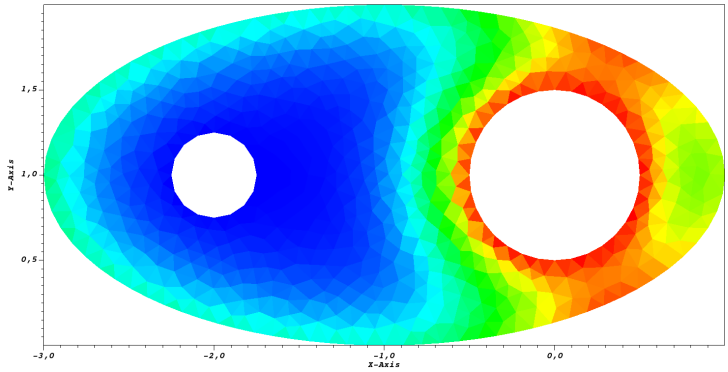
Matmeca 2ème année - TP du cours Introduction à la méthode des volumes finis
année 2024-2025

---

# TP : Approximation volumes finis de l'équation de la chaleur en dimension 2 sur maillage non structuré

---




## 1 Le modèle

Le but de ce TP est de programmer un schéma pour résoudre le problème suivant :

$$\begin{aligned}
&\partial_t T = \nabla \cdot (D \nabla T), \qquad t > 0, \quad \mathrm{x} \in \Omega \in \mathbb{R}^2 \\
&CI : T(0, \mathrm{x}) = T_{init}(\mathrm{x}) \\
&CL : T(t, \mathrm{x}) = T_b(t, \mathrm{x}) \text{ ou } -D(\mathrm{x}) \nabla T(t, \mathrm{x}) \cdot n(\mathrm{x}) = \phi_b(t, \mathrm{x}) \qquad \mathrm{x} \in \partial\Omega,
\end{aligned}$$

où l'inconnue est $T(t, \mathrm{x})$. Le coefficient de diffusion $D$ est une fonction positive de x. Des conditions aux limites différentes peuvent être appliquées sur différentes parties du bord de $\Omega$.

Nous étudierons plusieurs problèmes qui diffèrent par leur condition initiale et leurs conditions aux limites. À l'issue de ce TP, vous devriez être en mesure d'effectuer la simulation d'un problème de thermique tel que celui présenté dans l'illustration ci-dessus.

Pour la suite, commencez par lire le sujet en entier (sans les annexes). Puis prenez le temps de réfléchir en organisant votre programme sur le papier en vue de traiter le cas test n°1 : même si l'essentiel est donné dans ce document, ne foncez pas tête baissée dans l'écriture de votre programme. En particulier, il faut soigneusement traduire le schéma numérique en utilisant les structures de données proposées.

## 2 Préliminaires

Pour ce TP, vous aurez besoin du logiciel de maillage `gmsh` et du logiciel de visualisation `VisIT`. Ces logiciels peuvent être lancés en tapant dans un terminal leur chemin d'accès absolu `/usr/local/bin/gmsh_4.11.1` et `/opt/visit/3.4.1/bin/visit`.

Comme cela est un peu fastidieux, on propose de créer deux alias, c'est-à-dire des noms raccourcis, qui permettront de lancer les logiciels plus simplement. La procédure pour cela est la suivante :

1. dans votre répertoire personnel, ouvrez le fichier de configuration `.bashrc`, avec un éditeur de texte quelconque ;
2. à la fin de ce fichier, rajoutez les deux lignes suivantes :

   ```
   alias gmsh="/usr/local/bin/gmsh_4.11.1"
   alias visit="/opt/visit/3.4.1/bin/visit"
   ```

   puis sauvez le fichier et quittez l'éditeur ;
3. enfin, dans un terminal, dans votre répertoire personnel, tapez la commande `. .bashrc` (oui, vous avez bien lu : tapez bien « point espace point bashrc »), ce qui permet au système de prendre en compte les modifications faites au fichier de configuration.

Maintenant, vous pouvez lancer les deux logiciels en tapant dans un terminal les commandes `gmsh` et `visit`, quel que soit le répertoire dans lequel vous vous trouvez.

Une aide pour l'utilisation de `VisIT` est donnée en annexe D. Vous pouvez aussi utiliser le logiciel *Paraview*, qui se lance avec la commande `paraview`. Celui-ci est d'utilisation un peu plus complexe, mais permet d'obtenir le même type de résultats, avec les mêmes fichiers de donnée.

## 3 Le schéma

On considère le schéma explicite sur maillage non structuré vu en cours.

