

TPs EDP

Travaux Pratiques N° 1

Algorithmique : génération de maillages

Version du 24 septembre 2025

Table des matières

1	Introduction	1
2	Initiation	3
2.1	Archive fournie	3
2.2	Barycentres	5
2.3	Calcul d'aires	5
3	Génération de maillages	5
3.1	Maillage du carré unité	5
3.2	Rectangle $[a, b] \times [c, d]$	6
3.3	Maillage du triangle de référence	7
3.4	Maillage d'un triangle	8
4	Namespace <code>sMesh</code>	9

1 Introduction

Soit Ω un domaine borné de \mathbb{R}^2 .

♥ Definition 1.1

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

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

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

- (remarquer que Ω_h est fermé) ; tous les coins de $\Gamma_h = \partial\Omega_h$ doivent être sur Γ ;
- (iv) les triangles ne sont pas dégénérés, *ie.* ils ne sont pas d'aire nulle.

a. n_{me} number of mesh elements

Remarque 1.1

nous avons

$$\Omega_h = \bigcup_{k=1}^{n_{me}} T_k \text{ et } \bigcap_{k=1}^{n_{me}} \overset{\circ}{T}_k = \emptyset \quad (2)$$

En Figure 1, deux exemples de triangulation sont représentés.

Pour stocker les informations (minimales) relatives à un maillage, on utilise les tableaux ***q*** et ***me*** respectivement tableau des sommets/points et tableau de connectivité :

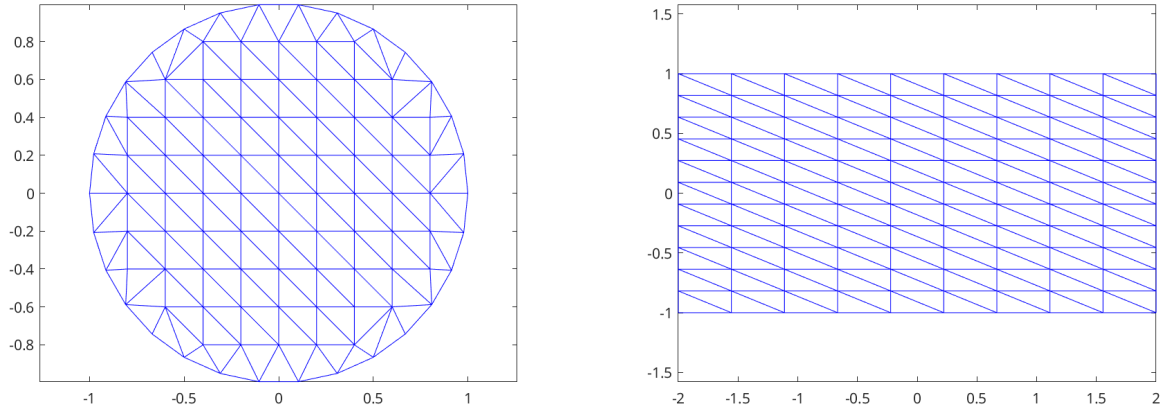
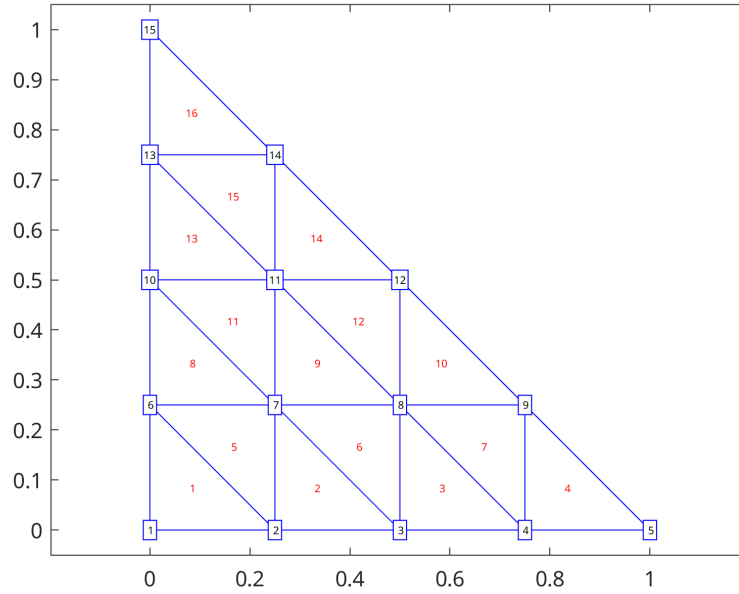


