**Table des matières**

[**Remerciement** 4](#_Toc530518651)

[**INTRODUCTION** 5](#_Toc530518652)

[**I) Principe des éléments finis :** 6](#_Toc530518653)

[A) Comment les obtient-on à partir d’un système continu pour arriver à un système discret ? 6](#_Toc530518654)

[B) Comment passe-t-on du système discrétisé aux fonctions de forme ? 6](#_Toc530518655)

[**II) Cas de matrice de masse :** 7](#_Toc530518656)

[A) Maillage à l’aide de la fonction ‘initmesh’ : 7](#_Toc530518657)

[B) Calcul de Matrice de masse : 8](#_Toc530518658)

[**III) Cas de matrice de Rigidité :** 10](#_Toc530518659)

[A) Maillage à l’aide de ‘pdetool’ : 10](#_Toc530518660)

[B) Calcul de Matrice de rigidité : 10](#_Toc530518661)

[**III) Programme pour une numérotation unifiée :** 12](#_Toc530518662)

[A) Numérotation des sommets : 12](#_Toc530518663)

[B) Numérotation des triangles : 14](#_Toc530518664)

[C) Calcul de Matrice de Rigidité : 14](#_Toc530518665)

[D) Calcul de matrice de masse : 15](#_Toc530518666)

[E) Calcul de second membre : 16](#_Toc530518667)

[**CONCLUSION** 18](#_Toc530518668)

**Table de Figures :**

[**Figure 1:**Maillage d’un carré par ‘initmesh’ 8](#_Toc530760605)

[**Figure 2:** Maillage d’un carré par ‘pdetool’ numéroté 10](#_Toc530760606)

[**Figure 3:** Maillage d’un carré par ‘pdetool’ 12](#_Toc530760607)

# Remerciement

Je tiens à remercier notre cher professeur monsieur MOUMIDA de nous avoir donner ce travail afin de se familiariser avec la notion des élément finis à l’aide de ce fameux logiciel. Grâce à son aide et ses conseils concernant les tâches évoquées dans ce compte rendu, on a pu comprendre, maîtriser et réaliser ce travail.

**INTRODUCTION**

La méthode des éléments finis est une méthode pour résoudre des problèmes de physique ou plus généralement des équations différentielles avec conditions aux limites. De façon générale, on part donc d’un problème (P) et on introduit sa formulation variationnelle (Q). On montre l’existence et l’unicité d’une solution par le théorème de LaxMilgram. On travaille dans des espaces de Hilbert : on peut donc faire de l’approximation interne.

Dans ce compte rendu, on va vous présenter une petite introduction concernant le principe des éléments finis, la méthode par laquelle on va calculer la matrice de rigidité et celle de masse que ça soit à l’aide d’un maillage déjà proposé par le logiciel utilisé qui est « MATLAB » ou grâce à un maillage qu’on va l’imposer au logiciel.

# **I) Principe des éléments finis :**

## A) Comment les obtient-on à partir d’un système continu pour arriver à un système discret ?

De façon mathématique :

Soit Ω le domaine ouvert de Rn (où n=1,2 ou3), de frontière Ω et sur lequel on cherche à résoudre un problème (P) ramené à une équation aux dérivées partielles, munie de conditions aux limites. On écrit la formulation variationnelle et on obtient (Q) suivant :



Où V est un espace de Hilbert. Sous réserve que l’équation de départ ait les bonnes propriétés (hypothèses du théorème de Lax-Milgram), (Q) admet une unique solution u. Pour obtenir une approximation numérique de u, on va remplacer V de dimension infinie par une série de sous-ensembles de dimensions finies, et on va résoudre le problème approché :



On sait que est de dimension finie, c’est donc un fermé de V. Or V est un espace de Hilbert : l’est aussi. On applique donc encore une fois le théorème de Lax-Milgram : existe et est unique. En pratique l’espace sera construit à partir d’un maillage du domaine Ω, l’indice h désignant la taille typique des mailles.

## B) Comment passe-t-on du système discrétisé aux fonctions de forme ?

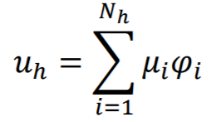
Résumons : nous avions une EDP à résoudre sur un domaine Ω. Nous avons écrit la formulation variationnelle (Q) et on s’est ramené au problème :



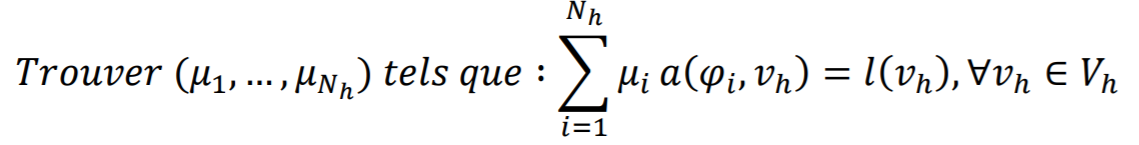
On va chercher une approximation u. Pour cela, on définit un maillage du domaine Ω grâce auquel on va définir un espace d’approximation (sous espace vectoriel de V et de dimension finie ). Le problème approché est donc :



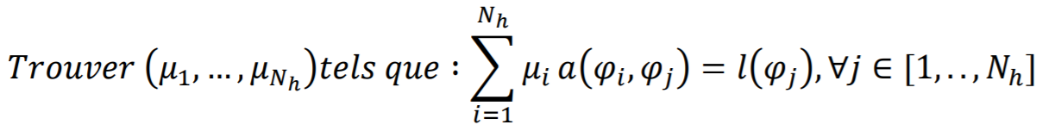
 On définit alors une base de . On peut donc décomposer notre solution approchée en une combinaison linéaire des fonctions de base :



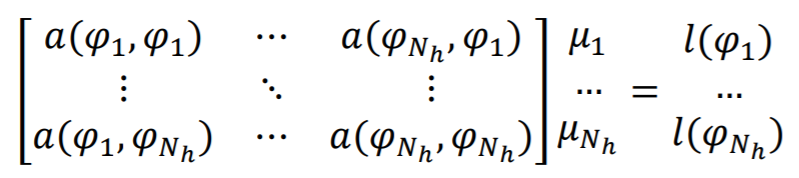
Et donc le problème devient :



Or a et l sont linéaires donc on peut écrire :



Cela revient à résoudre le système suivant : Aμ=B



Pour limiter le volume de calculs, on va définir des fonctions de petit support : chaque sera nulle partout sauf sur la ième maille. Ainsi sera le plus souvent nul (les fonctions sont de supports disjoints) et la matrice A sera creuse.

Donc dans ce compte rendu on va travailler sur le problème suivant :

∆u + u = f sur Ω

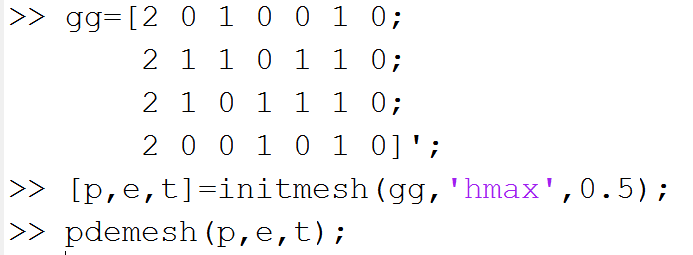
u = 0 sur ∂Ω

**APPLICATIONS SUR MATLAB :**

# II) Cas de matrice de masse :

## Maillage à l’aide de la fonction ‘initmesh’ :

Une façon de voir si on a mal décrit la géométrie (en particulier le sens de parcourt des différents côtés) est de mailler la géométrie (carré):



La fonction ***initmesh*** permet d'obtenir un premier maillage décrit à l'aide de :

* Une matrice **gg** décrit la géométrie du problème PDE
* Une matrice **p** de taille 2 ∗ np où np est le nombre de points constituant le maillage : 1ère ligne = abscisse des points, 2ème ligne leur ordonnée. (Dans notre cas np=13)
* Une matrice **t** de taille 4 ∗ nt où nt est le nombre de triangles : voir la description dans « initmesh.htm ». (Dans notre cas nt=16)
* Une matrice **e** de taille 7 ∗ ne où ne est le nombre de côtés des triangles qui sont sur la frontière ∂Ω : voir la description dans « initmesh.htm ».

