mercredi 14 janvier 2009

TP 3 d'analyse numérique

Le but de ce TP est de manipuler des tableaux à une dimension pour représenter des matrices ou des vecteurs réels (double précision), les multiplier, calculer des déterminants, calculer l'inverse d'une matrice, ...

Le fichier "`http://10.27.3.235/TP/matrices.txt`" contient les définitions des matrices A, B et C, de dimensions 3×3, 4×4 et 8×8. Il contient aussi les définitions des vecteurs $\vec{a}$, $\vec{b}$ et $\vec{c}$, de dimensions 3, 4 et 8. Vous pouvez copier ce fichier dans votre compte et l'inclure dans votre programme principal.

1. Ecrire en C, une fonction qui affiche sur l'écran une matrice carrée ($n \times n$), ligne par ligne. Vous appellerez cette fonction "`afficher_matrice`", et elle devra avoir 2 arguments : le premier argument sera la matrice (tableau) et le deuxième argument la dimension n. Dans le programme principal, cette fonction sera par exemple utilisée comme ceci :
```
afficher_matrice(A,n);
```
Vérifier, en écrivant un petit programme, que cette fonction affiche correctement les matrices A, B et C citées plus haut.

2. Ecrire en C, une fonction qui affiche sur l'écran un vecteur (de dimension n). Vous appellerez cette fonction "`afficher_vecteur`", et elle devra avoir 2 arguments : le premier argument sera le vecteur (tableau) et le deuxième argument la dimension n. Vérifier, en écrivant un petit programme, que cette fonction affiche correctement les vecteurs $\vec{a}$, $\vec{b}$ et $\vec{c}$ citées plus haut.

3. Ecrire en C, une fonction qui calcule le produit $\vec{y} = M\vec{x}$ (M est une matrice $n \times n$, et les vecteurs $\vec{x}$ et $\vec{y}$ ont n composantes). Vous appellerez cette fonction "`produit_matrice_vecteur`", et elle devra avoir 4 arguments. Cette fonction sera utilisée dans un programme comme ceci :
```
produit_matrice_vecteur(M,x,y,n);
```
Ecrire ensuite un petit programme qui calcule et affiche correctement les produits $A\vec{a}$, $B\vec{b}$ et $C\vec{c}$ (ce sont les matrices et vecteurs du fichier cité plus haut).

4. Ecrire en C, une fonction qui calcule le produit matriciel $R = MN$ (M, N et R sont des matrices $n \times n$) Vous appellerez cette fonction "`produit_matrice_matrice`", et elle devra avoir 4 arguments. Cette fonction sera utilisée dans un programme comme ceci :
```
produit_matrice_matrice(M,N,R,n);
```
Ecrire ensuite un petit programme qui calcule et affiche correctement les produits AA^t, A^tA, BB^t, B^tB, CC^t et C^tC (A^t est la transposée de A). Vérifier que les résultats que vous obtenez sont bien des matrices symétriques (en effet AA^t est égale à sa transposée $(AA^t)^t$).

5. Ecrire en C, une fonction appelée "`determinant`" qui calcule, en utilisant la méthode de Gauss, le déterminant d'une matrice carrée M, de dimension $n \times n$. Elle devra avoir 2 arguments et retourner un argument qui est le déterminant. Cette fonction sera utilisée dans un programme comme ceci :

```
z = determinant(M,n); /* z est un double déclaré auparavant*/
```

Ecrire ensuite un petit programme qui calcule et affiche les déterminants des matrices A, B et C.

6. Ecrire en C, une fonction appelée "`resoudre_avec_gauss`" qui résoud le système d'équations linéaires $A\vec{x} = \vec{b}$ (la matrice A de dimension $n \times n$, et le vecteur $\vec{b}$ sont supposés donnés). Cette fonction devra avoir 4 arguments et sera utilisée dans un programme comme ceci :

```
z = resoudre_avec_gauss(A,b,x,n); /* solution dans "x" */
```

Ecrire ensuite un petit programme qui résoud les systèmes $A\vec{x} = \vec{a}$, $B\vec{x} = \vec{b}$ et $C\vec{x} = \vec{c}$.
Vérifier ensuite que les solutions obtenues sont correctes en affichant sur l'écran les produits $A\vec{x}$, $B\vec{x}$ et $C\vec{x}$, et en les comparant aux vecteurs $\vec{a}$, $\vec{b}$ et $\vec{c}$.

7. Ecrire en C, une fonction appelée "`matrice_inverse`" qui calcule, en utilisant la méthode de Gauss, l'inverse M^{-1} d'une matrice carrée M de dimension $n \times n$. Cette fonction devra avoir 4 arguments et sera utilisée dans un programme comme ceci :

```
z = matrice_inverse(M,M_inverse,n); /* solution dans "M_inverse" */
```

Ecrire ensuite un petit programme qui calcule les inverses A^{-1}, B^{-1} et C^{-1}.
Vérifier ensuite que les solutions obtenues sont correctes en calculant et en affichant sur l'écran les produits AA^{-1}, BB^{-1} et CC^{-1} (vous devez obtenir la matrice identité).