**Le maillage.** Le maillage constitué de triangles, est généré avec le logiciel `gmsh`, et sauvegardé sous format medit (avec l'extension .mesh), comme vu en TP de C++.

Toutes les subroutines et fonctions pour lire le maillage et en tirer les informations nécessaires au schéma volumes finis sont stockées dans le module `mod_maillage`, à télécharger sur `http://www.math.u-bordeaux.fr/~lmieusse/PAGE_WEB/enseignement.html`.

Dans le programme principal, il suffit d'appeler la subroutine `maillage` pour calculer toutes les informations nécessaires (voir ci-dessous pour les arguments de la subroutine).

**Le schéma.** À l'initialisation, on pose pour chaque maille $\Omega_i$

$$T_i^0 = T_{init}(\mathrm{x}_i),$$

où $\mathrm{x}_i$ est le centre de gravité la maille $\Omega_i$.

Ensuite, on pose pour $n = 0$ à $n_{max}$ et $i = 1$ à $i_{max}$,

$$T_i^{n+1} = T_i^n - \frac{\Delta t}{|\Omega_i|} \sum_{e \in \partial\Omega_i} |e| F_{i,e}^n.$$

Les flux aux arêtes intérieures sont définis par

$$F_{i,e}^n = -D(\mathrm{x}_e) \frac{T_k^n - T_i^n}{d_e},$$

où $k$ est l'indice de la maille $\Omega_k$ qui partage l'arête $e$ avec $\Omega_i$, $\mathrm{x}_e$ est le milieu de $e$, et $d_e = \|\overrightarrow{\mathrm{x}_i\mathrm{x}_k}\|$ est la distance du segment joignant les centres des cercles circonscrits des mailles $\Omega_i$ et $\Omega_k$ adjacentes à $e$.

Les flux sur les arêtes de bord sont définis en utilisant les conditions aux limites. Pour une condition de Neumann, on pose

$$F_{i,e}^n = \phi_b(t_n, \mathrm{x}_e).$$

Pour une condition de Dirichlet, on pose

$$F_{i,e}^n = -D(\mathrm{x}_e)\frac{T_b(t_n, \mathrm{x}_e) - T_i^n}{d_e},$$

où $\mathrm{x}_e$ est le milieu de l'arête $e$, et $d_e = \|\overrightarrow{\mathrm{x}_i\mathrm{x}_e}\|$ est la distance du segment joignant le centre du cercle circonscrit à $\Omega_i$ au milieu de l'arête de bord.

On rappelle enfin que pour des CL de Neumann homogènes, ce schéma est $L^\infty$ stable sous la condition CFL

$$\Delta t \leq \min_i \frac{|\Omega_i|}{\sum_{e\in\partial\Omega_i} |e| D(\mathrm{x}_e)/d_e}. \tag{1}$$

Cette condition est nécessaire, même avec d'autres conditions aux limites, et vous devrez donc l'imposer dans votre programme.

# 4 Le code

Un certain nombre de modules sont fournis sur

`http://www.math.u-bordeaux.fr/~lmieusse/PAGE_WEB/enseignement.html`

pour la lecture et le traitement du maillage ainsi que pour les sorties fichier au format `vtk` permettant de visualiser les résultats du code.

Il vous reste à écrire le programme principal qui contient toutes les structures de données nécessaires et les instructions correspondant au schéma, et vous pouvez aussi éventuellement créer quelques modules pour définir les fonctions utilisées dans le problème (donnée initiale, coefficient de diffusion, solutions exactes, etc.).

Pour compiler votre code, utilisez un makefile. Durant la phase de développement, utilisez les options de détection des erreurs

```
-g -ffpe-trap=invalid,zero,overflow -fbounds-check -fcheck=all -Wall
```

Quand le code fonctionne correctement, vous pouvez accélérer son exécution en remplaçant toutes les options de compilation par `-O2`. Pour cela, le plus simple est d'utiliser une macro dans votre makefile.

## 4.1 Structures de données

On suggère fortement d'utiliser les structures de données (tableaux) suivantes :

**Solution numérique**

- `T(i=1:nb_mailles)` tel que `T(i)` $= T_i^n$
- `Tnp1(i=1:nb_mailles)}` tel que `Tnp1(i)` $= T_i^{n+1}$

