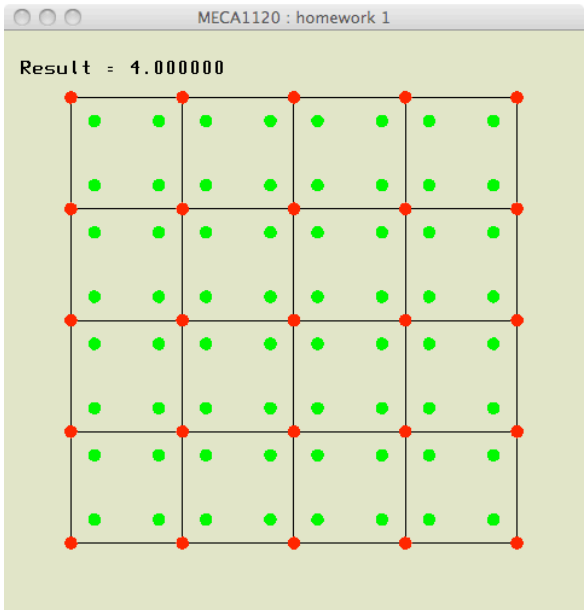
Finite elements 2013 for dummies : Gauss-Legendre

Il s'agit d'intégrer une fonction $f(x,y)$ sur un quadrilatère quelconque défini par ses quatre sommets dans le plan $(x,y) \in \mathbb{R}^2$, La règle de Gauss-Legendre consiste à estimer l'intégrale par la somme pondérée des valeurs de la fonction en quatre points judicieusement choisis.



$$\underbrace{\int_{\widehat{\Omega}} f(x,y)\ dx\ dy}_{I} \approx \underbrace{\sum_{k=1}^{4} w_k\ f(x_k,y_k)}_{I_h}$$

Sur le carré $\widehat{\Omega}$ dont les quatre sommets sont $(1,1)$, $(-1,1)$, $(-1,-1)$ et $(1,-1)$ dans le plan $(\xi,\eta) \in \mathbb{R}^2$, les poids et points d'intégration de la règle de Gauss-Legendre à quatre points sont donnés par :

ξ_k	η_k	w_k
$1/\sqrt{3}$	$1/\sqrt{3}$	1.0
$-1/\sqrt{3}$	$1/\sqrt{3}$	1.0
$-1/\sqrt{3}$	$-1/\sqrt{3}$	1.0
$1/\sqrt{3}$	$-1/\sqrt{3}$	1.0

Sur un quadrilatère quelconque Ω défini par la liste ordonnée des 4 sommets (X_1,Y_1), (X_2,Y_2), (X_3,Y_3) et (X_4,Y_4) cités en parcourant la frontière de manière anti-horlogique, on obtiendra le résultat escompté en effectuant un changement de variables donné par :

$$x(\xi,\eta) = \sum_{i=1}^{4} X_i\ \phi_i(\xi,\eta), \qquad y(\xi,\eta) = \sum_{i=1}^{4} Y_i\ \phi_i(\xi,\eta),$$

où les quatre fonctions de forme sont respectivement :

$$\begin{aligned} \phi_1(\xi,\eta) &= (1+\xi)(1+\eta)/4, \\ \phi_2(\xi,\eta) &= (1-\xi)(1+\eta)/4, \\ \phi_3(\xi,\eta) &= (1-\xi)(1-\eta)/4, \\ \phi_4(\xi,\eta) &= (1+\xi)(1-\eta)/4. \end{aligned}$$

Attention, il faut tenir compte du jacobien non constant de la transformation !

Plus précisément, on vous demande de :

1. Tout d'abord, écrire une fonction

```
double gaussJacobian(double x[4], double y[4], double xsi, double eta)
```

qui calcule le jacobien de la transformation en un point quelconque du quadrilatère. Si le quadrilatère est un carré, le jacobien est simplement le rapport des surfaces entre le carré considéré et le carré de référence. Ce calcul est évidemment indispensable pour le calcul numérique d'une intégrale par la règle de Gauss-Legendre sur un quadrilatère quelconque.