FIGURE 1 – Maillages triangulaires du disque unité (à gauche) et du rectangle $[-2, 2] \times [-1, 1]$ (à droite).

nom	type	dimension	descriptif
n_q	entier	1	nombre total de noeuds (sommets) du maillage
n_{me}	entier	1	nombre de triangles
q	réels	$2 \times n_q$	$q(il, i)$ est la il -ème coordonnée du i -ème sommet, $il \in \{1, 2\}$ et $i \in \{1, \dots, n_q\}$. Le i -ème sommet sera aussi noté $\mathbf{q}^i = (\mathbf{q}_x^i, \mathbf{q}_y^i)$ avec $\mathbf{q}_x^i = q(1, i)$ et $\mathbf{q}_y^i = q(2, i)$
me	entier	$3 \times n_{me}$	$me(jl, k)$ indice de stockage, dans le tableau q , du jl -ème sommet du triangle d'indice k , $jl \in \{1, 2, 3\}$ et $k \in \{1, \dots, n_{me}\}$. Pour tout triangle la numérotation des points est dans le sens direct . $q(:, me(1, k))$ est le 1er sommet du k -ème triangle, $q(:, me(2, k))$ est le 2ème sommet, ...

Voici sur un exemple le contenu des tableaux q et me sur un maillage du triangle de référence/unité :



Les nombres rouges (non encadrés) sont les numéros des triangles (relativement au tableau [me](#)) et les nombres bleus (encadrés) sont les numéros des points (relativement au tableau [q](#)). Le contenu des tableaux [q](#) et [me](#) sous Matlab/Octave correspond alors à :

```
q =
Columns 1 through 7
    0    0.2500    0.5000    0.7500    1.0000     0    0.2500
    0     0         0         0         0    0.2500    0.2500

Columns 8 through 14
    0.5000    0.7500     0    0.2500    0.5000     0    0.2500
    0.2500    0.2500    0.5000    0.5000    0.5000    0.7500    0.7500

Column 15
     0
    1.0000

me =
Columns 1 through 13
     1     2     3     4     2     3     4     6     7     8     7     8    10
     2     3     4     5     7     8     9     7     8     9    11    12    11
     6     7     8     9     6     7     8    10    11    12    10    11    13

Columns 14 through 16
    11    11    13
    12    14    14
    14    13    15
```

Ces deux tableaux sont fournis sous la forme d'un fichier *.mat* contenu dans une archive.

2 Initiation

2.1 Archive fournie

L'archive fournie [TP1.zip](#) ou [TP1.tar.gz](#) contient, [~/TPsAlgoEDPs/TP1](#)). Voici, pour notre exemple, l'arborescence obtenue :

entre autres, plusieurs fichiers *.mat* dans le répertoire *mat*. Il faut décompresser cette archive dans un nouveau répertoire de travail (par ex, sous Linux :

```

TPsAlgoEDPs
├── TP1
│   ├── +demos
│   │   └── demo00.m
│   ├── +utils
│   │   └── inputParserUnmatched2Cell.m
│   └── mat
│       ├── rectmesh_5_6.mat
│       ├── unittrimesh_5.mat
│       ├── rectmesh_10_12.mat
│       └── trianglemesh_5.mat

```

```

├── diskmesh_30.mat
├── unitsquaremesh_5_6.mat
├── diskmeshdel_polar_20.mat
├── diskmeshdel_grid_20.mat
└── +graphics
    ├── ElementsNumbers.m
    ├── plotmesh.m
    ├── VerticesNumbers.m
    └── plotmeshnumbers.m

```

Des fichiers contenant des maillages, donnés par leurs tableaux de points et de connectivité, sont stockés dans le répertoire `mat`. Chacun des fichiers de ce répertoire correspond à un maillage contenant un tableau des points `q` et un tableau de connectivité `me`. Pour lire le fichier `mat/unitsquaremesh_5_6.mat` sous Matlab/Octave, on peut utiliser la commande `load` pour initialiser les tableaux `q` et `me` :

Si le répertoire de travail contient le répertoire `mat`, la commande

```
load( fullfile('mat','unitsquaremesh_5_6.mat'))
```