La fonction ***pdemesh*** trace le maillage.

Alors le résultat de ce maillage nous donne la figure ci-dessous :

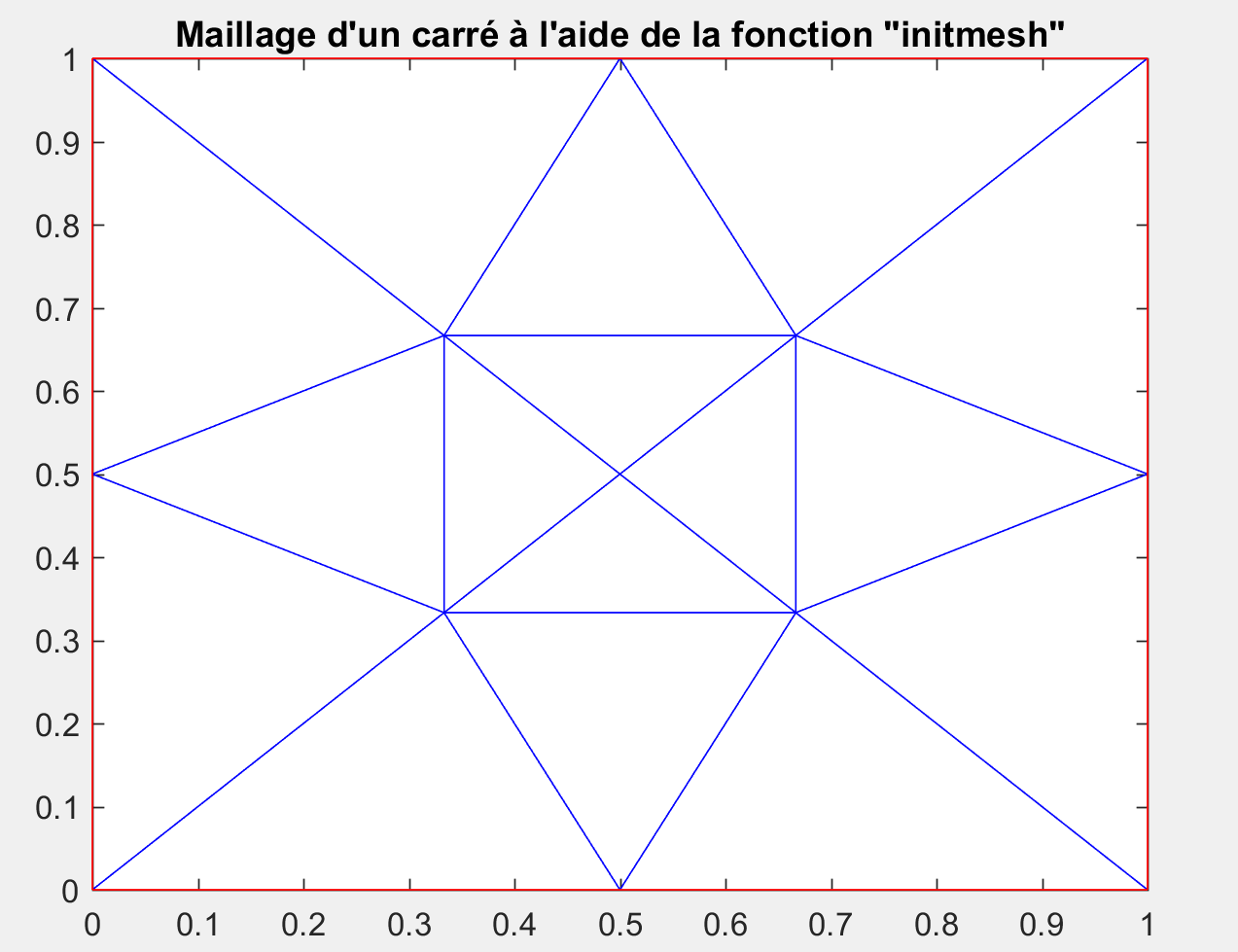
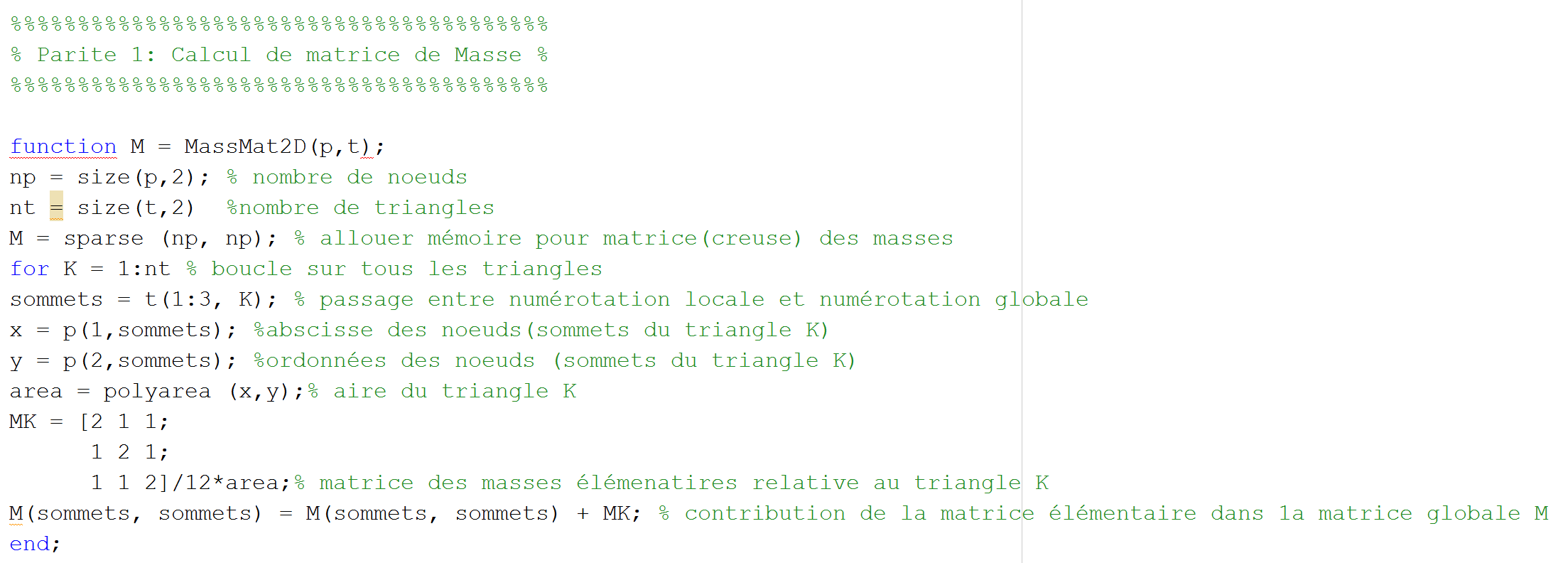


Figure 1:Maillage d’un carré par ‘initmesh’

## Calcul de Matrice de masse :

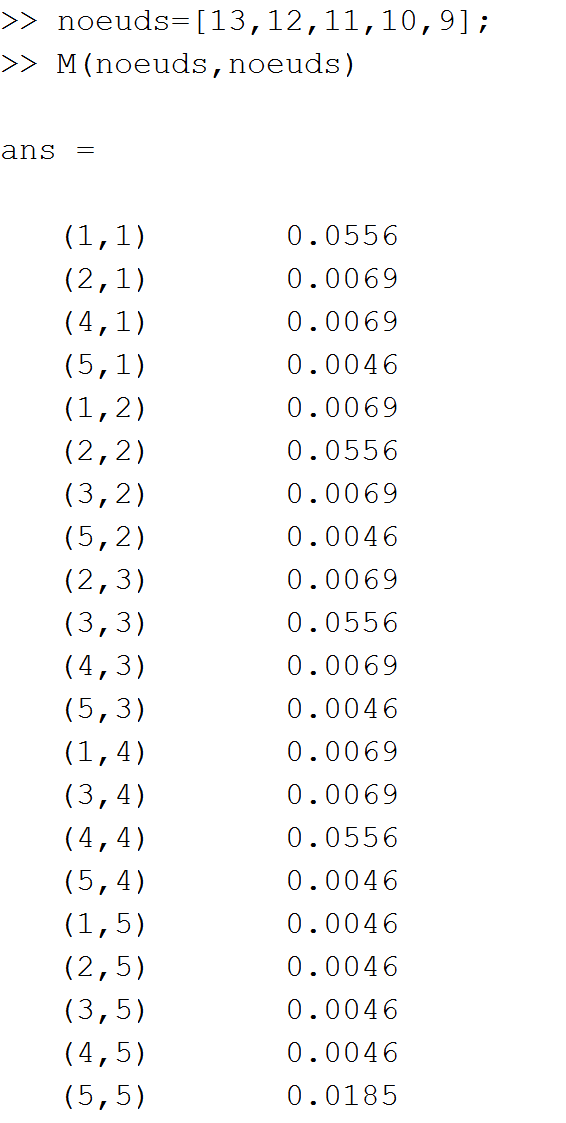
Par la suite, on peut facilement calculer la matrice de masse globale en utilisant le code MATLAB suivant :



