Examen du 19 mai 2015, de 14h à 17h.

Documents, calculatrices et ordinateurs ultraportables (netbooks) déconnectés du réseau autorisés.

Ce sujet comporte 2 pages. Barême donné à titre indicatif et non contractuel.

1. Méthode de la puissance et de Newton (9 points)

Soit P un polynôme unitaire de degré $d > 0$ à coefficients réels (ou complexes)

$$P = x^d + p_{d-1}x^{d-1} + \ldots + p_1 x + p_0$$

On suppose que P admet une seule racine de module maximal, que l'on notera z. On souhaite trouver une approximation de z par une méthode combinant la méthode de la puissance et de Newton. Par exemple pour fixer les idées et tester :

$$P = x^7 - 11x^4 + 5x - 55, \quad d = 7$$

(1) On construit une matrice compagnon M de P (commande `M:=tran(companion(P))` de Xcas) définie par :

$$M = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -p_0 & -p_1 & -p_2 & \ldots & -p_{d-2} & -p_{d-1} \end{pmatrix}$$

on admettra que le polynôme caractéristique de M est P. Expliquer pourquoi la méthode de la puissance appliquée à M permet de déterminer la racine z de P.

(2) En utilisant la structure particulière de la matrice M, expliquer comment on peut calculer efficacement $v_{n+1} = Mv_n$ à partir de la liste l des coefficients de P par ordre croissant (`l:=revlist(symb2poly(P,x))`) en $O(d)$ opérations.

(3) Étant donné un petit réel $\varepsilon > 0$, quel test d'arrêt vous parait le plus judicieux pour stopper le calcul des v_n ? Pourra-t-on certifier que l'estimation de la valeur propre obtenue sera proche à ε près de z ?

(4) Programmer l'algorithme en passant en paramètre la liste `l` et le petit réel `eps` (on pourra utiliser `v[1..d-1]` qui extrait les éléments de `v` d'indice `1` à `d-1` inclus, `dotprod(l,v)` effectue le produit scalaire de `l` et `v` en tronquant le vecteur le plus long lorsqu'ils ne sont pas de taille égale, et les instructions `append` et `normalize`)

(5) Dans les questions suivantes, sauf la dernière, on travaille sur l'exemple. Donner une estimation z_4 de z pour l'exemple pour ε=`1e-4` et z_8 pour ε=`1e-8`. Combien d'itérations sont-elles nécessaires pour obtenir cette estimation ? Pouvait-on s'y attendre ?

(6) Afin d'accélérer le calcul, on se propose d'utiliser la suite itérative u_n de la méthode de Newton en partant de $u_0 = z_4$. Donner l'expression de u_{n+1} en fonction de u_n. Quelle précision peut-on espérer en effectuant 2 itérations ?

(7) Calculer u_2. Proposer un raisonnement pour majorer l'erreur $|u_2 - z|$ et calculer ce majorant.

(8) Si on souhaite généraliser la méthode ci-dessus à un polynôme unitaire quelconque (sans faire l'hypothèse de l'existence d'une unique racine de module maximal), quels sont les obstacles prévisibles ? Comment peut-on les contourner ?

2. INTERPOLATION, INTÉGRATION (11 POINTS)

On souhaite expérimenter une méthode d'intégration où on approche la fonction à intégrer f sur $[-1,1]$ par son polynôme d'interpolation P aux n points d'abscisses $x_1,...,x_n$ les racines du n-ième polynôme de Tchebychev $T_n(\cos(x)) = \cos(nx)$. On pose donc :

$$I(f) = \sum_{j=1}^{n} \omega_j f(x_j) = \int_{-1}^{1} P(t)\, dt$$

On va d'abord calculer les ω_j puis expérimenter.

(1) Déterminer x_1, x_2, x_3 pour $n = 3$. En posant $f_1 = f(x_1), f_2 = f(x_2), f_3 = f(x_3)$, déterminer la valeur de P avec un logiciel de calcul formel, puis $\int_{-1}^{1} P(t)\, dt$, factoriser le résultat et en déduire $\omega_1, \omega_2, \omega_3$ pour $n = 3$. Que pensez-vous de cette méthode pour calculer les ω_j lorsque j est grand ?

(2) Rappeler pourquoi l'ordre de la méthode est au moins $n-1$, donner la valeur de $I(1), I(x), ..., I(x^{n-1})$, et en déduire un système linéaire vérifié par les ω_j.

(3) On rappelle qu'une matrice de Vandermonde est inversible si tous ces arguments sont distincts. En déduire que le système précédent permet de déterminer les ω_j.

(4) Faire le calcul par cette méthode pour $n = 3$ et vérifier les résultats de la première question.

(5) Calculer une valeur approchée des ω_j pour $n = 15$. Déterminer le conditionnement de la matrice du système. Que peut-on dire de la précision de la valeur des ω_j calculés en résolvant le système ?

(6) Quel est l'ordre de la méthode pour $n = 3$? Est-il meilleur pour $n = 4$?

(7) On se place maintenant sur une subdivision $[\alpha, \beta]$ où on transporte la méthode

$$I(f) = \sum_{j=1}^{n} \omega_j f(t_j), \quad t_j = \frac{\alpha+\beta}{2} + \frac{\beta-\alpha}{2} x_j$$

Donner une majoration de l'erreur sur une subdivision, puis sur $[a,b]$ si on le découpe en N subdivisions (de taille $h = (b-a)/N$)

(8) Écrire une fonction `I(f,a,b,N)` appliquant la méthode précédente pour $n = 3$.

(9) Appliquer la méthode précédente pour approcher $\int_0^1 e^{-t^2}\, dt$ à `1e-5` près lorsque $n = 3$: on déterminera une valeur de N qui assure la majoration de l'erreur.

(10) Comparer avec la valeur approchée de l'intégrale renvoyée par le logiciel, la majoration de l'erreur vous parait-elle fine ou grossière ? Si la majoration est grossière, pourrait-on l'améliorer ?