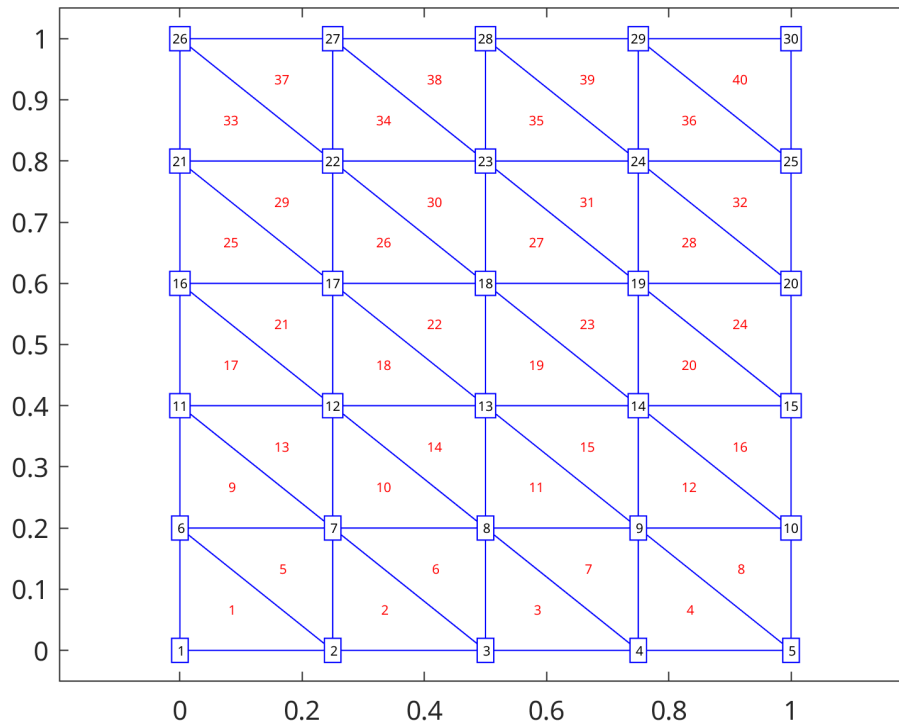
permet de récupérer les tableaux `q` et `me`.

L'archive fournie contient aussi des fonctions permettant de représenter un maillage, d'afficher les numéros des sommets, des triangles... Les deux fonctions principales, du *namespace* `graphics`, sont

- `graphics.plotmesh(q,me)` : représentation d'un maillage donné par `q` et `me`,
- `graphics.plotmeshnumbers(q,me)` (non utilisable pour le moment) : représentation d'un maillage donné par `q` et `me`, avec les numéros des sommets et des triangles.

Voir Listing 1 pour un exemple d'utilisation (script `demos.demo00`).

Attention, la fonction `graphics.plotmeshnumbers(q,me)` utilise la fonction `graphics.ElementsNumbers` qui appelle la fonction `meshes.barycenters` qui **n'est pas fournie** !. Elle sera à écrire en question 1.



Listing 1 – Exemple d'utilisation de la fonction `graphics.plotmeshnumbers`

```

load( fullfile('mat','unitsquaremesh_5_6.mat')) % loading q and me arrays
graphics.plotmeshnumbers(q,me)
axis equal

```

2.2 Barycentres

Q. 1 ★★

Ecrire une fonction `ba=meshes.barycenters(q,me)` (fonction `barycenters` du namespace `meshes`) retournant le tableau `ba` de dimension $2 \times n_{\text{me}}$ tel que `ba(:,k)` soit le barycentre du k -ème triangle spécifié par le tableau des sommets `q` et le tableau de connectivité `me`. Pour cela créer et utiliser une fonction `triangle.barycenter` retournant le barycentre d'un triangle.

2.3 Calcul d'aires

L'objectif ici est de calculer les aires de tous les triangles d'un maillage donné par ses tableaux de sommets et de connectivité.