Après avoir la matrice de masse globale, on peut facilement trouver la matrice correspondante aux sommets interne qu’ils nous intéressent. En effet, on cherche au premier temps la numérotation de ces points effectuées par MATLAB, et cela à l’aide de la matrice p.

On trouve que ces points sont : [13,12,11,10,9].

Et donc la matrice de masse correspondante aux sommets internes est donnée par les commandes suivantes :



**Remarque :**

Si on veut afficher toute la matrice avec les zéros aussi on utilise ‘zeros’ au lieu de ‘sparse’ dans la matrice M.

# III) Cas de matrice de Rigidité :

## Maillage à l’aide de ‘pdetool’ :

Parfois pour des géométries plus spécifiques (autres que des morceaux d'ellipses ou de segments de droite), on a recours à utiliser la commande ‘pdetool’ de MATLAB, puis on dessine la forme géométrie souhaitée.

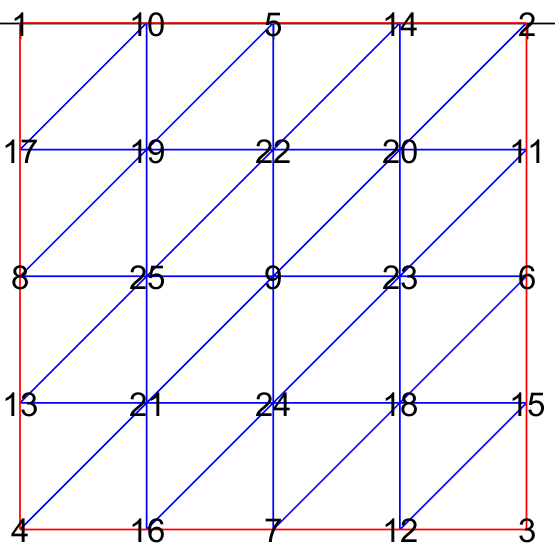
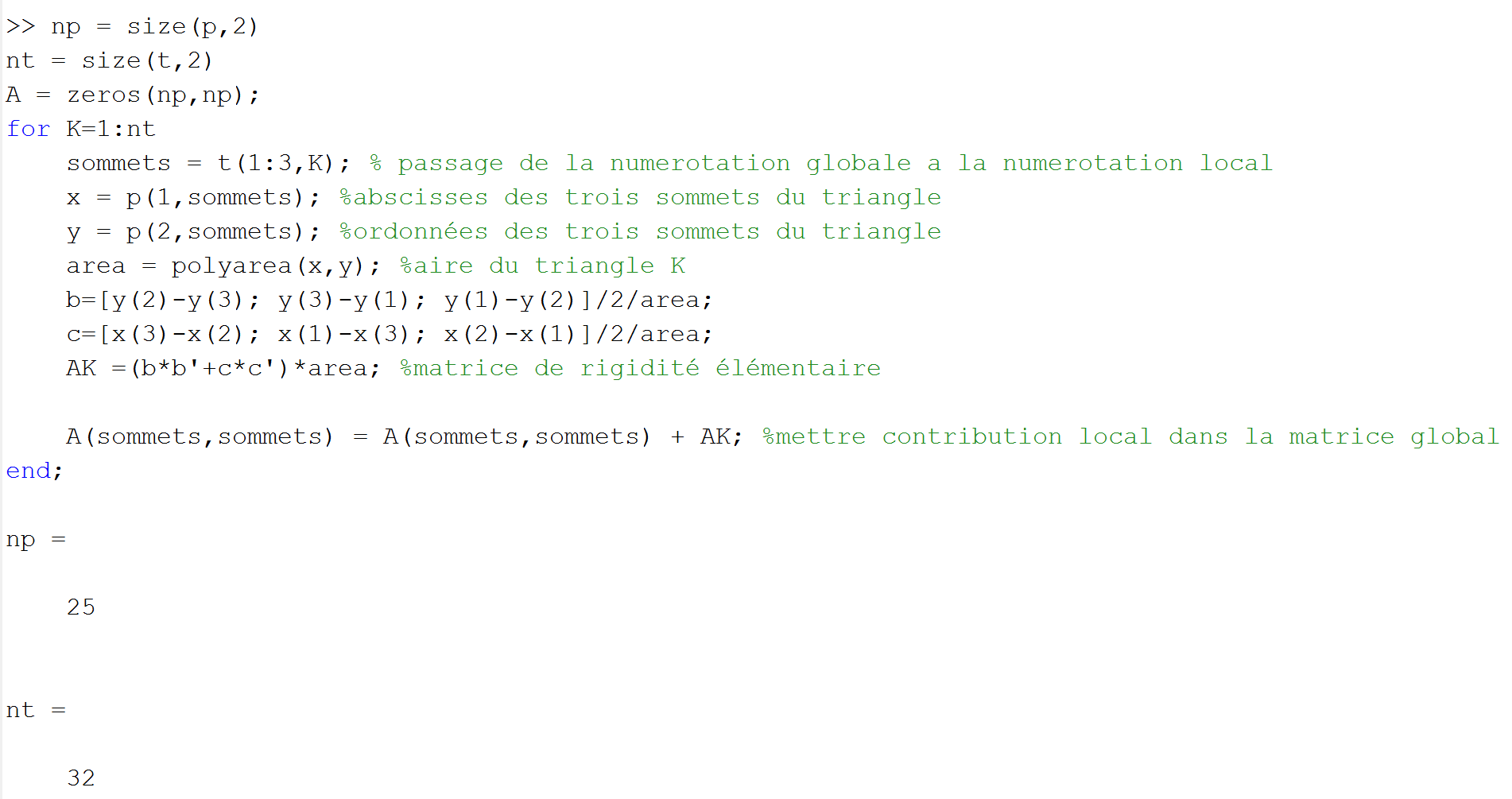


Figure 2: Maillage d’un carré par ‘pdetool’ numéroté

## Calcul de Matrice de rigidité :

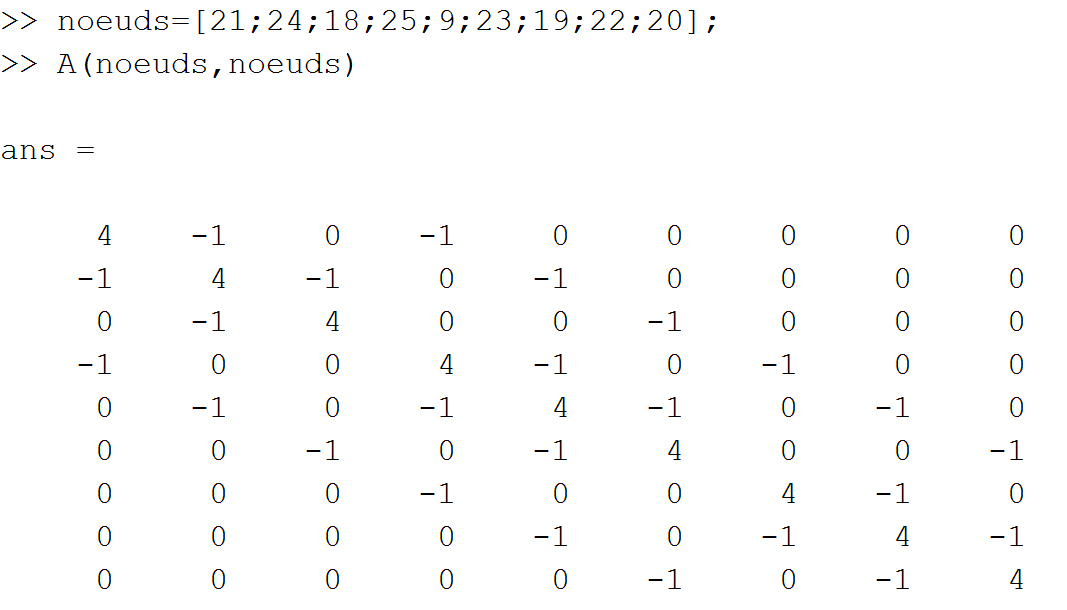
Après avoir exporter le maillage on peut facilement, grâce au code MATLAB suivant, calculer la matrice de rigidité globale associée à la figure 2.



