Partiel MM031, Master 1
Master Mathématique; UPMC
Les documents sont autorisés

F. Hecht/ Ionut Danaila

Mardi 1 Mars 2011

1 Partie Théorique (1 heure, F. Hecht)

le but est écrire une méthode numérique pour résoudre le problème non-linéaire suivant

Trouver u une fonction de $]a,b[\mapsto \mathbb{R}$ tel que $0<a<b$ et vérifiant les conditions (CL) : les valeurs sont données en a et b , telle que $u(a)=\alpha$ et $u(b)=\beta$. De plus cette fonction est le minimiseur de la fonctionnel J suivante

$$J(u)=2\pi\int_a^b r\sqrt{1+(u'(r))^2}dr \qquad (P)$$

On supposera, que ce problème est bien posé dans $H^1(]a,b[)$ c'est à dire, il a une unique u solution et qui ne dépend continument des 4 données a,b,α,β.

Q0 Monter que J est la surface définir par le graphe de la fonction $z=u(r)$ en coordonnée polaire avec donc $r=\sqrt{x^2+y^2}$, et pour $a<r<b$.

Q1 Calculer la différentiel $DJ(u)$ de J en u qui est une application linéaire de $H_0^1(]a,b[)$ dans $\mathbb{R}$.

Q2 Pourquoi u est solution de problème suivant

$$\forall v\in H_0^1((]a,b[),\quad 2\pi\int_a^b r\frac{u'v'}{\sqrt{1+(u'(r))^2}}dr=0$$

Q3 Montrer que la caténoïde $c(r)=acosh(r)$ est la solution du problème précédent pour $a=1$, $\alpha=0$, et $\beta=cosh(b)$, on rappel que $acosh'(r)=1/\sqrt{1-r^2}$.

Q4 On veut discrétiser le problème (P) avec des éléments finis P_1 Lagrange, soit N un entier strictement positif, on définit $h=(b-a)/N$, $q_i=a+hi$, pour $i=0,..,N$, les points du maillage et l'espace V_N des fonctions de $\mathcal{C}^0(]a,b[)$ et affines par morceaux sur $]q_i,q_{i+1}[$ pour $i=0,..,N-1$. Soit w_i la base de V_N telle que $w_i(q_j)=\delta_{ij}$, où δ_{ij} est le symbole de Kroneker.

Ecrire un algorithme formelle qui calcule la fonctionnel J pour une fonction v_N définir par

$$v_N=\sum_{i=0}^N v_i w_i,\quad \text{avec}\quad (v_i)_{i=0}^N\in\mathbb{R}^{N+1}$$

en fonction des valeurs v_i.

Q5 Ecrire une base de l'espace vectoriel $V_N\cap H_0^1(]a,b[)$.

Q6 On veut utiliser l'algorithme de minimisation à pas fixe qui s'écrit comme suit : Soit u_N^0 un élément de V_N donné vérifiant les conditions (CL), on calcul récursivement $u_N^{n+1}=u_N^n-\rho\nabla J(u_N^n)$ où ρ est une donné définie par l'utilisateur, et où $\nabla J(u)$ défini par partir d'un produit scalaire $(,)_N$ de V_N comme suit :

$$(\nabla J(u),v)=DJ(u)(v),$$

pour simplifier on prendra comme produit scalaire de $(u,v)_N=\sum_{i=0}^N u_i v_i$ de V_N où $v=\sum_{i=0}^N v_i w_i$ et où $u=\sum_{i=0}^N u_i w_i$.

Evaluer g_i^n, si on a on $\sum_{i=0}^N g_i^n w_i=\nabla J(u_N^n)$, et écrire complètement l'algorithme composante par composante.

2 Partie Pratique C++ (1 heure, I. Danaila)

Le but de cette partie est de programmer l'algorithme précédent,
On supposera que cette définition est active : `typedef double R;`

Q7 Ecrire une fonction `C++` de prototype `AireTroncDeCone(R ri,R ri1, R vi,Rvi1) ;` qui calcule l'aire d'un tronc de cône $c(r)$ défini pour ri < r < ri1 et où $c(r)$ la fonction affine telle que $c(\text{ri}) = \text{vi}$ et $c(\text{ri1}) = \text{vi1}$.

Q8 Ecrire une fonction `C++` de prototype `R J(R a, R b, int N,R *v) ;` qui évalue J pour la fonction $v = \sum_{i=0}^{N} v_i w_i$

Q9 Ecrire une fonction `C++` de prototype `R * GJ(R a, R b,int N,R *v,R *g) ;` qui évalue ∇J pour la fonction $v = \sum_{i=0}^{N} v_i w_i$ et qui retourne le pointeur g qui en entrée est fait pour stocker le vecteur ∇J.

Q10 Décrire une classe tableau `VN` , et les prototype les méthodes et des fonctions associer afin d'écrire l'algorithme de la question Q6 avec que la fonction suivante :

```
bool AlgoQ6(R a,R b,R alpha, R beta, VN & un,R eps);
{
   int N= un.N;
   Vn gn(N),
   un(0) = alpha;
   un(N) = beta;
   for(int n=0;n<100;++n)
     {
        un -= rho*GJ(a,b,N,un,gn);
        if(gn.norme_infini<eps) return true;
     }
     else return false;
}
```

Rappel : les règles de conversion (≪ cast ≫ en Anglais) d'un type T en A par défaut qui sont génèrées à partir d'un constructeur `A(T)` dans la classe A ou avec l'opérateur de conversion `operator (A) ()` dans la classe T, `operator (A) (T)` hors d'une classe. De plus il ne faut pas oublier que `C++` fait automatiquement un au plus un niveau de conversion pour trouver la bonne fonction ou le bon opérateur.

Q11 Bien sur il faut qu'il n'y ai aucun fuite de mémoire de cette classe, qu'elle est les code qu'il faut ajouter

Q12 Ecrire les programme principale qui test et valide votre algorithme.