Université René Descartes – Paris 5
UFR de Mathématiques et Informatique
45, rue des Saints-Pères 75270 Paris cedex 06

CORRIGÉ DE L'EXAMEN PARTIEL
LICENCE 2ème année
Environnement de calcul scientifique et modélisation

Année 2006 - 2007
23 avril 2007

Questions de cours :

QC1.1
• Av est un vecteur colonne $(n,1)$ et $(Av)_i = \sum_{k=1}^{n} A_{ik}v_k$, $(1 \leq i \leq n)$, ce qui fait n multiplications et $n-1$ additions, pour chacune des n coordonnées, d'où un total de $2n^2 - n$ opérations ;
• v^tA est un vecteur ligne $(1,n)$ et $(v^tA)_j = \sum_{k=1}^{n} v_kA_{kj}$, $(1 \leq j \leq n)$, ce qui donne comme avant $2n^2 - n$ opérations ;
• v^tv est un scalaire et $v^tv = \sum_{k=1}^{n} v_kv_k$, et l'on effectue en tout $2n-1$ opérations ;
• vv^t est une matrice (n,n) et $(vv^t)_{ij} = v_iv_j$, $(1 \leq i,j \leq n)$, ce qui fait une multiplication pour chacun des n^2 coefficients de la matrice,
Rq. : on peut réduire les calculs grâce à la symétrie, $(vv^t)_{ij} = (vv^t)_{ji}$, il suffit de calculer $n + (n-1) + ... + 1 = (n+1)n/2 = n^2/2 + n/2$ termes.

QC1.2
• vv^tA est une matrice (n,n) ;
• (vv^t) est aussi une matrice (n,n) dont le calcul nécessite $n^2/2 + n/2$ opérations ; le produit de deux matrices (n,n) nécessite n^3 multiplications et $n^2(n-1)$ additions, d'où un total de $2n^3 - 1/2n^2 + n/2$ opérations pour calculer $(vv^t)A$;
ce qui fait $O(n^3)$ opérations ;
• pour déterminer le vecteur ligne v^tA il faut $2n^2 - n$ opérations ; le calcul de $v(v^tA)$ nécessite 1 multiplication pour chacun des n^2 coefficients car il n'y a pas de symétrie, en tout il faut $3n^2 - n$ opérations ;
ce qui fait $O(n^2)$ opérations .
• On vérifie ainsi le résultat du cours : on n'a pas nécessairement le même nombre d'opérations dans l'évaluation de $ABC = (AB)C = A(BC)$

QC2
1. Il faut faire attention aux *singularités*, à la *taille* de la fenêtre de représentation et à l'*échantillonage* de l'intervalle $[0,1]$.
2. `deff("y=f(x)","y=10+log10(abs(x-0.5))")`
3. `x  = [ 0.  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.]` la singularité de f, $x = 1/2$, sera évalué (sauf si l'on utilise le mode `ieee(2)`).
4. Pour avoir des valeurs négatives il faut $|x - 1/2| \leq 10^{-10}$;
avec un pas inférieur à 10^{-10} on obtiendra des valeurs négatives, en faisant attention de ne pas trop ($=$`%eps`) se rapprocher de 1/2.

Exercice 1

1. est faux (dessine une seule courbe) et 2. est correct.

Exercice 2 **1.** Une possibilité de nom de fichier est `polaire.sci`, en tous les cas le nom doit se terminer en `.sci`. Après un `getf` il suffit d'appeler `polaire` avec les bons arguments.
2. La commande `polaire(1,1)` ou `r=polaire(1,1)` donnera $\sqrt{2}$,
tandis que `[r,t]=polaire(1,1)` donnera $t = \pi/4$ et $r = \sqrt{2}$.
3. Il faut étendre la fonction au plan complexe entier, c.-à-d. sur les quatre quadrants. La détermination de `r=sqrt(x^2+y^2)` reste inchangé, mais :
pour $x > 0$, `theta= atan(y/x)` ;
pour $x = 0$, on a `theta= signe(y)* %pi/2`,
où `signe(y)` vaut 1 si $y > 0$, 0 si $y = 0$ et -1 si $y < 0$;
pour le cas $x < 0$, on a `theta= atan(y/x) + signe(y) * %pi`.
Remarque : tout ceci est fait par l'appel suivant en *Scilab* : `atan(y,x)`.
4. Il suffit d'utiliser des opérations élément par élément :
`r=sqrt(x.^2+y.^2)` et `theta=atan(y./x)` .

Exercice 3
1. Les entrées B et C doivent être des matrices $(4, 4)$, dans ce cas la sortie A est une matrice $(4, 4)$. Si B et C ne passent pas les deux tests, on obtient un message d'erreur.
2. Elle calcule le produit matriciel `A=B*C` par blocs :

$$\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} B_{11}C_{11} + B_{12}C_{21} & B_{11}C_{12} + B_{12}C_{22} \\ B_{21}C_{11} + B_{22}C_{21} & B_{21}C_{12} + B_{22}C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = A$$

où les A_{ij} sont des sous-matrices $(2, 2)$ de A :
A_{11} =`A(1:2,1:2)`, A_{12} =`A(1:2,3:4)` A_{21} =`A(3:4,1:2)`, A_{22} =`A(3:4,3:4)`.
De même pour B et C.

3. La multiplication matricielle s'écrit avec boucles :

```
 function A=tata(B,C)
 if (size(B)<>[4,4]) then
  error("Mauvaise dimension")
 end
 if (size(B)<>size(C)) then
  error("Dimensions incompatibles")
 end
 A=zeros(4,4)
 for i=1:4
  for j=1:4
   for k=1:4
     A(i,j)=A(i,j)+B(i,k)*C(k,j)
   end
  end
 end
endfunction
```

4. D'après le cours, les opérations par blocs sont plus efficaces en temps de calcul que les boucles. Ce qui peut se vérifier sur machine :

```
-->B=rand(4,4);
-->C=rand(4,4);
-->timer(); for i=1:100000, A=toto(B,C);end; timer()
 ans  =    9.34
-->timer(); for i=1:100000, A=tata(B,C);end; timer()
 ans  =   61.01
```