Remarque 2.1

Soit \hat{K} le triangle de référence de sommets $\hat{\mathbf{q}}^0 = (0, 0)$, $\hat{\mathbf{q}}^1 = (1, 0)$, $\hat{\mathbf{q}}^2 = (0, 1)$, et K un triangle non dégénéré de sommets $(\mathbf{q}^0, \mathbf{q}^1, \mathbf{q}^2)$. La fonction \mathcal{F}_K définie par

$$\begin{aligned} \mathcal{F}_K : \hat{K} \subset \mathbb{R}^2 &\longrightarrow K \subset \mathbb{R}^2 \\ \hat{\mathbf{q}} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} &\longmapsto \mathbf{q} = \mathbf{q}^0 + (\mathbf{q}^1 - \mathbf{q}^0)\hat{x} + (\mathbf{q}^2 - \mathbf{q}^0)\hat{y} \end{aligned} \quad (3)$$

est une bijection avec $\mathcal{F}_K(\hat{\mathbf{q}}^i) = \mathbf{q}^i$, $\forall i \in \llbracket 0, 2 \rrbracket$.

Soit \mathbf{q}^0 , \mathbf{q}^1 et \mathbf{q}^2 les trois sommets d'un triangle K non-dégénéré. On note $\mathbb{A}_K = (\mathbf{q}^1 - \mathbf{q}^0 \mid \mathbf{q}^2 - \mathbf{q}^0)$ la matrice 2×2 .

La fonction \mathcal{F}_K définie en (3) peut alors s'écrire

$$\mathcal{F}_K(\hat{\mathbf{q}}) = \mathbb{A}_K \hat{\mathbf{q}} + \mathbf{q}^0 \quad (4)$$

et on a

$$\int_K f(\mathbf{q}) d\mathbf{q} = |\det(\mathbb{A}_K)| \int_{\hat{K}} f \circ \mathcal{F}_K(\hat{\mathbf{q}}) d\hat{\mathbf{q}}. \quad (5)$$

Q. 2 ★★

- [rapport]** Décrire de manière détaillée une méthodologie permettant de construire le tableau des aires contenant l'ensemble des aires de chacun des triangles contenu dans un maillage (quelconque) donné par ses tableaux des points et de connectivité.
- [code]** Ecrire une fonction `[areas]=meshes.areas(q,me)`^a retournant le tableau des aires associé au maillage donné tel que `areas(k)` soit l'aire du k -ème triangle spécifié par le tableau des sommets `q` et le tableau de connectivité `me`.

a. La fonction `areas` appartient au namespace `meshes`

3 Génération de maillages

3.1 Maillage du carré unité

L'objectif est d'écrire une fonction retournant les tableaux de points et de connectivité associé au carré unité $[0, 1] \times [0, 1]$ avec `Nx` points suivant x et `Ny` points suivant y comme présenté en Figure 2 avec pour valeurs particulières `Nx=5` et `Ny=6`.

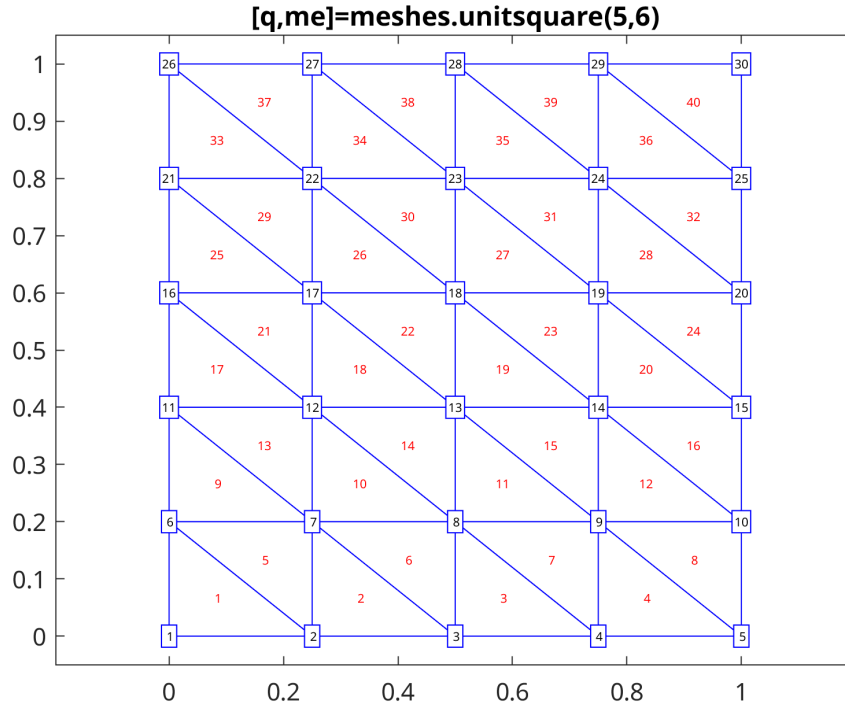


FIGURE 2 – Maillage du carré unité avec des triangles à l'aide de commande `[q,me]=meshes.unitsquare(5,6)`.