**Quantités géométriques**

- `aire_maille(1:nb_mailles)}` tel que `aire_maille(i)` $= |\Omega_i|$
- `l_arete(1:nb_aretes)` tel que `l_arete(k)` soit la longueur de l'arête de numéro $k$
- `d_arete(1:nb_aretes)` tel que `d_arete(k)` soit la distance $d_e$ utilisée pour le calcul du flux sur l'arête $e$ de numéro $k$ définie ci-dessus
- `coord_noeud(1:nb_noeuds,1:2)` tel que `coord_noeud(i,1)` et `coord_noeud(i,2)` soient l'abscisse et l'ordonnée du noeud n°$i$
- `milieu_arete(1:nb_aretes,1:2)` tel que `milieu_arete(k,1)` et `milieu_arete(k,2)` soient l'abscisse et l'ordonnée du milieu de l'arête (c.- à-d. les coordonnées du point $x_e$ où $e$ est l'arête de numéro $k$)

**Conditions aux limites**

- `cl_arete(1:nb_aretes)` tel que `cl_arete(k)` soit un entier qui indique la condition aux limites à appliquer à l'arête n°$k$, si celle-ci est une arête de bord. Pour les arêtes intérieures, cette valeur n'est pas utilisée.

Le tableau `cl_arete` est bien plus grand que nécessaire, car il suffirait de le définir pour les arêtes de bord uniquement, mais il permet une implémentation plus simple. Le logiciel de maillage `gmsh` permet d'attribuer un numéro entier à toute les arêtes d'une partie du bord, et ainsi de tenir compte de plusieurs conditions aux limites différentes. C'est à vous de décider quel numéro correspond à telle ou telle condition aux limites, et de maintenir une cohérence entre le numéro donné dans `gmsh` et celui utilisé dans votre programme.

Pour ce TP, on suggère de tenir compte d'au plus 3 conditions aux limites différentes. Ainsi, si vous voulez appliquer 2 conditions de Dirichlet différentes et une condition de Neumann sur 3 parties du bord de $\Omega$ : dans `gmsh`, vous pouvez attribuer les numéro 10, 11, et 20 à ces trois parties (voir le cas test 1 dans la section 6) ; puis dans votre code, sur une arête $k$ telle que `cl_arete(k)` vaut 10, on appliquera la première condition aux limites de Dirichlet. Si elle vaut 11, on appliquera la seconde, et si elle vaut 20, on appliquera la condition aux limites de Neumann.

Pour traiter un nombre de conditions aux limites arbitraire, le code devient plus complexe, cela ne sera pas utilisé ici.

**Informations de connectivité**

- `arete_maille(1:nb_mailles,1:3)` tel que `arete_maille(i,1:3)` contienne les numéros des 3 arêtes du bord de la maille $\Omega_i$ : ces numéros sont stockés en parcourant le bord dans le sens trigonométrique
- `noeud_maille(1:nb_mailles,1:3)` : comme le tableau précédent, pour les noeuds des mailles
- `maille_arete(1:nb_aretes,1:2)` tel que `maille_arete(k,1)` et `maille_arete(k,2)` soient les deux mailles adjacentes à l'arête n°$k$.

Ce dernier tableau possède la propriété suivante : si l'arête n°$k$ est sur le bord, il n'y a qu'une maille adjacente, et le code génère `maille_arete(k,2)=0`. Cela donne donc le moyen de repérer si une arête est intérieure ou sur le bord et permet de traiter les conditions aux limites facilement.

## 4.2 Structure du code

On propose de programmer le schéma avec une structure *par arêtes*, détaillée ci-dessous :

1. lecture des données dans un fichier de données (donnée initiale, conditions aux limites, temps maximum, cfl, nom du fichier de maillage)
2. lecture du maillage
3. allocation et initialisation des tableaux pour `T` et `Tnp1`
4. calcul du pas de temps
5. boucle en temps
   - boucle sur les arêtes
     - calcul du flux sur l'arête (intérieure ou de bord)
     - ajout de la contribution du flux à la température dans les deux ou l'unique maille(s) adjacente(s)
   - mise à jour de `T`
   - sortie fichier

L'étape 2 se fait en appelant la subroutine `maillage` de la façon suivante :

```
call maillage(fichier_maillage, nb_mailles, nb_aretes                    &
     &        , coord_noeud, noeud_maille, aire_maille, l_arete, d_arete &
     &        , milieu_arete, arete_maille, maille_arete, cl_arete)
```

Cette subroutine lit le fichier de maillage, en déduit le nombre de mailles et d'arêtes, puis alloue tous les tableaux donnés en arguments et les remplit correctement. La variable `fichier_maillage` est une chaîne de 30 caractères contenant le nom du fichier de maillage (donné par l'utilisateur dans le fichier de données). Cette subroutine se trouve dans le module `mod_maillage`. Si vous voulez avoir une idée de la façon dont on peut calculer les informations de connectivité, vous pouvez lire cette subroutine et les sous-programmes appelés.

