

PROJET LANGAGE C AVANCÉE : INTÉGRATION NUMÉRIQUE EN 2D

Version provisoire du 12 décembre 2013 à 15:30

**Travail individuel et personnel**

**Table des matières**

<b>1 Les maillages 2D de freeFEM++</b>	<b>2</b>
1.1 Triangulation	2
1.2 Algorithmique : structure de données associée aux maillages	2
1.3 Langage C : structure de données associée aux maillages	8
<b>2 Intégration numérique sur un triangle <math>T</math></b>	<b>9</b>
2.1 Changement de variables	9
2.2 Intégration numérique sur $\hat{T}$	10
2.2.1 Méthode $P_1$ -Lagrange	10
2.2.2 Méthodes de Gauss	10
2.3 Intégration numérique sur $\Omega_h$	10
<b>A Intégration exacte</b>	<b>11</b>
A.1 sur le carré unité $\Omega = [0, 1] \times [0, 1]$	11
A.2 Sur le disque unité $\mathcal{D}(0;1)$	11

L'objectif est de développer un code permettant le calcul d'intégrales sur des domaines quelconque de  $\Omega \subset \mathbb{R}^2$  :

$$\int_{\Omega} f(x, y) dx dy.$$

Pour cela on utilisera des méthodes d'intégration numérique composite utilisant des formules de Gauss sur un triangle de référence. Le domaine d'intégration sera maillé avec **freeFEM++** (voir [1]).

**Instructions**

- L'ensemble des codes sera envoyé par mail à l'adresse [cuvelier@math.univ-paris13.fr](mailto:cuvelier@math.univ-paris13.fr) sous la forme d'une archive compressée et **sans les fichiers de maillages** (extension **.msh**) avant le 19 janvier 2014 inclu.
- Les codes seront gérés à l'aide de Makefile.
- Un fichier texte nommé **README** sera fourni. Il contiendra les instructions pour compiler (avec un Makefile) et executer chacun des programmes demandés.

# 1 Les maillages 2D de freeFEM++

Voir [2] (pages 57 à 64) ou/et la documentation officielle : freefem++doc.pdf.

## 1.1 Triangulation

Nous découpons d'abord  $\Omega$  en petits éléments. Nous les choisissons ici de forme triangulaire, mais nous pourrions également envisager une décomposition en petits rectangles, par exemple. Les triangles doivent satisfaire à la

**Définition 1** On appelle triangulation de  $\Omega$ , une famille  $\mathcal{T}_h$  de triangles  $T_k$ ,  $k = 1, \dots, n_{me}$ , ayant les propriétés suivantes :

- (i) l'intersection entre deux triangles distincts est soit vide, soit réduite à une coté entier ou à un point ;
- (ii) tous les coins de la frontière  $\Gamma$  sont des sommets de triangles de  $\mathcal{T}_h$  ;
- (iii) réciproquement, soit

$$\Omega_h = \bigcup_{k=1}^{n_{me}} T_k \quad (1.1)$$

- (remarquer que  $\Omega_h$  est fermé) ; tous les coins de  $\Gamma_h = \partial\Omega_h$  doivent être sur  $\Gamma$  ;  
(iv) les triangles ne sont pas dégénérés, i.e. ils ne sont pas d'aire nulle. ■  
▲

**Remarque 2** nous avons

$$\Omega_h = \bigcup_{k=1}^{n_{me}} T_k \quad \text{et} \quad \bigcap_{k=1}^{n_{me}} \overset{\circ}{T}_k = \emptyset \quad (1.2) \quad \blacksquare$$

Les sommets de tous les triangles de  $\mathcal{T}_h$  sont les **sommets de la triangulation**. S'agissant d'une approximation par éléments finis  $P^1$ , nous verrons que ces sommets sont aussi les **nœuds de la triangulation**, i.e. les points où nous allons calculer la solution du problème discret. Il y a donc ici coïncidence entre les notions de sommets et de nœuds du maillage, coïncidence qui n'aurait pas lieu pour une approximation par éléments finis d'ordre quelconque.