Q. 3 ★★

- [rapport]** Décrire de manière détaillée une méthodologie permettant de construire les tableaux de points `q` et de connectivité `me` associé au carré unité pour tous $N_x \geq 2$ et $N_y \geq 2$ en respectant la numérotation imposée en Figure 2 pour les tableaux de points et de connectivité.
- [code]** Ecrire une fonction `[q,me]=meshes.unitsquare(Nx,Ny)`^a retournant les tableaux de points et de connectivité associé au carré unité $[0, 1] \times [0, 1]$ avec N_x points suivant x et N_y points suivant y comme présenté en Figure 2.
- [code]** Ecrire le script `demos.unitsquaremesh01` permettant de tester cette fonction en représentant graphiquement les maillages obtenus avec $(N_x, N_y) = (5, 6)$, $(N_x, N_y) = (8, 5)$, et $(N_x, N_y) = (13, 10)$ dans trois figures distinctes.

a. La fonction `unitsquare` appartient au namespace `meshes`

3.2 Rectangle $[a, b] \times [c, d]$

L'objectif est d'écrire une fonction retournant les tableaux de points et de connectivité associé au rectangle $[a, b] \times [c, d]$ avec N_x points suivant x et N_y points suivant y comme présenté en Figure 3 avec pour valeurs particulières $(a, b, c, d, N_x, N_y) = (-2, 2, -1, 1, 5, 6)$.

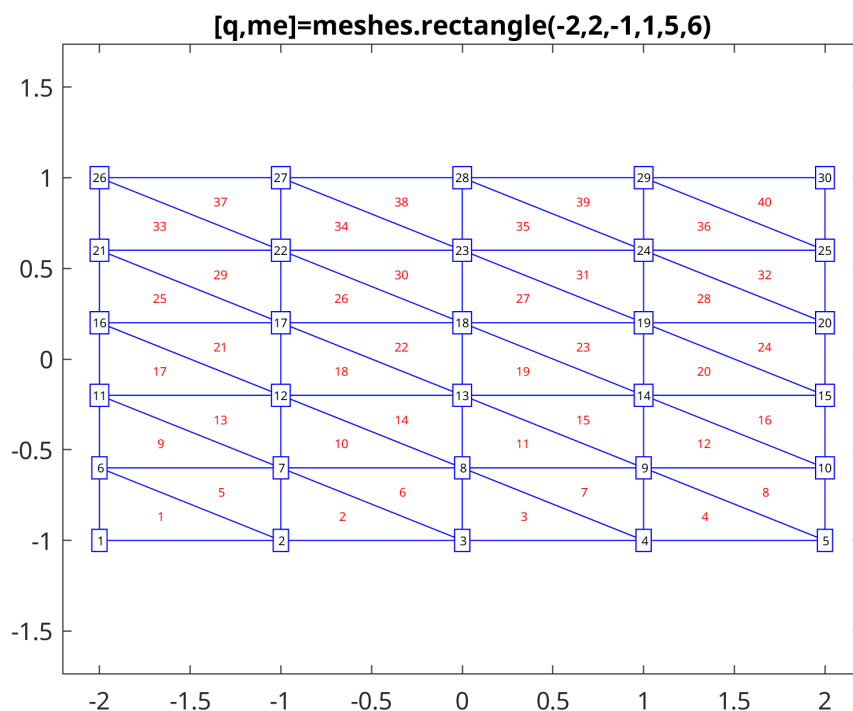


FIGURE 3 – Maillage du rectangle $[a, b] \times [c, d]$ avec des triangles avec $(a, b, c, d, N_x, N_y) = (-2, 2, -1, 1, 5, 6)$.