À l'étape 5, à la suite du calcul du flux : attention, le flux a deux contributions *opposées* aux deux mailles adjacentes. Selon la façon dont vous calculez votre flux, il doit apparaître avec un signe $+$ dans une maille, et avec un signe $-$ dans l'autre maille.

En fin d'étape 5, l'appel à la subroutine `sortie` se fait de la façon suivante :

```
call sortie(iter, T, coord_noeud, noeud_maille)
```

Cette subroutine (contenue dans le module `mod_sortie`) écrit dans un fichier au format vtk le maillage et la valeur de $T$ dans chaque maille. La variable `iter` est un compteur entier contenant le numéro de l'itération en temps. Il doit être incrémenté à chaque itération. Ce compteur est utilisé pour créer le nom du fichier de sortie, qui est du type `sortie_[iter].vtk`.

Pour éviter de stocker de trop nombreux fichiers, on pourra se contenter de 10 sorties par simulation, par exemple en définissant `nplot = nmax / 10` et en ne faisant appel à la subroutine de sortie fichier que si `iter` est divisible par `nplot`. Vous pouvez aussi vous contenter d'une seule sortie finale, en particulier pour les maillages les plus fins, car les fichiers de sorties sont en format ascii, et prennent donc beaucoup de place.

## 5 La visualisation

Elle sera faite avec le logiciel Paraview ou VisIt, en visualisant le maillage et le champ de température sous forme de *contours*, voir le TP1 de C++. Avec VisIT ou Paraview, ouvrez la suite de fichiers `sorties_*.vtk` générée par le code : à chaque appel par le programme principal, la

subroutine `sorties_fichier` génère un fichier ascii au format VTK. VisIT et Paraview sont alors capables d'afficher le champ de température correspondant à chacun de ces fichiers, et aussi de permettre une visualisation de l'évolution du champ par une animation. Voir l'annexe D pour une marche à suivre détaillée.

# 6 Les cas tests

Dans cette section, vous trouverez une série de problèmes, appelés cas tests, qui vont vous permettre de programmer votre code pas à pas. Suivez bien toutes les consignes indiquées.

**Cas test 1 : solution 1D dans un carré.** Téléchargez le fichier de maillage `cas_1d.mesh`. Ce fichier contient le maillage du carré $[0,1] \times [0,1]$, avec le code de conditions aux limites 10 pour le bord ouest, le code 11 pour le bord est, et un même code pour les conditions aux limites sur les bords sud et nord (code 20). Le but de cette partie est de résoudre le problème défini par des conditions de Dirichlet sur les bords ouest ($T_b = T_G = 100$) et est ($T_b = T_D = 300$), des conditions de Neumann homogènes sur les bords sud et nord ($\phi_b = 0$), et la donnée initiale $T_{init} = T_G$, voir la figure 1. Les autres paramètres sont $t_{max} = 1$, $D = 1$, et la cfl sera prise égale à 1.

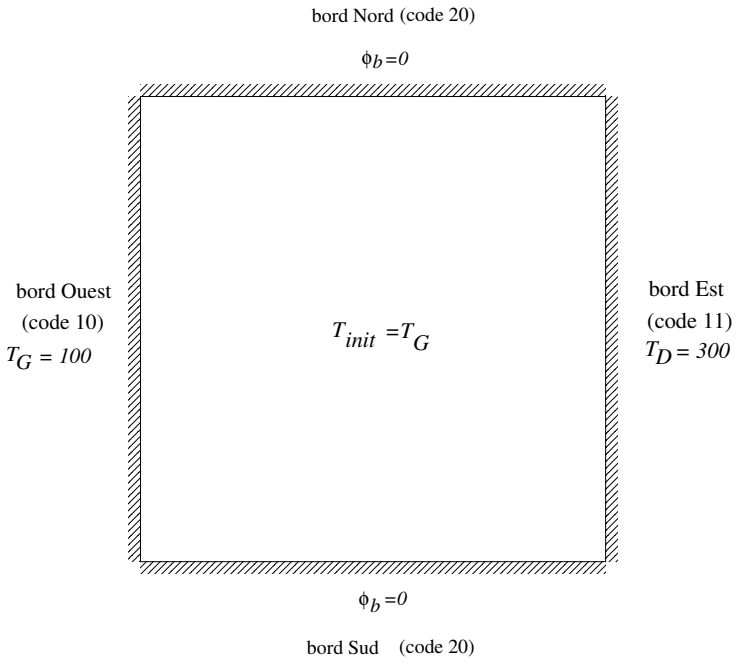



FIGURE 1 – Géométrie du cas test 1, donnée initiale et conditions aux limites.