La triangulation  $\mathcal{T}_h$  à  $n_{me}$  triangles,  $n_q$  sommets et  $n_f$  sommets situés sur  $\Gamma_h$ . Cette triangulation est indexée par  $h$  qui est la longueur du plus grand des cotés de tous ses triangles.

## 1.2 Algorithmique : structure de données associée aux maillages

On utilise 3 tableaux (un tableau de réels  $q$ , 4 tableaux d'entiers  $me$ ,  $ql$ ,  $be$  et  $bel$ ) et 2 entiers ( $n_q$  et  $n_{me}$ ) pour stocker les informations relatives au maillage ; il s'agit de :

nom	type	dimension	descriptif
$n_q$	entier	1	nombre total de nœuds du maillage ( <i>number of nodes</i> )
$n_{me}$	entier	1	nombre de triangles ( <i>number of mesh elements</i> )
$n_{be}$	entier	1	nombre d'arêtes sur le bord $\Gamma_h$ , ( <i>number of boundary elements</i> )
$q$	réels	$2 \times n_q$	$q(\alpha, j)$ est $\alpha$ -ème coordonnée du $j$ -ème sommet, $\alpha \in \{1, 2\}$ et $i \in \{1, \dots, n_q\}$ . Le $j$ -ème sommet sera aussi noté $q^j$ avec $q_x^j = q(1, j)$ et $q_y^j = q(2, j)$ ( <i>nodes</i> )
$me$	entier	$3 \times n_{me}$	$me(\beta, k)$ indice de stockage, dans le tableau $q$ , du $\beta$ -ème sommet du triangle d'indice $k$ , $\beta \in \{1, 2, 3\}$ et $k \in \{1, \dots, n_{me}\}$ . ( <i>mesh elements</i> )
$be$	entier	$2 \times n_{be}$	$be(\alpha, l)$ indice de stockage, dans le tableau $q$ , du $\alpha$ -ème sommet de l'arête d'indice $l$ appartenant au bord $\Gamma_h$ , $\alpha \in \{1, 2\}$ et $l \in \{1, \dots, n_{be}\}$ . ( <i>boundary elements</i> )
$ql$	entier	$1 \times n_q$	$ql(i)$ indique si le $i$ -ème sommet est un point strictement intérieur du domaine $\Omega$ ( $ql(i) = 0 \Leftrightarrow q^i \notin \Gamma_h$ sinon indique le numéro du bord, $i \in \{1, \dots, n_q\}$ ). ( <i>nodes label</i> )
$bel$	entier	$1 \times n_{be}$	$bel(l)$ indique le numéro du bord d'appartenance de l'arête $l$ , $l \in \{1, \dots, n_{be}\}$ . ( <i>boundary elements label</i> )
$mel$	entier	$1 \times n_{me}$	$mel(k)$ indique la region dans laquelle est le triangle $T_k$ , $k \in \{1, \dots, n_{me}\}$ . ( <i>mesh elements label</i> )

Pour illustrer cette structure, nous allons générer à l'aide de **freeFEM++** un maillage du disque unité où la frontière  $\Gamma$  sera réunion des quatre quarts de cercle (voir figure 1)

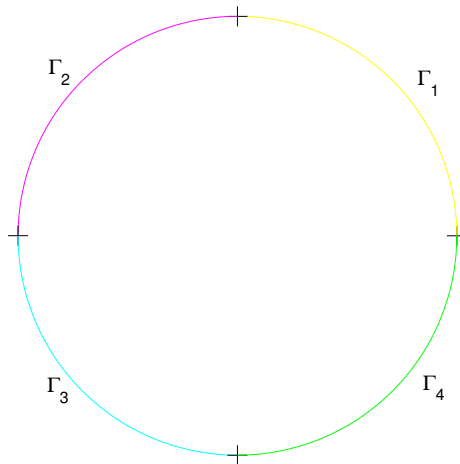


FIGURE 1 – Disque à mailler avec 4 bords

Voici le code **freeFEM++** :