Q. 4 ★★

- [rapport]** Décrire de manière détaillée une méthodologie permettant de construire les tableaux de points \mathbf{q} et de connectivité \mathbf{me} associé au rectangle $[a, b] \times [c, d]$ pour tous $N_x \geq 2$ et $N_y \geq 2$.
- [code]** Ecrire une fonction `[q,me]=meshes.rectangle(a,b,c,d,Nx,Ny)` retournant les tableaux de points et de connectivité associé au rectangle $[a, b] \times [c, d]$ avec N_x points suivant x et N_y points suivant y comme présenté en Figure 3.
- [code]** Ecrire le script `demos.rectangle01` permettant de tester cette fonction en représentant graphiquement trois maillages distincts de rectangles.

3.3 Maillage du triangle de référence

L'objectif est d'écrire une fonction retournant les tableaux de points et de connectivité associé au triangle de référence de sommets $(0, 0)$, $(1, 0)$ et $(0, 1)$ avec N points sur chacune des arêtes. L'ordre de numérotation choisie des points et des triangles est celle présentée en Figure 4 avec pour valeur particulière $N=5$.

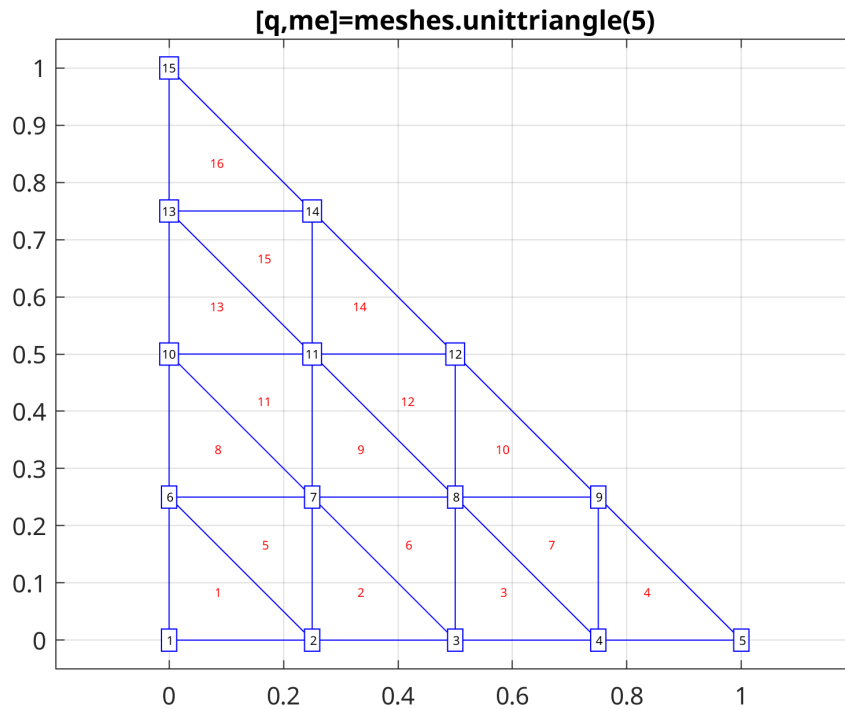


FIGURE 4 – Maillage du triangle de référence avec des triangles.

Q. 5 ★★

- [rapport]** Décrire de manière détaillée une méthodologie permettant de construire les tableaux de points \mathbf{q} et de connectivité \mathbf{me} associé au triangle unité pour tout $N \geq 2$ et respectant l'ordre de numérotation de la Figure 4.
- [code]** Ecrire une fonction `[q,me]=meshes.unittriangle(N)` retournant les tableaux de points et de connectivité associé au triangle unité avec N points sur chacune des arêtes.
- [code]** Ecrire le script `demos.unittriangle01` permettant de tester cette fonction en représentant graphiquement trois maillages distincts du triangle unité.

3.4 Maillage d'un triangle

Q. 6 ★★

- [rapport]** Décrire de manière détaillée une méthodologie permettant de construire les tableaux de points \mathbf{q} et de connectivité \mathbf{me} associé à un triangle non dégénéré de sommets $(\mathbf{q}^0, \mathbf{q}^1, \mathbf{q}^2)$.
- [code]** Ecrire une fonction `[q,me]=meshes.triangle(q0,q1,q2,N)` retournant les tableaux de points et de connectivité associé au triangle de sommets $\mathbf{q}_0, \mathbf{q}_1$ et \mathbf{q}_2 avec N points sur chacune des arêtes comme présenté en Figure 5.
- [code]** Ecrire le script `demos.triangle01` permettant de tester cette fonction en représentant graphiquement trois maillages de triangles distincts et différents du triangle unité.

