1 Intégration

Exercice 1
Calculer une valeur approchée de
$$\int_0^1 \frac{dx}{1+x}$$
par la méthode des rectangles, du point milieu et des trapèzes en utilisant un pas de 1/10 et de 1/100 (vous pouvez utiliser le tableur ou écrire un programme effectuant ce calcul avec comme arguments la fonction, la borne inférieure, la borne supérieure et le nombre de subdivision). Observez numériquement la différence entre les valeurs obtenues et la valeur exacte de l'intégrale.

Exercice 2
Calculer le polynôme interpolateur P de Lagrange de $f(x) = \frac{1}{1+x}$ aux points d'abscisse $\frac{j}{4}$ pour j variant de 0 à 4. Donner un majorant de la différence entre P et f en un point $x \in [0,1]$. Représenter graphiquement ce majorant. Calculer une majoration de l'erreur entre l'intégrale de f et l'intégrale de P sur $[0,1]$. En déduire un encadrement de $\ln(2)$.

Exercice 3
On reprend le calcul de $\int_0^1 \frac{dx}{1+x}$ mais en utilisant un polynôme interpolateur de degré 4 sur N subdivisions de $[0,1]$ (de pas $h = 1/N$). Déterminer une valeur de N telle que la valeur approchée de l'intégrale ainsi calculée soit proche à 10^{-8} près de $\ln(2)$. En déduire une valeur approchée à 10^{-8} de $\ln(2)$. Même question pour $\int_0^1 \frac{dx}{1+x^2}$ et $\pi/4$ (pour majorer la dérivée n-ième de $\frac{1}{1+x^2}$, on pourra utiliser une décomposition en éléments simples sur $\mathbb{C}$).

2 Algèbre linéaire

2.1 Instructions xcas

1. `charpoly(A,x)` : polynôme caractéristique
2. `det(A)` : déterminant d'une matrice A,
3. `dotprod(A,B)` : produit scalaire des deux vecteurs A et B,
4. `eigenvals(A)` : valeurs propres de la matrice A. Si vous voulez des valeurs numériques (et si A est à valeurs numériques) écrivez au moins un des coefficients de A comme nombre réel ou complexe.
5. `eigenvects(A)` : vecteurs propres de A,
6. `linsolve(A,b)` : résout l'équation linéaire $Ax = b$,
7. `ludecomp(A)` ou `lu(A)` : donne la décomposition LU de A.
8. `f :=(i,j)->(1/(i+j-1)) ;matrix(2,2,f)` : pour créer une matrice (ici de Hilbert).
9. `idn(n)` crée la matrice identité de taille n
10. `nrows(A), ncols(A)` : nombre de lignes/colonnes d'une matrice,
11. `normalize(v)` : renvoie $v/||v||$
12. `trace(A)` : trace de A
13. `transpose(A)` : transposée de A
14. `?linalg` pour la liste des commandes d'algèbre linéaire

2.2 Calculatrices

2.2.1 Commandes de la HP49

Pour entrer une matrices A vous pouvez utiliser le MatrixWriter (`MTRW` sur le clavier) ou directement le clavier (par exemple `[[1,2],[3,4]]`). On peut aussi créer une matrice dont l'élément $m_{i,j}$ est donné par une fonction $(i,j) \to m_{i,j}$ à l'aide de `LC2M`.

Commandes utiles :
- `EGV`, `EGVL` : vecteurs propres et valeurs propres d'une matrice
- `IDN(n)`, (resp. `RANM(n,m)`) : crée une matrice identité (resp. matrice aléatoire)
- `JORDAN` : réduction sous forme de Jordan d'une matrice
- `LU` : factorisation LU d'une matrice (numérique uniquement)
- `RREF` : réduction sous forme échelonnée

2.2.2 TI92/89, Casio Classpad

Pour entrer une matrice, on peut utiliser l'éditeur statistique ou directement au clavier (par exemple `[[1,2][3,4]]`).

Commandes utiles :
- `egv` (non disponible sur la 92) : valeurs propres/vecteurs propres. Attention, le calcul est numérique uniquement.
- `identity(n)` et `randmat(n,m)` pour créer des matrices identité et aléatoire
- `lu`, décomposition LU
- `rref` : réduction de Gauß
- Sur tous les modèles on peut bien sur calculer le polynome caractéristique avec `det(A-xI)` ou I désigne la matrice identité, puis factoriser avec `factor` et calculer les vecteurs propres à la main en résolvant le système.

2.3 Exercices

2.3.1 Méthode du pivot

1. Choisissez 5 vecteurs aléatoires exacts dans $\mathbb{R}^3$, trouvez deux relations linéaires indépendantes entre ces 5 vecteurs en calculant le noyau d'une application linéaire.
2. Déterminez le temps mis par votre logiciel pour résoudre un système linéaire de n équations à n inconnues à coefficients aléatoires numériques puis exacts de taille $n = 5, 10, 20, 40$. Commentez.
3. Écrire une fonction `gaussinv` qui prend en argument une matrice carrée, lui accolle à droite la matrice identité de même taille puis applique la fonction de réduction sous forme échelonnée, et renvoie la partie droite de la matrice réduite. Testez votre fonction avec une matrice aléatoire M de taille 3 à coefficients entiers, calculer en particulier `M * gaussinv(M)`.

2.3.2 Réduction des endomorphismes.

1. Polynôme caractéristique par interpolation
   Écrire une fonction qui calcule le polynôme caractéristique d'une matrice par interpolation de Lagrange (on calcule $\det(A - \lambda I)$ pour $\lambda = 0, ..., n$). Factorisez le polynôme, puis comparez avec les valeurs propres numériques de $A = B\ {}^tB$ (obtenues par la fonction de calcul de valeurs propres du logiciel) pour B une matrice aléatoire numérique de taille 10, 20, 40.
2. Écrire un programme mettant en oeuvre la méthode de la puissance. Utilisez ce programme pour trouver une valeur approchée de la valeur propre de norme maximale par la méthode de la puissance de $B\ {}^tB$ où B est une matrice aléatoire.
3. En utilisant la méthode des itérations inverses, trouvez la plus petite valeur propre de la matrice précédente.
4. (bonus)
   Pour trouver les autres valeurs propres/vecteurs propres, on élimine la valeur propre trouvée en remplaçant A par $A' = A - \lambda_1 w_1\ {}^tw_1$. Une fois une valeur propre de A' trouvée, on peut améliorer la précision en utilisant des itérations inverses sur la matrice A. Trouvez toutes les valeurs propres de la matrice :

$$D = \begin{pmatrix} 9 & -1 & 2 \\ -1 & 5 & -2 \\ 2 & -2 & -7 \end{pmatrix}$$