---

```

real R=1.0; // rayon du disque
int N=3; // nombre de discrétisation pour un quart de cercle
border gamma1(t=0, pi/2) // Bord  $\Gamma_1$ 
    {x=R*cos(t); y=R*sin(t); label=1;};
border gamma2(t=pi/2, pi) // Bord  $\Gamma_2$ 
    {x=R*cos(t); y=R*sin(t); label=2;};
border gamma3(t=pi, 3*pi/2) // Bord  $\Gamma_3$ 
    {x=R*cos(t); y=R*sin(t); label=3;};
border gamma4(t=3*pi/2, 2*pi) // Bord  $\Gamma_4$ 
    {x=R*cos(t); y=R*sin(t); label=4;};

mesh Th = buildmesh ( gamma1(N)+gamma2(N)
    +gamma3(N)+gamma4(N) );

plot(Th, ps="disque4.eps", wait=true);

savemesh(Th, "disque4-"+R+"-"+N+".msh");

```

---

Pour obtenir un maillage de petite taille, nous avons délibérément choisi un petit nombre de points sur le bord : chaque quart de cercle sera discrétisé avec  $N = 3$  points.

Voici le fichier `disque4-1-3.msh` généré :

---

```

20 26 12 // nq, nme, nbe
-0.866025403784 0.5 2 // qx1, qy1, ql(1)
-0.579246826544 0.262259524467 0 // qx2, qy2, ql(2)
-0.5 0.866025403784 2 // qx3, qy3, ql(3)
-1 1.22460635382e-16 3 // qx4, qy4, ql(4)
6.12303176911e-17 1 2 // qx5, qy5, ql(5)
0.5 0.866025403784 1 // qx6, qy6, ql(6)
-0.158493650014 0.524519052007 0 // qx7, qy7, ql(7)
-0.433960984027 -0.11887024084 0 // qx8, qy8, ql(8)
-0.866025403784 -0.5 3 // qx9, qy9, ql(9)
0.183012699972 0.183012699972 0 // qx10, qy10, ql(10)
0.866025403784 0.5 1 // qx11, qy11, ql(11)
-0.0528312207692 -0.158493650014 0 // qx12, qy12, ql(12)
-0.288675135364 -0.5 0 // qx13, qy13, ql(13)
-0.5 -0.866025403784 3 // qx14, qy14, ql(14)
0.524519052007 -0.158493650014 0 // qx15, qy15, ql(15)
1 -2.44921270764e-16 4 // qx16, qy16, ql(16)
0.288675133315 -0.5 0 // qx17, qy17, ql(17)
-1.83690953073e-16 -1 4 // qx18, qy18, ql(18)
0.866025403784 -0.5 4 // qx19, qy19, ql(19)
0.5 -0.866025403784 4 // qx20, qy20, ql(20)
17 20 19 0 // me(1,1), me(2,1), me(3,1), mel(1)
15 19 16 0 // me(1,2), me(2,2), me(3,2), mel(1)
10 12 15 0 // me(1,3), me(2,3), me(3,3), mel(1)
14 13 9 0 // me(1,4), me(2,4), me(3,4), mel(1)
14 18 13 0 // me(1,5), me(2,5), me(3,5), mel(1)
18 20 17 0 // me(1,6), me(2,6), me(3,6), mel(1)
18 17 13 0 // me(1,7), me(2,7), me(3,7), mel(1)
7 8 12 0 // me(1,8), me(2,8), me(3,8), mel(1)
7 10 6 0 // me(1,9), me(2,9), me(3,9), mel(1)
3 7 5 0 // me(1,10), me(2,10), me(3,10), mel(10)
3 1 2 0 // me(1,11), me(2,11), me(3,11), mel(11)
10 11 6 0 // me(1,12), me(2,12), me(3,12), mel(12)
19 15 17 0 // me(1,13), me(2,13), me(3,13), mel(13)
16 11 15 0 // me(1,14), me(2,14), me(3,14), mel(14)