Commencez par programmer les étapes 1 à 4 du code. L'étape la plus délicate est celle du calcul du pas de temps, basée sur la formule (1) : elle nécessite que vous compreniez bien toutes les structures de données définies section 4.1. Une astuce consiste à d'abord calculer l'inverse de $\Delta t$. Vous devez finalement trouver la valeur suivante :

$$\Delta t = 2.8887438505932329E - 004.$$

Pour cela, il est conseillé d'écrire sur le papier l'algorithme correspondant avec toutes les structures de données nécessaires. Ne passez pas à la suite tant que vous n'avez pas réussi cette étape.

Construisez ensuite le reste du code pour calculer la solution numérique au temps final $t_{max}$. Dans les premiers temps, il est possible que votre code ne fonctionne pas (par exemple avec un message d'erreur du type `floating point exception`) : pour détecter une erreur dans le schéma, faites afficher la norme de $T$ (qui ne doit pas grossir démesurément), visualisez le champ $T$ après la 1ère itération, ou encore, modifiez votre code pour tester une solution constante (voir l'exercice 2 vu en TD).

Vérifiez avec VisIt ou Paraview que la solution est bien 1D (voir l'exercice 3 vu en TD). Comparez votre solution numérique à la solution approchée 1D à 100 modes (calculée avec la méthode de séparation des variables et une décomposition en série de sinus)

$$\theta(t,x) = T_G + (T_D - T_G)\left(x + \sum_{n=1}^{100} \frac{2}{n\pi}(-1)^n \exp(-D(n\pi)^2 t)\sin(n\pi x)\right)$$

Pour cette comparaison, le plus simple est de calculer une erreur absolue dans chaque maille (différence entre la solution numérique et la solution exacte évaluée au centre de la maille), puis une erreur relative globale (avec une norme quelconque). Vous devez avoir une erreur inférieure à 1

Ensuite, faites une analyse de convergence en maillage. Pour cela, il vous faut raffiner le maillage : lisez la marche à suivre donnée en annexe A. Pour chaque maillage, rentrez à la main dans un fichier le nombre de mailles et l'erreur globale. Que constatez-vous ? Voir en annexe B pour une explication.

À la fin de ce cas test, essayez de refaire vous même le maillage en totalité : construisez d'abord le fichier de géométrie (que vous nommerez différemment), en suivant la marche à suivre donnée en annexe C.1. Comparez votre fichier à `cas_1d.geo` (avec un éditeur de texte). Générez ensuite le maillage. Relancez alors votre programme avec ce nouveau maillage.

**Cas test 2 : solution axisymétrique dans un disque.** Créez un maillage du disque de rayon $R = 2$ centré en $(0, 0)$ pour le problème défini par les données suivantes : $T_{init} = 100$, condition de Dirichlet $T_b = 300$ sur le bord du disque, $D = 1$, $t_{max} = 0.1$, puis modifiez votre code pour qu'il résolve ce problème.

Vérifiez que votre solution est bien axisymétrique. Vous pouvez ensuite comparer votre solution numérique à la solution exacte de l'équation 1D axisymétrique correspondante. Celle-ci peut se calculer avec une décomposition en série de fonctions de Bessel (voir le cours *Phénomènes de transfert* au S8) et l'on trouve :

$$T(t,r) = T_b + \sum_{n\geq 1} c_n e^{-\frac{z_n^2}{R^2}Dt} J_0(\frac{r}{R}z_n),$$

où les $c_n$ sont des coefficients de la décomposition, et les $z_n$ sont les zéros de la fonction de Bessel de première espèce $J_0$. Vous trouverez le code fortran correspondant dans la fonction `T1d_axi` et que vous pouvez intégrer à l'un de vos modules.

Une autre possibilité est de calculer une solution numérique de cette équation 1D axisymétrique via un petit programme utilisant un schéma numérique volumes finis pour cette équation 1D (le code n'est pas fourni).

**Cas test 3 : solution axisymétrique dans une couronne.** Créez un maillage de la couronne comprise entre un disque de rayon $R_{int} = 1$ centré en $(0, 0)$ et d'un disque de rayon $R_{ext} = 2$ centré au même point, pour problème défini par les données suivantes : $T_{init} = 1$, conditions de Dirichlet avec $T_b = 1$ sur le bord du disque intérieur et $T_b = 2$ sur le bord du disque extérieur, et $t_{max} = 1$ (voir l'annexe C.2).