De même que précédemment, Après avoir la matrice de rigidité globale A, on peut facilement trouver la matrice correspondante aux sommets interne qu’ils nous intéressent. En effet, on cherche au premier temps la numérotation de ces points effectuées par MATLAB, et cela à l’aide de la matrice p donnée par l’outil de maillage qu’on a utilisé.

On trouve que ces points sont : [21;24;18;25;9;23;19;22;20]

Et donc la matrice de rigidité correspondante aux sommets internes est donnée par les commandes suivantes :



**Remarque :**

Dans ce cas, on a eu la matrice de rigidité associée au sommets interne pleine de zeros vu qu’on a utilisé la fonction ‘zeros’ pour la matrice A.

# III) Programme pour une numérotation unifiée :

## Numérotation des sommets :

MATLAB nous donne une numérotation aléatoire des sommets, ce qui nous a poussé à penser à programmer une méthode pour laquelle on impose à MATLAB une numération bien spécifique qui priorise les sommets internes au premier temps, la chose qui va nous faciliter la tâche par la suite surtout lors des calculs.

A l’aide de la commande ‘***pdetool*’** on vous présente ci-dessous la géométrie sur laquelle on va travailler dans cette partie.

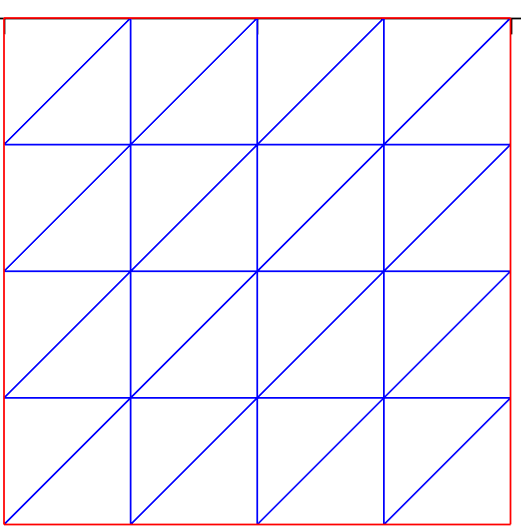
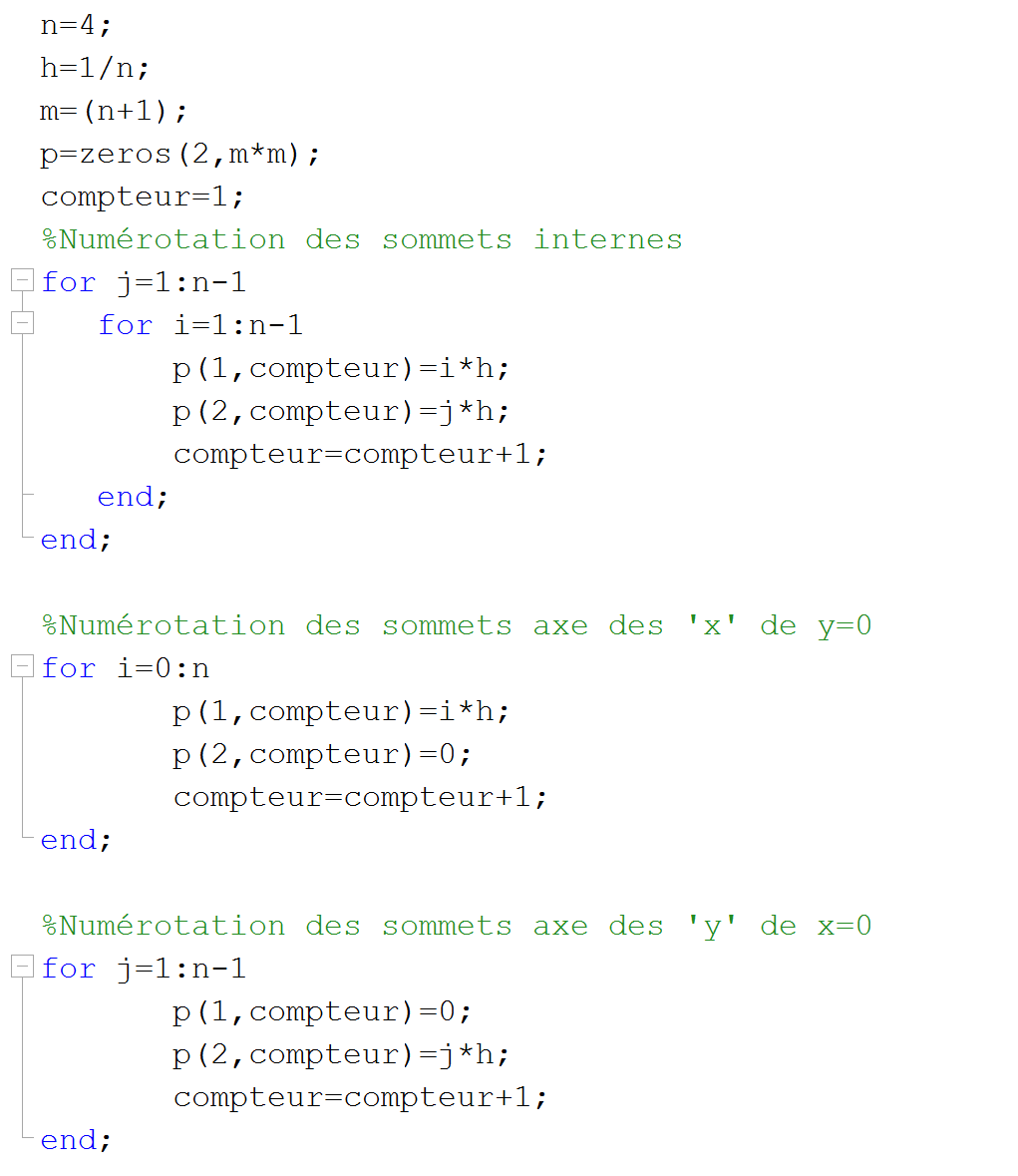
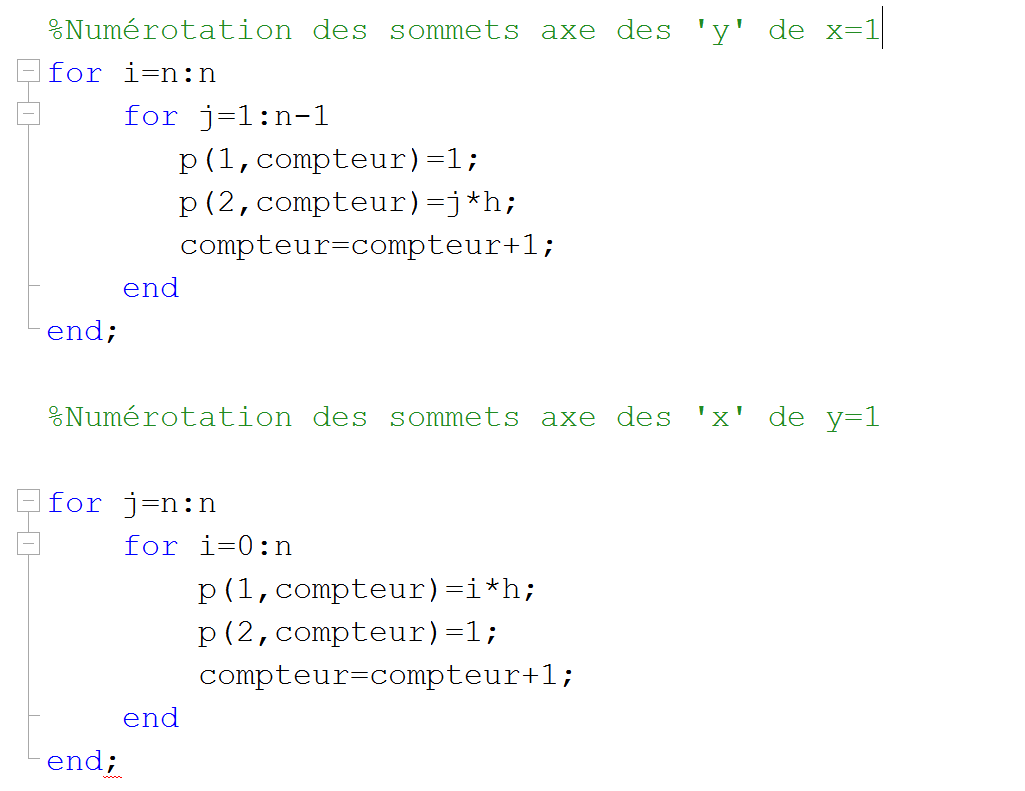