```

```

5 7 6 0 // me(1,15), me(2,15), me(3,15), mel(15)
8 9 13 0 // me(1,16), me(2,16), me(3,16), mel(16)
12 8 13 0 // me(1,17), me(2,17), me(3,17), mel(17)
2 4 8 0 // me(1,18), me(2,18), me(3,18), mel(18)
12 10 7 0 // me(1,19), me(2,19), me(3,19), mel(19)
2 8 7 0 // me(1,20), me(2,20), me(3,20), mel(20)
3 2 7 0 // me(1,21), me(2,21), me(3,21), mel(21)
4 9 8 0 // me(1,22), me(2,22), me(3,22), mel(22)
13 17 12 0 // me(1,23), me(2,23), me(3,23), mel(23)
4 2 1 0 // me(1,24), me(2,24), me(3,24), mel(24)
17 15 12 0 // me(1,25), me(2,25), me(3,25), mel(25)
15 11 10 0 // me(1,26), me(2,26), me(3,26), mel(26)
18 20 4 // be(1,1), be(2,1), bel(1)
20 19 4 // be(1,2), be(2,2), bel(2)
19 16 4 // be(1,3), be(2,3), bel(3)
4 9 3 // be(1,4), be(2,4), bel(4)
9 14 3 // be(1,5), be(2,5), bel(5)
14 18 3 // be(1,6), be(2,6), bel(6)
5 3 2 // be(1,7), be(2,7), bel(7)
3 1 2 // be(1,8), be(2,8), bel(8)
1 4 2 // be(1,9), be(2,9), bel(9)
16 11 1 // be(1,10), be(2,10), bel(10)
11 6 1 // be(1,11), be(2,11), bel(11)
6 5 1 // be(1,12), be(2,12), bel(12)

```

La représentation classique du maillage est donnée en Fig. 2

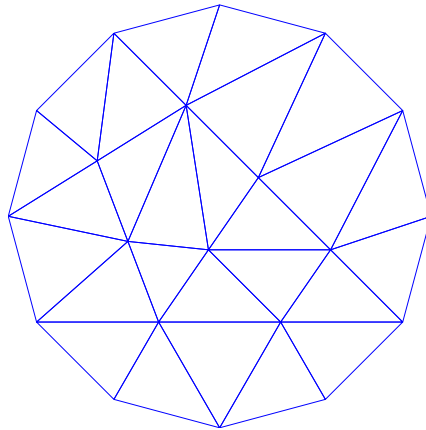


FIGURE 2 – Maillage obtenu avec **freeFEM++**

Après lecture, les tableaux sont :

```

q      : [2x20 double]
me     : [3x26 int32]
ql     : [2 0 2 3 2 1 0 0 3 0 1 0 0 3 0 4 0 4 4 4]
mel    : [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
be     : [2x12 double]
bel    : [4 4 4 3 3 3 2 2 2 1 1 1]
nq   : 20
nme  : 26
nbe  : 12

```

où les tableaux q, me et be sont respectivement donnés par

1	2	3	...	17	18	19	20
-0.8660	-0.5792	-0.5000	...	0.2887	-0.0000	0.8660	0.5000
0.5000	0.2623	0.8660	...	-0.5000	-1.0000	-0.5000	-0.8660

TABLE 1 – Tableau q associé au maillage disque4-1-3.msh

1	2	3	4	...	24	25	26
17	15	10	14	...	4	17	15
20	19	12	13	...	2	15	11
19	16	15	9	...	1	12	10

TABLE 2 – Tableau me associé au maillage disque4-1-3.msh

1	2	3	4	5	6	7	8	9	10	11	12
18	20	19	4	9	14	5	3	1	16	11	6
20	19	16	9	14	18	3	1	4	11	6	5