Modifiez votre code pour qu'il résolve ce problème, puis calculez la solution numérique de ce problème avec votre code. Là encore, il est possible de calculer une solution exacte. Il est cependant plus simple de calculer la solution exacte en régime stationnaire. Vous pouvez alors comparer à la solution numérique donnée par votre code : il vous faut vérifier que le régime stationnaire est bien atteint, par exemple en mesurant l'écart relatif entre $T^n$ et $T^{n+1}$.

**Cas test 4 : problème de thermique dans un domaine quelconque.** Essayez de créer un domaine de type elliptique, avec deux trous de rayons différents. Faites en sorte de pouvoir utiliser une condition aux limites différentes sur chacun des 3 bords du domaine. Définissez une donnée initiale et des conditions aux limites, et testez votre code sur ce problème. Vous pouvez éventuellement utiliser aussi une solution manufacturée.

Si vous avez le temps, paramétrez votre code pour qu'il fonctionne pour tous les cas tests précédents : il suffit d'indiquer dans le fichier de données combien il y a de bords avec des CL différentes, et quelle valeur de $T_b$ ou de $\phi_b$ est appliquée sur chacun de ces bords. Le code doit alors être capable de traiter ces données automatiquement.

# 7 Consignes pour l'évaluation du TP

À l'issue des 3 séances de TP, vous devez rédiger un rapport répondant aux questions suivantes. Les consignes de dépot du rapport et du votre code sont précisées sur la page moodle du cours.

**Calcul du flux de chaleur à la paroi.** Dans un problème de thermique, imposer les températures à la paroi impose également les flux de chaleur. Ces flux dépendent linéairement des températures de paroi selon une relation non analytique assez complexe (faisant intervenir un opérateur appelé opérateur « Dirichlet-to-Neumann »). Ils peuvent néanmoins être facilement estimés en utilisant la solution numérique calculée par votre code. Il vous est donc demandé, pour le cas test n°3, de calculer l'estimation des flux de chaleur sur les parois des deux cylindres, dans le régime stationnaire. Pour cela :

- rappelez la définition des flux de chaleur sur ces deux parois ;
- proposez une approximation des ces flux utilisant les valeurs numériques calculées par votre code ;
- expliquez comment vous implémentez cette approximation dans votre code ;
- donnez les valeurs calculées par votre code ;
- proposez un moyen de vérifier que ces valeurs sont correctes, et mettez le en oeuvre.

Chacun des items ci-dessus doit faire l'objet d'un paragraphe soigneusement rédigé dans votre rapport.

**Cas test n°4.** Faites une présentation détaillée du problème que vous avez traité pour le cas test n°4 : domaine, maillage, conditions aux limités, etc. Puis commentez brièvement les résultats.

# A Raffinement d'un maillage

On indique ici la marche suivre pour créer plusieurs maillages obtenus par raffinements successifs, pour le premier cas test.

**Création du maillage**

1. lancez `gmsh`, puis sélectionnez le fichier `cas_1d.geo` avec `File / Open`
2. pour éviter la création de triangles non admissibles, nous allons changer l'algorithme de génération (il ne sera pas nécessaire de refaire cette étape par la suite) :
   — dans le menu du haut de `gmsh`, cliquez sur `Tools/Options`
   — dans la fenêtre de dialogue qui s'ouvre, choisissez `Mesh` dans le menu de la colonne de gauche
   — dans le menu de la ligne du haut de cette fenêtre, cliquez sur `General`
   — dans le menu déroulant de la case `2D Algorithm` : remplacez `Automatic` par `Frontal-Delaunay`, puis fermez la fenêtre
3. cliquez sur `Modules/Mesh/Define/2D` : un premier maillage assez grossier, est automatiquement créé
4. cliquez sur `Modules/Mesh/Define/Refine by splitting` deux ou trois fois pour avoir un maillage plus fin
5. si les triangles vous semblent peu admissibles, cliquez sur `Modules/Mesh/Define/Smooth 2D` : il n'y a pas de résultat visible à l'écran, il faut à nouveau cliquer sur `Modules/Mesh/Define/2D`. N'hésitez pas à itérer ce processus plusieurs fois jusqu'à obtenir le résultat voulu.
6. pour générer le fichier de maillage, allez dans le menu `File / export` : dans la fenêtre de dialogue, donnez un nom à votre fichier, par exemple `carre.mesh`, et dans le menu déroulant en bas à droite de la fenêtre, remplacez `Guess From Extension(*.*)` par `Format: Mesh-INRIA Medit (*.mesh)`, puis cliquez sur `Enregistrez`
7. dans la nouvelle fenêtre de dialogue `MESH Options` qui s'ouvre, dans le menu déroulant `Element tag`, remplacez l'option par défaut `Elementary entity` par `Physical entity` (sinon, vous n'aurez pas les codes de CL sur les bords), puis validez avec `OK`. Vous avez alors créé le fichier de maillage `carre.mesh` utilisable par votre code.
8. gardez la session `gmsh` ouverte