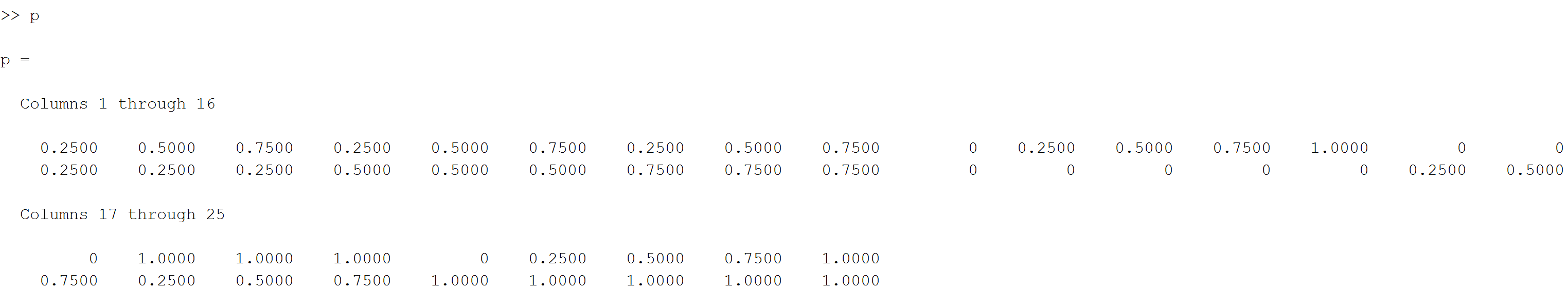
Figure 3: Maillage d’un carré par ‘pdetool’

On vous présente ci-dessous le code associé à la numérotation qu’on va imposer au maillage :



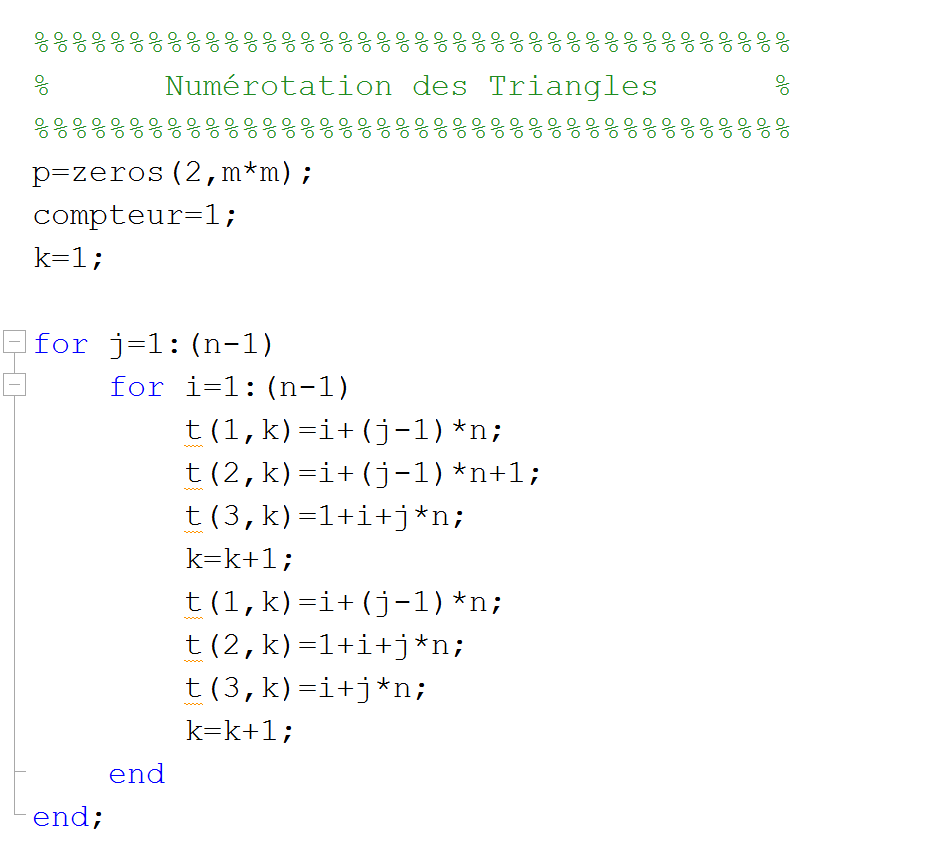


Et bien effectivement lorsqu’on fait appel à la matrice p de sommets, on obtient la numérotation par ordre souhaité comme le montre le résultat ci-dessous :



## Numérotation des triangles :

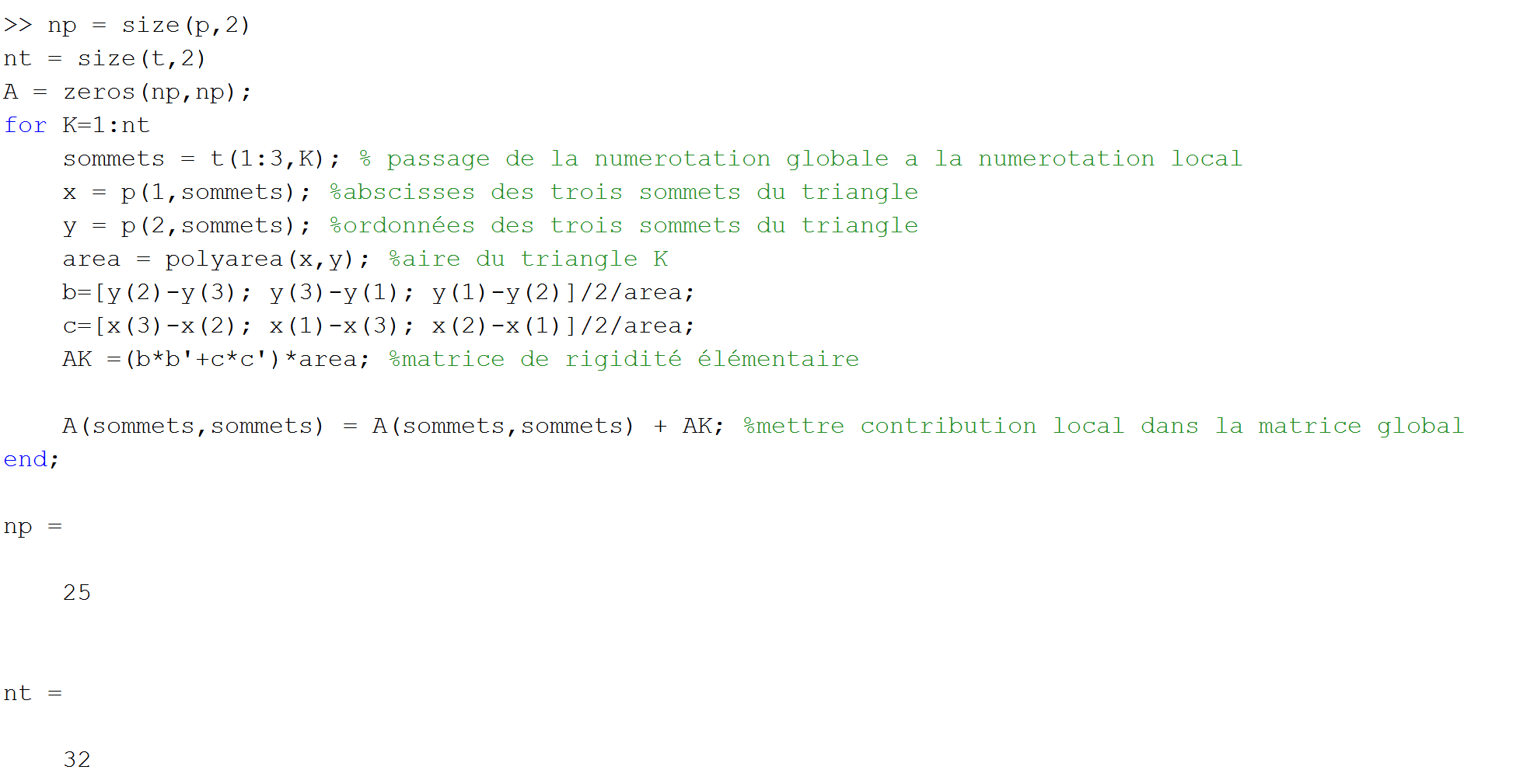
Voici le programme à suivre pour permettre à MATLAB d’utiliser la numérotation des triangles unifiés et souhaité :