TABLE 3 – Tableau be associé au maillage disque4-1-3.msh

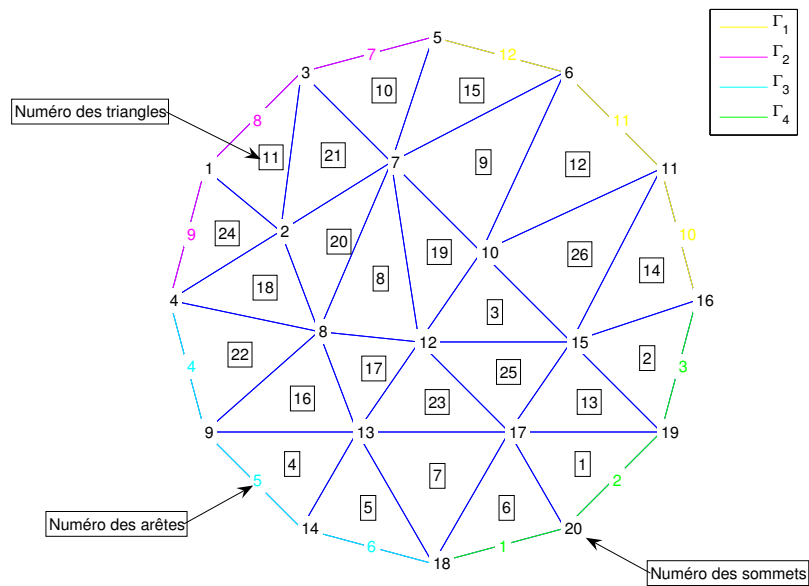


FIGURE 3 – Maillage avec numérotation

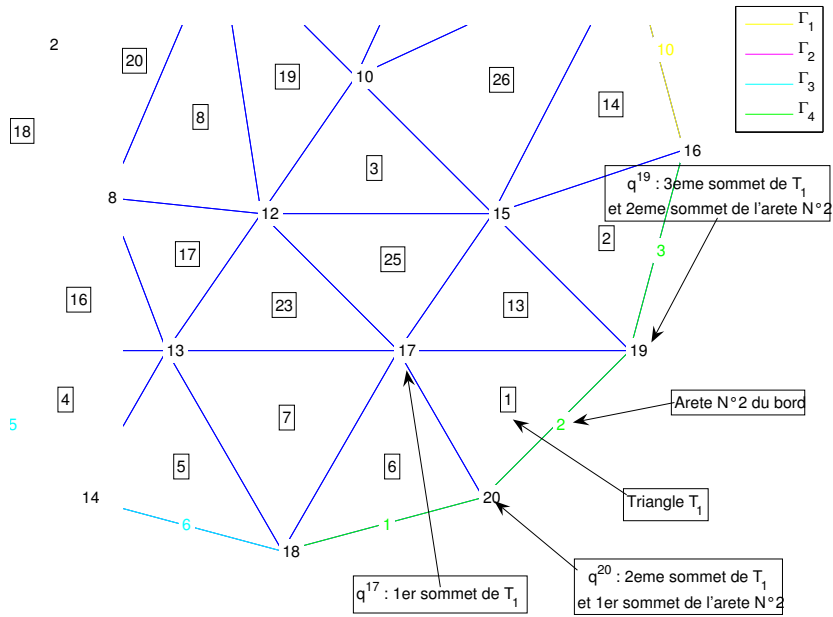


FIGURE 4 – Maillage avec numérotation : triangle  $T_1$

---

**Algorithme 1** Lecture d'un maillage FreeFEM++

---

**Données :** Filename : nom du fichier de maillage généré par FreeFEM++

**Résultat :**  $n_q$  :  
 $n_{me}$  :  
 $n_{be}$  :  
 $q$  :  
 $ql$  :  
 $me$  :  
 $mel$  :  
 $be$  :  
 $bel$  :

```

1: Fonction [  $n_q, n_{me}, n_{be}, q, ql,$   

    $me, mel, be, bel$  ]  $\leftarrow$  READMESHFREEFEM(Filename)
2:    $fid =$ OUVRIR(Filename,"r")
3:   [  $n_q, n_{me}, n_{be}$  ]  $\leftarrow$ LIRE(fid,"%d %d %d")
4:   Pour  $i \leftarrow 1$  à  $n_q$  faire
5:     [  $q(1, i), q(2, i), ql(i)$  ]  $\leftarrow$ LIRE(fid,"%lf %lf %d")
6:   Fin Pour
7:   Pour  $k \leftarrow 1$  à  $n_{me}$  faire
8:     [  $me(1, k), me(2, k), me(3, k), mel(k)$  ]  $\leftarrow$ LIRE(fid,"%d %d %d %d")
9:   Fin Pour
10:  Pour  $l \leftarrow 1$  à  $n_{be}$  faire
11:    [  $be(1, l), be(2, l), bel(l)$  ]  $\leftarrow$ LIRE(fid,"%d %d %d")
12:  Fin Pour
13:  FERMER(fid)
14: Fin Fonction