**Raffinement du maillage**

1. Cliquez sur `Mesh / Refine by splitting`. Chaque triangle est alors divisé en 4 sous triangles semblables.
2. Sauvegardez dans un fichier avec un nouveau nom pour ne pas perdre le maillage plus grossier (cf ci-dessus)
3. Répétez cette opération pour chaque niveau de raffinement supplémentaire.
4. Attention : si vous fermez `gmsh`, vous serez obligé de recommencer au début. Vous pourrez réouvrir le fichier de maillage précédent pour le raffiner, mais il semble que cela fasse perdre les codes de CL pour les arêtes de bord.

# B Convergence du schéma

Vous devez constater que l'erreur est divisée par 2 à chaque nouveau raffinement de maillage. Cela est dû au fait que le schéma est d'ordre un : on peut montrer que l'erreur est majorée par une constante multipliée par une longueur caractéristique du maillage (par exemple la plus petite hauteur de tous les triangles). Quand on raffine une fois le maillage, l'aire des triangles est divisée par 4, et la hauteur est donc divisée par 2 (l'aire est proportionnelle au carré de la hauteur), et donc l'erreur aussi.

En traçant la courbe de convergence avec `gnuplot`, vous devez aussi constater qu'en échelle log, vous obtenez une droite de pente 1.

# C Création d'une géométrie et d'un maillage

## C.1 Carré (cas test 1)

On décrit ici la création de la géométrie et du maillage pour le domaine carré du cas test 1. Il vous sera facile de faire la même chose pour le disque du cas test 2.

**Création de la géométrie.**

1. Lancez le logiciel gmsh : dans un terminal, tapez
   `gmsh`
2. Création du fichier de géométrie : menu `File/New`, tapez `carre.geo` (toutes les opérations que vous réalisez ensuite sont automatiquement sauvegardées dans le fichier). À la question `Which geometry kernel do you want to use?`, cliquez sur `OpenCASCADE`, mais cela n'est pas important ici.
3. Création des 4 coins du carré :
   menu `Module / Geometry / Elementary entities / Add / Point`
   — dans la fenêtre de dialogue, rentrez les coordonnées du premier point (0,0,0), puis cliquez sur `Add`
   — faites de même pour les 3 autres points, (1,0,0), (1,1,0), (0,1,0)
   — positionnez la souris sur le carré sans cliquer, puis tapez `q` pour quitter ce mode
4. Création des côtés :
   menu `Module / Geometry / Elementary entities / Add / Line`

   — à la souris, sélectionnez le premier point, puis le 2ème (par exemple le coin bas-gauche et le coin bas-droit) : le segment se colore en bleu. Attention, votre souris est bien positionnée sur le point à sélectionner si une bulle apparaît indiquant le numéro du point et ses coordonnées : sinon, votre sélection ne sera pas prise en compte et la ligne ne sera pas créée.
   — faites de même pour sélectionner les 3 autres segments : parcourez votre domaine dans le sens trigonométrique (donc ici [coin bas-droit, coin haut-droit], puis [coin haut-droit, coin haut-gauche], puis enfin [coin haut-gauche, coin bas-gauche])
   — terminez en tapant `q` pour quitter ce mode
5. Vous pouvez regarder dans votre fichier `carre.geo` avec un éditeur de texte : il devrait contenir la définition des 4 coins et des 4 côtés du carré. Sinon, vous avez fait une erreur, recommencez.
6. Création du domaine :
   menu `Module/Geometry/Elementary entities/Add/Plane Surface`

   — à la souris, sélectionnez un des bords du carré : si votre souris est bien positionnée dessus, une bulle apparaît indiquant le numéro du côté et les numéros de ses sommets, vous pouvez alors cliquer et tous les côtés du carré changent de couleur.
   — tapez `e` pour valider ce choix. Deux lignes en traits tirés apparaissent sur le carré. Vous pouvez alors quitter ce mode en tapant `q`.