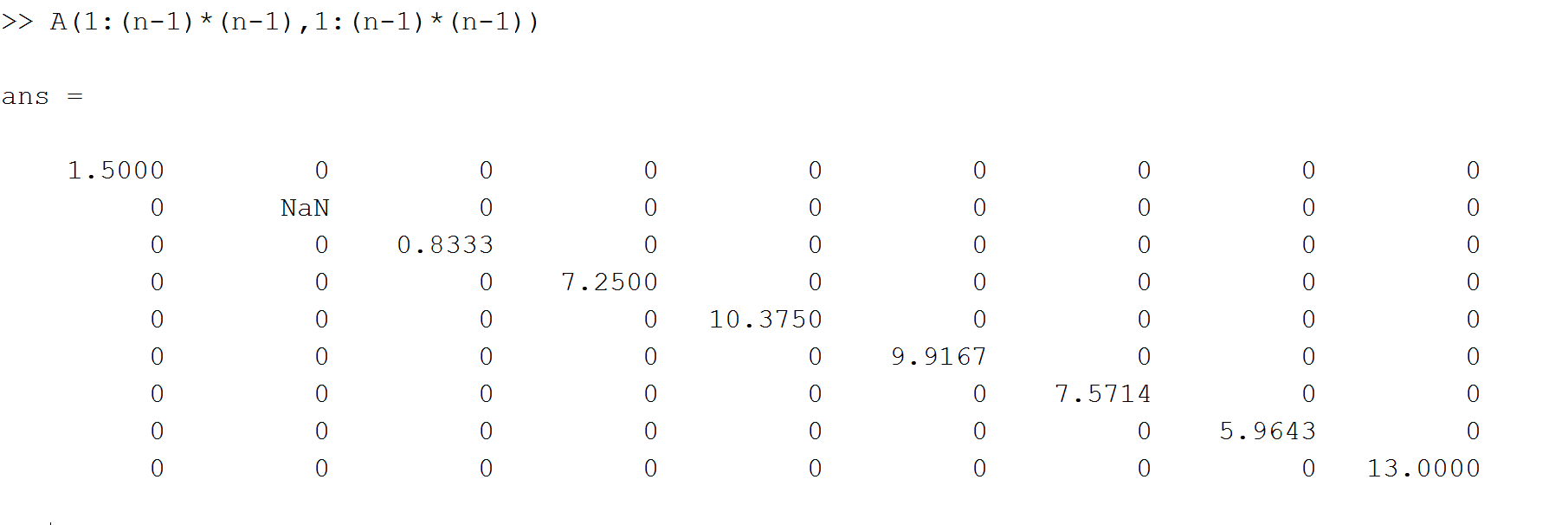


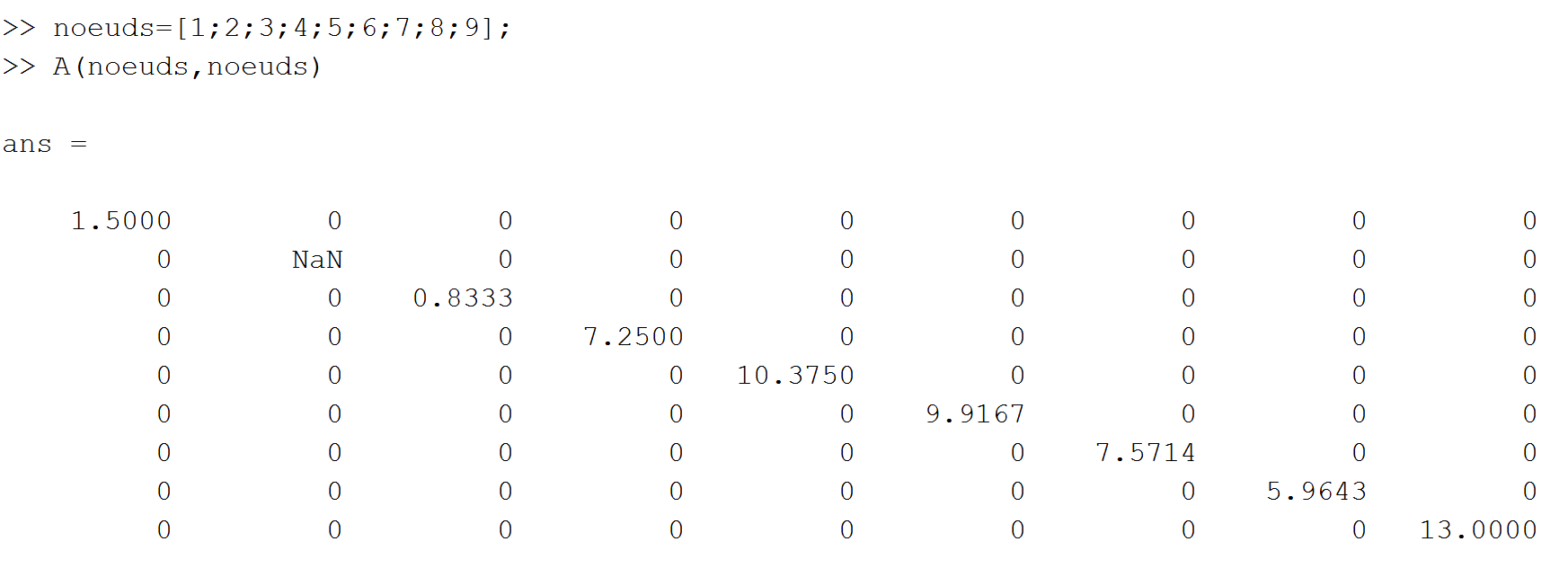
## Calcul de Matrice de Rigidité :

De même que précédemment le programme associé au calcul de la matrice de rigidité est le suivant :