```

---

### 1.3 Langage C : structure de données associée aux maillages

La structure de données associée aux maillages est imposée. Elle est donnée dans le fichier header `mesh.h` fourni et rappelée ici :

Listing 3 – Structure Mesh

```
9 typedef struct mesh{
10 int nq; /* Nombre de sommets */
11 int nme; /* Nombre de triangles */
12 int nbe; /* Nombre d'arêtes sur le bord */
13 double *q[2]; /* Tableau des points 2*nq : q[il][j] composante il du sommet j */
14 int *me[3]; /* Tableau des triangles 3*nme
15             me[jl][k] indice dans le tableau q du jl-ième sommet du triangle */
16 int *ql; /* 1*nq (0 pour point interieur) */
17 int *mel; /* 1*nme ("numero" de la region) */
18 int *bel; /* 1*nbe ("numero" du bord) */
19 int *be[2]; /* Tableau des arêtes (2*be)
20             be[il][j] indice dans le tableau q du il-ième sommet de l'arête j */
21 double *areas; /* Aire des triangles */
22 double h; /* maximum des longueurs des arêtes de tous les triangles */
23 } Mesh;
```

Dans le même fichier header, un ensemble de prototypes de fonctions est donné :

Listing 4 – Prototypes du fichier `mesh.h`

```
26 Mesh ReadMesh(char * nom_fichier);
27 double *MeshElementsAreas(Mesh M);
28 double MeshElementsMaxEdgesLengths(Mesh M);
29 void InitMesh(Mesh *pM);
30 Mesh ReadAndInitMesh(char * nom_fichier);
31 void PrintMesh(Mesh M);
32 void FreeMesh(Mesh M);
```

- `Mesh ReadMesh(char *nom_fichier);` :  
Retourne une structure de type `Mesh` contenant le maillage **freeFEM++** à partir du fichier nommé par `nom_fichier`. Les champs `areas` et `h` de la structure ne seront ni initialisés ni calculés par cette fonction.

Listing 5 – `ReadMesh` - exemple

```
Mesh Th;
Th=ReadMesh("disque4-1-3.msh");
```

- `double *MeshElementsAreas(Mesh M);` :  
Retourne un tableau de dimension `M.nme` contenant les aires des triangles du maillage `M`.
- `double MeshElementsMaxEdgesLengths(Mesh M);` :  
Retourne le maximum des longueurs des arêtes de tous les triangles du maillage `M`.
- `void InitMesh(Mesh *pM);` :  
Initialise et calcule les champs `areas` et `h` du maillage `*pM`.
- `Mesh ReadAndInitMesh(char * nom_fichier);` :  
Retourne une structure de type `Mesh` contenant le maillage **freeFEM++** à partir du fichier nommé par `nom_fichier`. Les champs `areas` et `h` de la structure seront initialisés et calculés dans cette fonction.

Listing 6 – `ReadAndInitMesh` - exemple

```
Mesh Th;
Th=ReadAndInitMesh("disque4-1-3.msh");
```

- `void PrintMesh(Mesh M);` :  
Affiche (de manière pertinente) les informations contenu dans `M`.

- **void** FreeMesh(Mesh M); :  
Libère les différentes *zones* mémoires allouées lors de la création de M.

— **A faire ...** —

1. Ecrire le fichier `mesh.c` contenant l'ensemble des définitions des fonctions dont les prototypes sont donnés dans le fichier `mesh.h`.
2. Compiler et exécuter le programme donné dans `essai01.c`.
3. Ecrire le fichier `essai02.c` contenant un programme permettant de calculer l'aire totale du domaine  $\Omega_h$  fourni par un maillage.

## 2 Intégration numérique sur un triangle $T$

Soit  $T$  un triangle de  $\mathbb{R}^2$  défini par ses trois sommets  $q^1$ ,  $q^2$  et  $q^3$ . Soit  $f$  une fonction définie de  $T$  à valeurs dans  $\mathbb{R}$ . On souhaite calculer numériquement  $\int_T f(q)dq$ .