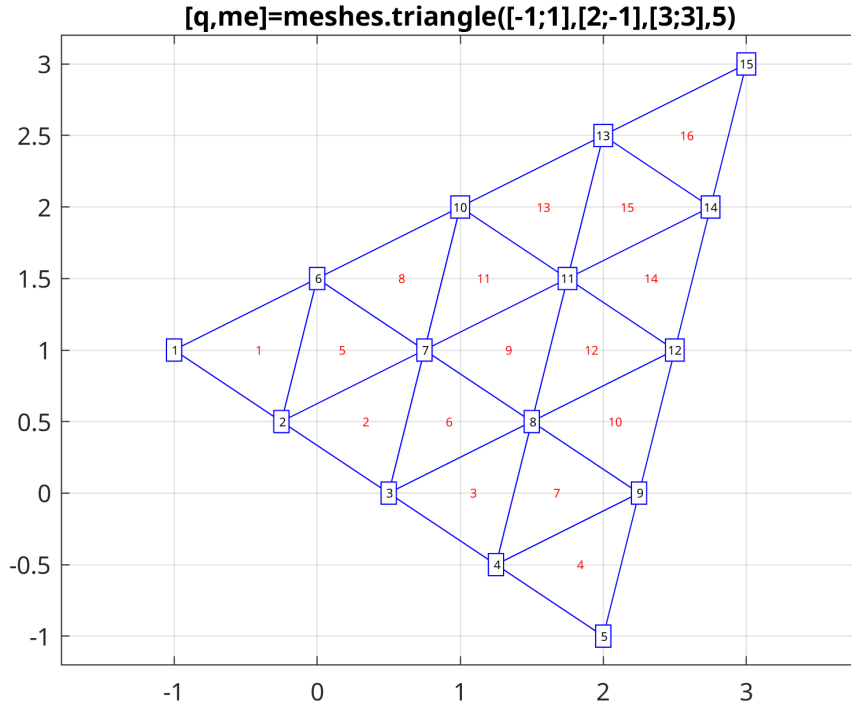


FIGURE 5 – Maillage du triangle de sommets $\mathbf{q}^0 = (-1, 1)$, $\mathbf{q}^1 = (2, -1)$ et $\mathbf{q}^2 = (3, 3)$ avec des triangles. Ici, $N=5$.

4 Namespace `sMesh`

Dans cette section, on va utiliser une structure pour stocker des informations sur un maillage. Voici les champs imposés :

- `q` : tableau des sommets,
- `me` : tableau de connectivité,
- `nq` : nombre de sommets dans le maillage,
- `nme` : nombre de triangles dans le maillage,
- `ba` : tableau des barycentres des triangles du maillages,
- `areas` : tableau des aires des triangles du maillages.

Q. 7 ★★
[code]

- a. Ecrire une fonction `S=sMesh.constructor(q,me)` retournant la structure `S` contenant les champs décrits précédemment et correctement initialisés.
- b. Ecrire une fonction `S=sMesh.rectangle(a,b,c,d,Nx,Ny)` retournant la structure `S` représentant le maillage du rectangle $[a, b] \times [c, d]$ avec `Nx` points suivant x et `Ny` points suivant y .
- c. Ecrire une fonction `S=sMesh.triangle(q0,q1,q2,N)` retournant la structure `S` représentant le maillage du triangle de sommets $(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2)$ et avec `N` points sur chaque arête.
- d. Ecrire une fonction `sMesh.plotmesh` représentant graphiquement le maillage donné par la structure `S` sans numérotation en utilisant la fonction `graphics.plotmesh` avec les mêmes paramètres optionnels.
- e. Ecrire un script `demos.sMesh01` utilisant les fonctions `sMesh.plotmesh` et `sMesh.rectangle`.
- f. Ecrire une fonction `sMesh.plotmeshnumbers` représentant graphiquement le maillage donné par la structure `S` d'un rectangle avec les numérotations des sommets et des triangles en utilisant la fonction `graphics.plotmeshnumbers` avec les mêmes paramètres optionnels..
- g. Ecrire un script `demos.sMesh02` utilisant les fonctions `sMesh.plotmeshnumbers` et `sMesh.triangle`.