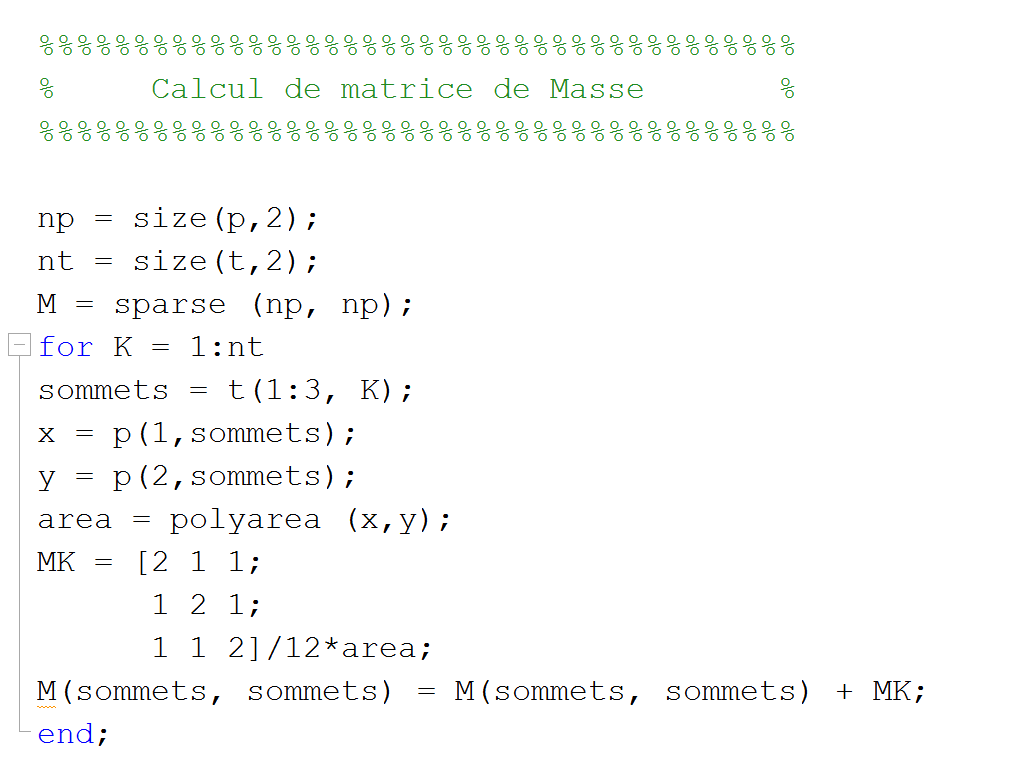


Ce qui diffère cette fois par rapport à l’exemple précèdent, c’est que nous connaissons déjà numérotation des sommets internes, on n’a pas besoin de les chercher dans la matrice p des sommets. En effet, pour trouver la matrice de rigidité correspondante au sommets internes il suffit de taper la commande suivante :Ou encore :

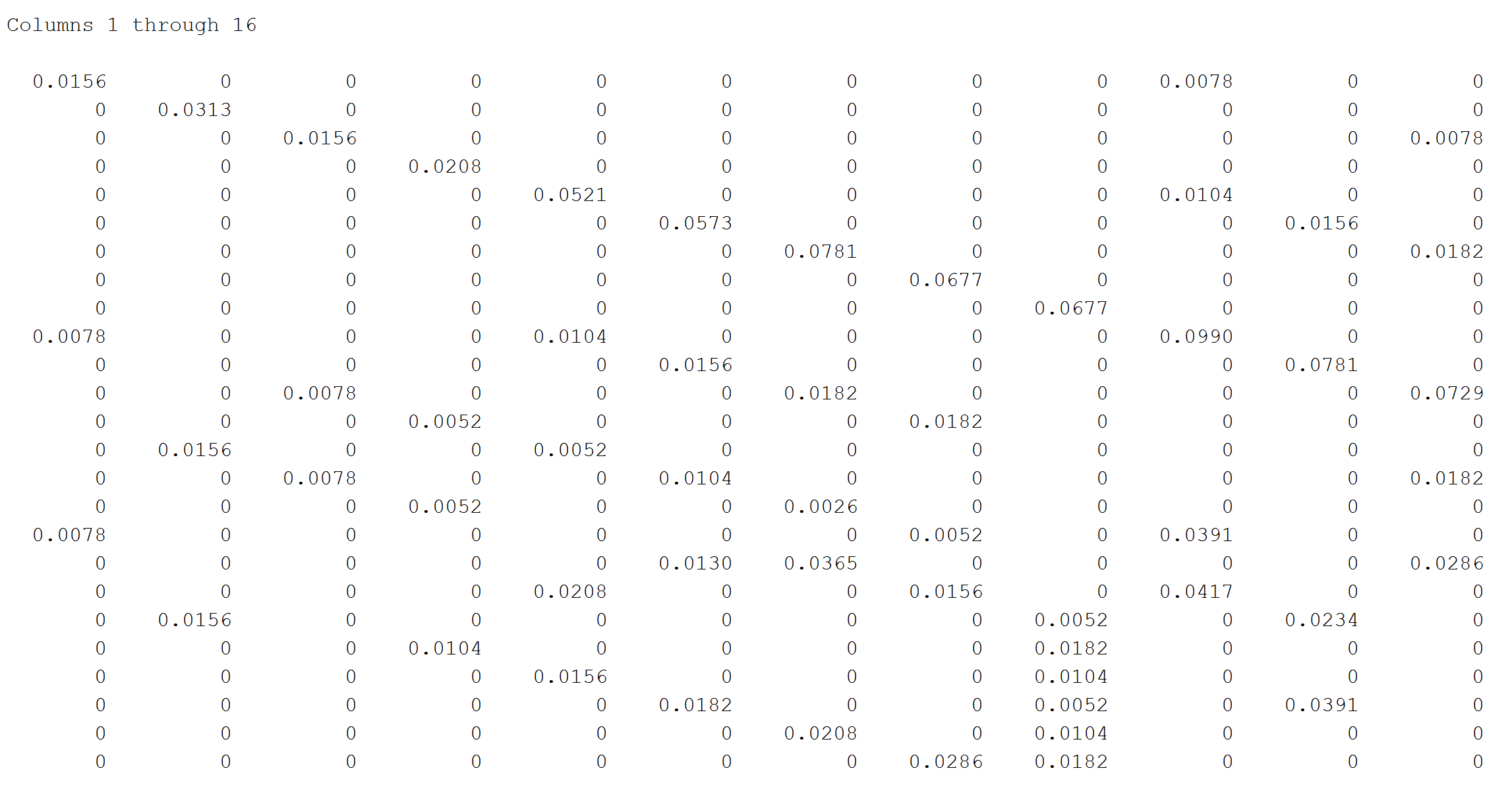


## Calcul de matrice de masse :

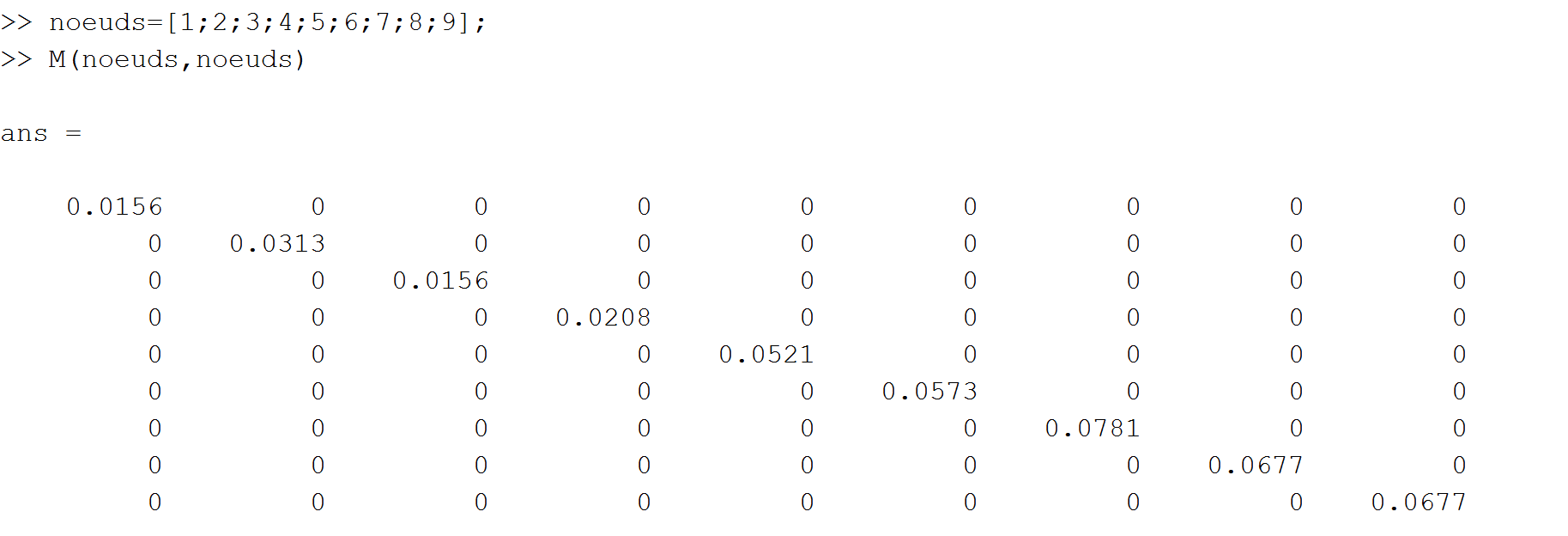
En utilisant le même programme que précedemment ;dans l’exemple de la matrice de masse;présenté ci-dessous on calcul la matrice de masse dans ce cas :