### 2.1 Changement de variables

On note  $\hat{T}$  le triangle de référence défini par ses trois sommets

$$\hat{q}^1 = (0, 0)^t, \quad \hat{q}^2 = (0, 1)^t \quad \text{and} \quad \hat{q}^3 = (1, 0)^t$$

et  $\hat{q} = \begin{pmatrix} \xi \\ \eta \end{pmatrix} \in \hat{T}$ .

Soit  $q = \begin{pmatrix} x \\ y \end{pmatrix} \in T$ . On note  $\mathcal{F}_T$  la transformation affine (qui est une bijection) de  $\hat{T}$  dans  $T$  définie par

$$q = \mathcal{F}_T(\hat{q}) = q^1 + \xi(q^2 - q^1) + \eta(q^3 - q^1).$$

On en déduit l'écriture matricielle de  $\mathcal{F}_T$  :

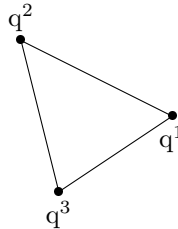
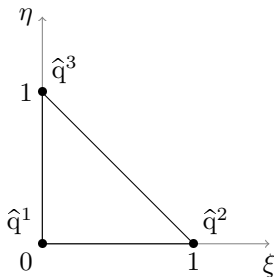
$$\mathcal{F}_T(\hat{q}) = \mathbb{B}_T \hat{q} + q^1, \quad \text{avec} \quad \mathbb{B}_T = \begin{pmatrix} q_x^2 - q_x^1 & q_x^3 - q_x^1 \\ q_y^2 - q_y^1 & q_y^3 - q_y^1 \end{pmatrix}.$$

En appliquant ce changement de variables à l'intégrale on obtient

$$\int_T f(q)dq = \int_{\hat{T}} f(\mathcal{F}_T(\hat{q})) |\det \mathbb{B}_T| d\hat{q}$$

En notant que  $|\det \mathbb{B}_T| = 2|T|$  avec  $|T|$  aire du triangle, on abouti à

$$\int_T f(q)dq = 2|T| \int_{\hat{T}} f(\mathcal{F}_T(\hat{q})) d\hat{q}. \tag{2.1}$$



## 2.2 Intégration numérique sur $\hat{T}$

Nous allons approcher  $\int_{\hat{T}} g(\hat{q})d\hat{q}$  par

$$\int_{\hat{T}} g(\hat{q})d\hat{q} \approx \sum_{i=1}^n w_i g(\bar{q}^i) \quad (2.2)$$

où  $\bar{q}^i \in \hat{T}$  sont les  $n$  points d'intégrations (distincts) et  $w_i \in \mathbb{R}$  les poids associés. Les valeurs des  $\bar{q}^i$  et des  $w_i$  seront données en fonction de la formule d'intégration numérique choisie.

On dira que la formule (2.2) est d'ordre  $p$  si elle est exacte pour les polynômes de degrés inférieur ou égal à  $p$ .

En combinant les formules (2.1) et (2.2), on obtient

$$\int_T f(q)dq \approx 2|T| \sum_{i=1}^n w_i f(\mathcal{F}_T(\bar{q}^i)). \quad (2.3)$$

### 2.2.1 Méthode $P_1$ -Lagrange

Pour cette méthode, on a  $n = 3$ , et  $\forall i \in \llbracket 1, n \rrbracket$ ,  $\bar{q}^i = \hat{q}^i$ ,  $w_i = \frac{1}{6}$ . Ce qui donne

$$\int_{\hat{T}} g(\hat{q}) \approx \frac{1}{6}(g(0,0) + g(1,0) + g(0,1)) \quad (2.4)$$

Cette formule est d'ordre 1.