2. Ecrire la fonction

```
double gaussIntegrate(double x[4],double y[4],double(*f)(double,double))
```

qui estime l'intégrale de la fonction `f` sur le quadrilatère défini par ses quatre sommets. Si la fonction à intégrer est la fonction $f(x, y) = 1$, vous devez normalement obtenir la surface du quadrilatère. C'est un bon test de validation !

3. Ensuite, écrire une fonction

```
double gaussIntegrateRecursive(double x[4],double y[4],double(*f)(double,double),int n)
```

qui calcule la même intégrale divisant récursivement le quadrilatère en quatre sous-quadrilatères. Lorsque $n = 0$, on applique la règle de Gauss-Legendre sur le quadrilatère original, tandis qu'on considère 4, 16 ou 64 sous-quadrilatères lorsque $n = 1, 2$ ou 3. La décomposition d'un quadrilatère en quatre sous-quadrilatères s'effectue en ajoutant des noeuds sur le milieu des côtés. On peut réaliser une implémentation récursive et utiliser la première fonction définie.

4. Un morceau de code `main.c` vous est fourni pour tester votre fonction. Il faut donc compiler les deux fichiers séparément et ensuite effectuer une édition de lien entre les deux modules objets.

```
#include <stdio.h>
#include <math.h>

double gaussJacobian(double x[4], double y[4], double xsi, double eta);
double gaussIntegrate(double x[4], double y[4], double(*f)(double,double));
double gaussIntegrateRecursive(double x[4], double y[4], double(*f)(double,double), int n);

double fun(double x, double y)         { return cos(x) + y * y; }
double stupid(double x, double y)      { return 1.0; }

int main()
{
   double x[4] = { 2.0, -1.0, -1.0,  1.0} ;
   double y[4] = { 2.0,  1.0, -1.0, -1.0} ;
   int n;
   printf("Jacobian value in (0.0,0.0)  : %14.7e \n",gaussJacobian(x,y,0.0,0.0));
   printf("Jacobian value in (0.5,0.5)  : %14.7e \n",gaussJacobian(x,y,0.5,0.5));
   printf("Surface integration by Gauss-Legendre  : %14.7e \n",gaussIntegrate(x,y,stupid));
   printf("More funny integration         : %14.7e \n",gaussIntegrate(x,y,fun));
   for (n=0;  n <= 10; n++)
       printf("Recursive integration (n = %2d) : %14.7e \n",n,gaussIntegrateRecursive(x,y,fun,n));
   return 0;
}
```

5. Vos trois fonctions seront incluses dans un unique fichier `gauss.c`, sans y adjoindre le programme de test fourni ! Ce fichier devra être soumis via le web et la correction sera effectuée automatiquement. Il est donc indispensable de respecter strictement la signature des fonctions. Votre code devra être strictement conforme au langage `C` et il est fortement conseillé de bien vérifier que la compilation s'exécute correctement sur le serveur. Attention, de nombreuses extensions liées au `C++` sont souvent admises dans les environnements de développements standards, même lorsque votre fichier a le suffixe du langage C. Il faut aussi y être attentifs et bien vérifier (bien avant l'échéance finale !) que votre fichier se compile sur le serveur !

6. Si vous n'avez pas effectué une des étapes du devoir, il est judicieux d'inclure une fonction vide dans votre soumission pour au moins réussir le test de compilation.

7. Il est tout-à-fait inutile d'inclure des commentaires, car le correcteur automatique ne va pas les lire. Il n'est toutefois pas inutile d'inclure vos noms et prénoms dans l'entête du programme et de mentionner vos éventuelles sources d'inspiration ou de collaboration. Pour rappel, le plagiat est susceptible d'être sanctionné :-(