Un extrait de la matrice de masse globale trouvée est :

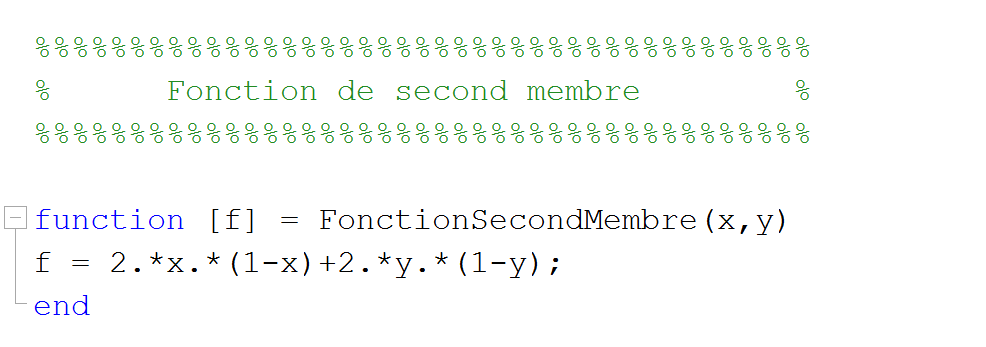


Donc pour arriver au but, qui est la matrice de masse correspondante au sommet interne il suffit de taper la commande suivante :

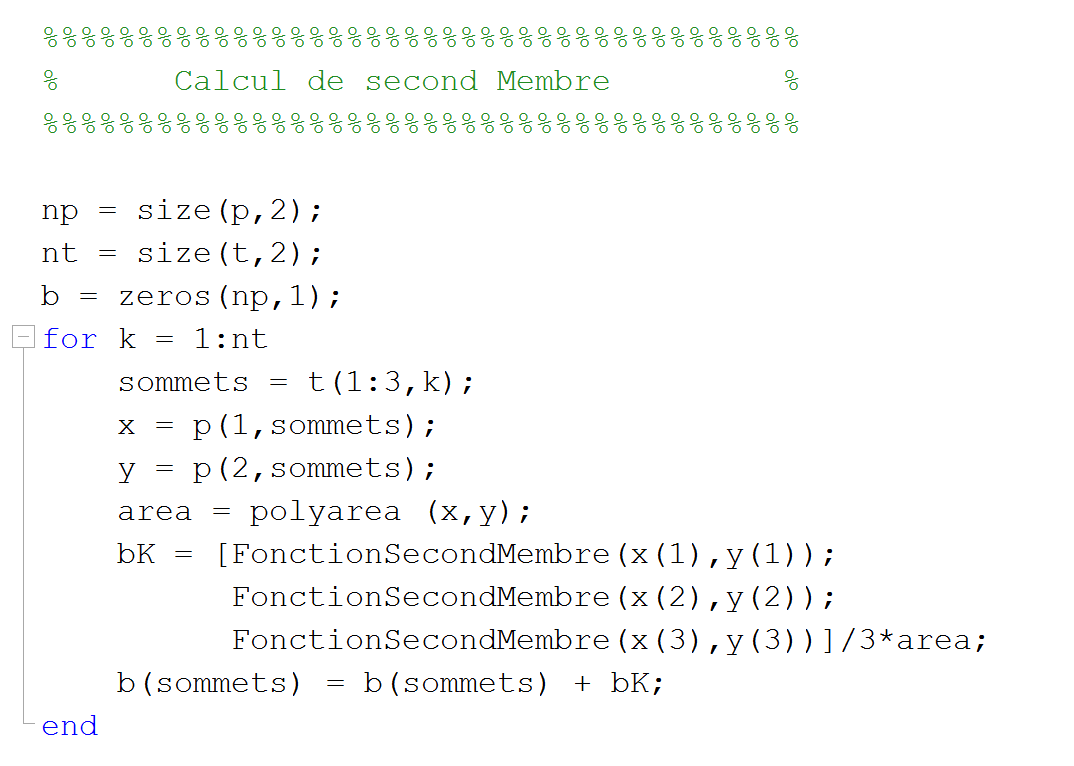


## Calcul de second membre :

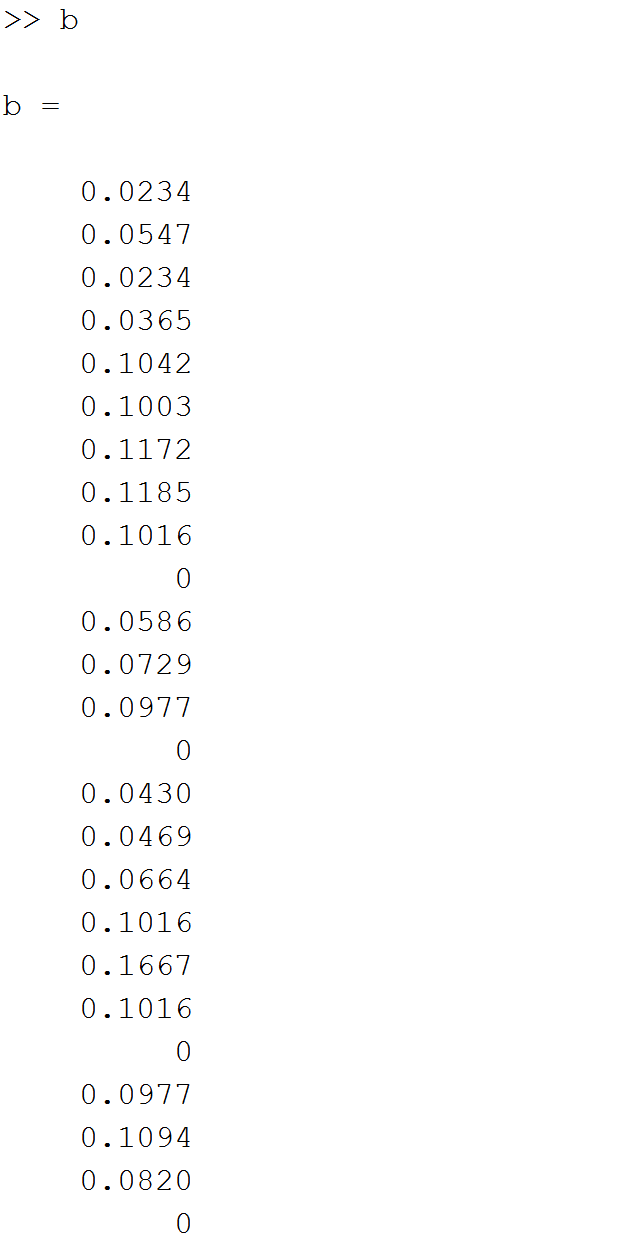
Pour calculer le second membre, on aura besoin de définir au premier temps la fonction f c’est pour cela on fait appel à un script dont on l’a défini clairement sous le nom de ‘FonctionSecondMembre.m’ :



Donc on peut facilement calculer le second membre à l’aide de code suivant :



Et donc après l’exécution on obtient le vecteur b :



**CONCLUSION**

Tous ces screenshot reflètent l’étude réalisée sur le logiciel « MATLAB » accompagnés par des résultat. Alors on peut conclure que résoudre un problème par une méthode d’éléments finis nous permet d’aboutir à une résolution relativement précise, c’est ce que ce compte rendu nous a rajouté de plus .