

TP 15-16

13 et 14 Mars 2012

Projet 2 : implémenter en FreeFem++ le problème de minimisation de surface

- Q1 et Q2 : méthode du gradient à pas-simple
- Q3 et Q4 : méthode NLGC

Définition de J :

```
//the fonctionnel to minimize ..
func real J(real[int] & X)
{
  Vh u;
  u[]=ongamma[]? gh[]: X;// // inside we get X and on the border we get sur gh
  real jX= int2d(Th)(sqrt(1.+dx(u)*dx(u)+dy(u)*dy(u)));
  iter++;
  if(iter\%100==1)
  cout << iter << " J(X) = " << jX << endl;
  return jX;
}
```

Définition de DJ :

```
func real[int] DJ(real[int] & X)
{ // DJ return the gradient of J
  Vh u,du;
  u[]=ongamma[]? gh[]: X;// inside we get X and on the border we get sur gh
  varf vDJ(du,v) =int2d(Th)( (dx(v)*dx(u)+dy(v)*dy(u))/sqrt(1.+dx(u)*dx(u)+dy(u)*dy(u))
                          +on(1,du=0);
  du[]=vDJ(0,Vh); // du[][i] : is the derive partiel / variable i
  return du[];
}
```

Définition de NLGC :

```
NLCG(DJ,u[],eps=1.e-10,nbiter=200,precon=lap);
```

1 Compilation du Laplacien 2d :

- Télécharger l'archive RMN.zip et s8.zip
- Décompresser les archives dans le même répertoire.
- Dans le répertoire s8, taper la commande : "make". Exécuter le test *"/Lap2d nom_du_maillage"*.
- Visualiser les solutions avec Gnuplot

2 Compilation de l'exemple OpenGL :

- Télécharger OpenGL.zip.
- Décompresser dans le même répertoire que s8 et RMN.
- Modifier dans la makefile les librairies à inclure selon votre architecture :
LIBGL = ...
- Compiler et exécuter.