### 2.2.2 Méthodes de Gauss

Sur le même principe, on résume dans le tableau suivant trois méthodes de Gauss respectivement à 1, 2 et 3 points :

$n$	$\bar{q}^i$	$w_i$	order
1	$\bar{q}^1 = (1/3, 1/3)$	$w_1 = 1/2$	1
3	$\bar{q}^1 = (1/6, 1/6)$	$w_1 = 1/6$	2
	$\bar{q}^2 = (2/3, 1/6)$	$w_2 = 1/6$	
	$\bar{q}^3 = (1/6, 2/3)$	$w_3 = 1/6$	
4	$\bar{q}^1 = (1/3, 1/3)$	$w_1 = -27/96$	3
	$\bar{q}^2 = (1/5, 1/5)$	$w_2 = 25/96$	
	$\bar{q}^3 = (3/5, 1/5)$	$w_3 = 25/96$	
	$\bar{q}^4 = (1/5, 3/5)$	$w_4 = 25/96$	

## 2.3 Intégration numérique sur $\Omega_h$

De part les propriétés du maillage, on a

$$\int_{\Omega_h} f(q)dq = \sum_{k=1}^{n_{me}} \int_{T_k} f(q)dq.$$

On note  $\mathcal{F}_k$  la transformation affine permettant de passer de  $\hat{T}$  à  $T_k$  (voir section 2.1). On a alors, en utilisant (2.1),

$$\int_{\Omega_h} f(q)dq = \sum_{k=1}^{n_{me}} 2|T_k| \int_{\hat{T}} f(\mathcal{F}_k(\hat{q}))d\hat{q} \quad (2.5)$$

En choisissant une méthode numérique d'intégration d'ordre  $p$  pour l'approximation des intégrales sur  $\hat{T}$ , la formule précédente reste d'ordre  $p$  si  $\Omega = \Omega_h$ .

— A faire ... —

1. Ecrire les fichiers `quad.c` et `quad.h` contenant un ensemble de fonctions cohérentes permettant de calculer les intégrales de fonctions sur un maillage donné par les formules composites  $P_1$ -Lagrange et de Gauss avec  $n = 1, 3$  et  $4$ .
2. Ecrire un programme permettant de vérifier l'ordre des différentes méthodes.
3. Ecrire un programme permettant de comparer les différentes méthodes.

## A Intégration exacte

### A.1 sur le carré unité $\Omega = [0, 1] \times [0, 1]$ .

	$f(x, y)$	$\int_0^1 \int_0^1 f(x, y) dx dy$
1	$x + y$	1
2	$2x + 3y - 2$	$\frac{1}{2}$
3	$x^2 + 2y^2 - 2y - 1$	-1
4	$3x^3 - 2x^2y - y^2 + y - 1$	$-\frac{5}{12}$
5	$y^4 + x^3 + 4xy$	$\frac{29}{20}$
6	$3x^5 + 2x^2y + y^3 - y^2 + 1$	$\frac{7}{4}$
7	$x^6 - 3x^3 + 4y^2$	$\frac{61}{84}$
8	$\cos(\pi(x - y))$	$\frac{4}{\pi^2}$

### A.2 Sur le disque unité $\mathcal{D}(0; 1)$

	$f(x, y)$	$\int_{\mathcal{D}(0;1)} f(x, y) dx dy$
1	$x + y$	0
2	$2x + 3y - 2$	$-2\pi$
3	$x^2 + 2y^2 - 2y - 1$	$-\frac{1}{4}\pi$
4	$3x^3 - 2x^2y - y^2 + y - 1$	$-\frac{5}{4}\pi$
5	$y^4 + x^3 + 4xy$	$\frac{1}{8}\pi$
6	$3x^5 + 2x^2y + y^3 - y^2 + 1$	$\frac{3}{4}\pi$
7	$x^6 - 3x^3 + 4y^2$	$\frac{69}{64}\pi$
8	$\cos\left(\frac{1}{2}\pi(x^2 + y^2)\right)$	2

## Références

- [1] FREDERIC HECHT, *freeFEM++*, <http://www.freefem.org/ff++/index.htm>
- [2] FRANÇOIS CUVELIER, *Méthodes des éléments finis. De la théorie à la programmation.*, <http://www.math.univ-paris13.fr/~cuvelier/docs/poly/polyFEM2D.pdf>