C=moins(A,B)`
`C=A-B;`
`endfunction`
 puis charger le fichier par `getf("f.sci")`

C : (FAUX) Sauver dans le fichier `f.sci` les lignes
`function moins(A,B)`
`C=A-B;`
`endfunction`
 puis charger le fichier par `getf("f.sci")`

D : (FAUX) Sauver dans le fichier `f.sci` les lignes
`function C=moins(A,B)`
`for i=1:n, for j=1:n, C(i,j)=A(i,j)-B(i,j); end; end;`
`endfunction`
 puis charger le fichier par `getf("f.sci")`

E : (VRAI) Sauver dans le fichier `f.sce` la ligne
`def("C=moins(A,B)","C=A-B")`
 puis charger le fichier par `exec("f.sce")`

Question 5. Les lignes de commande suivantes affichent une représentation graphique correcte de la fonction :

$$f(x) = \frac{1}{\sin(x)}, \text{ pour } x \in]-\pi, +\pi[:$$

- A : (VRAI) `e=0.1; x=linspace(e,%pi-e,200); x=[x'-%pi,x']; y=(1)./sin(x); plot2d(x,y,[5,5]);`
- B : (FAUX) `x=[0:0.001:%pi]; x=[x'-%pi,x']; y=(1)./sin(x); plot2d(x,y,[5,5]);`
- C : (FAUX) `x=[-%pi:%pi]; y=(1)./sin(x); plot(x,y);`
- D : (FAUX) `x=[-%pi+0.1:0.1:%pi-0.1]; y=1/sin(x); plot(x,y);`
- E : (VRAI) `x=[0.1:0.01:%pi-0.1]'; y=(1)./sin(x); plot2d([x-%pi,x],[-y,y],[5,5]);`

Question 6. La ligne de commande suivante :

`x=linspace(-%pi,%pi,50)'; y=[]; for i=-1:1, y=[y,sin(x.*i)]; end; plot2d([x,x,x],y);`

- A : (FAUX) trace les courbes $y = \sin(-x)$, $y = 0$ et $y = \sin(x)$ de même couleur.
- B : (FAUX) renvoie un message d'erreur.
- C : (VRAI) trace les courbes $y = \sin(-x)$, $y = 0$ et $y = \sin(x)$ avec des couleurs différentes.
- D : (VRAI) donne le même résultat que :
`x=linspace(-%pi,%pi,50)'; y=[sin(-x),zeros(50,1),sin(x)]; plot2d([x,x,x],y,[1,2,3]);`
- E : (FAUX) ne trace que la courbe $y = \sin(x)$.

Question 7. La ligne de commande suivante affiche une valeur numérique proche de $\int_0^{10} \sqrt{t} dt$:

- A : (VRAI) `integrate("sqrt(x)","x",0,10)`
- B : (FAUX) `t=[0,10]; inttrap(t,sqrt(t))`
- C : (VRAI) `t=[0:0.001:10]; inttrap(t,sqrt(t))`
- D : (FAUX) `t=[0:0.001:10]; cumsum(sqrt(t))`
- E : (FAUX) `t=[0:0.001:10]; sum(sqrt(t)/0.001)`

Question 8. On pose $A=[1:4;4:-1:1;0:3]$ et $v=[2;2;3]$:

- A : (FAUX) $v' * A * v$ est un réel.
- B : (VRAI) $v * v'$ est une matrice à trois lignes et trois colonnes.
- C : (FAUX) $v(3:-1:1)' == A(3,:)$ donne [T T T]
- D : (VRAI) $v(3:-1:1) == A(:,3)$ donne [T T T]'
- E : (FAUX) $B=A'$; $B(:,3) == v$ donne [T T T]

Question 9. La ligne de commande proposée affiche le vecteur ligne $v=[1,2,3,\dots,99,100]$:

(NB : v est le vecteur ligne qui contient *tous* les entiers de 1 à 100, mais pour des raisons évidentes de place, nous n'avons pu tous les écrire...)

- A : (VRAI) `v=linspace(1,100,100)`
- B : (FAUX) `v=[1:0.1:100]`
- C : (VRAI) `v=[]; for i=1:100, v=[v,i]; end; v`
- D : (FAUX) `v=1; for i=1:100, v=[v,i+1]; end; v`
- E : (FAUX) `for i=1:100, v=[v,i]; end; v`

Question 10. Soit $A=[0:3;3:-1:0]+%i*[1:4;4:-1:1]$. La ligne de commande proposée affiche une matrice réelle ou complexe à deux lignes et quatre colonnes :

- A : (FAUX) `A.'`
- B : (FAUX) `conj(A);`
- C : (VRAI) `real(A)`
- D : (VRAI) `abs(A)`
- E : (FAUX) `real(A)<imag(A)`