7. Le fichier carre.geo doit maintenant contenir les lignes

```
Curve Loop(1) = {3, 4, 1, 2};
//+
Plane Surface(1) = {1};
```

8. Définition des codes de CL par bord : on passe du module `Elementary entities` au module `Physical groups` en cliquant sur `Module/Geometry/Physical groups/Add/Curve`

   — on commence par le bord ouest : dans la fenêtre de dialogue, donnez un nom au bord (par exemple « Ouest », mais ce nom ne sera pas utilisé par le code), décochez la case Automatic Numbering, et écrivez le code 10, puis cliquez sur le bord Ouest qui devrait changer de couleur. Tapez ensuite `e` pour passer au bord suivant. Notez que le bord ouest redevient bleu, c'est normal. Au besoin, regardez dans le fichier carre.geo si votre action a été enregistrée, vous devriez vous la ligne `Physical Curve("Ouest", 10) = {4};`
   — Continuez avec le bord est (nom « Est », notez que le code s'est incrémenté tout seul et est passé à 11).
   — Pour les bords sud et nord, on peut les traiter en même temps car on va y appliquer la même CL, identifiée avec le même code 20. Donnez par exemple le nom « NordSud », le code 20 (et pas le code 12 qui s'est affiché automatiquement), puis cliquez sur le bord nord puis sur le sud, avant de valider en tapant`e`. Vous pouvez alors quitter ce mode en tapant `q`

9. Définition du domaine complet :
   menu `Module / Geometry / Physical groups / Add / Surface` Pour sélectionner le carré : dans la fenêtre de dialogue, il faut ici donner un nom et un code qui permette de choisir différentes CI pour différentes parties du domaine, même si cela ne sera pas utilisé dans notre cas. Par exemple, donnez le nom « carré » et le code 100. Puis sélectionnez le carré en cliquant sur une des lignes en trais tirets à l'intérieur du domaine, puis en tapant `e`, en enfin en tapant `q` pour quitter ce mode.
10. Vous devriez vous la nouvelle ligne suivante dans votre fichier : `Physical Surface("carre", 100) = 1;`

**Création du maillage**

1. cliquez sur `Modules/Mesh/Define/2D` : un premier maillage assez grossier, est automatiquement créé
2. cliquez sur `Modules/Mesh/Define/Refine by splitting` deux ou trois fois pour avoir un maillage plus fin
3. si les triangles vous semblent peu admissibles, cliquez sur `Modules/Mesh/Define/Smooth 2D` : il n'y a pas de résultat visible à l'écran, il faut à nouveau cliquer sur `Modules/Mesh/Define/2D`. N'hésitez pas à itérer ce processus plusieurs fois jusqu'à obtenir le résultat voulu.
4. pour générer le fichier de maillage, allez dans le menu `File / export` : dans la fenêtre de dialogue, donnez un nom à votre fichier, par exemple `carre.mesh`, et dans le menu déroulant en bas à droite de la fenêtre, remplacez `Guess From Extension(*.*)` par `Format: Mesh-INRIA Medit (*.mesh)`, puis cliquez sur `Enregistrez`
5. dans la nouvelle fenêtre de dialogue `MESH Options` qui s'ouvre, dans le menu déroulant `Element tag`, remplacez l'option par défaut `Elementary entity` par `Physical entity` (sinon, vous n'aurez pas les codes de CL sur les bords), puis validez avec `OK`. Vous avez alors créé le fichier de maillage `carre.mesh` utilisable par votre code.

## C.2 Couronne (cas test 3)

**Création de la géométrie.**

1. Lancez le logiciel gmsh : dans un terminal, tapez
   `gmsh`
2. Création du fichier de géométrie : cliquez sur `File / New` et donnez le nom `couronne.geo` à votre nouveau fichier (toutes les opérations que vous réalisez ensuite sont automatiquement sauvegardées dans le fichier). À la question `Which geometry kernel do you want to use?`, cliquez sur `OpenCASCADE`, mais cela n'est pas important ici.
3. Création des deux bords circulaires :
   `menu Module / Geometry / Elementary entities / Add / circle`
   - pour définir le cercle intérieur : dans la fenêtre de dialogue, donnez les coordonnées du centre et le rayon, cliquez ensuite sur `add` pour créer le cercle (ces opérations peuvent aussi se faire avec la souris uniquement, il faut alors taper `e` à la fin pour valider la création du cercle).
   - idem pour le cercle extérieur. Puis quittez ce mode en tapant `q`.
4. Création du domaine :
   menu `Module / Geometry / Elementary entities / Add / Plane Surface`
   - positionnez votre souris sur le cercle extérieur pour qu'une bulle apparaisse contenant les informations sur ce cercle, puis cliquez avec la souris pour le sélectionner (il change alors de couleur)
   - faites de même avec le cercle intérieur (c'est le "trou" (hole) mentionné par gmsh)
   - tapez `e` pour valider la sélection du domaine. Deux lignes en traits tirés apparaissent sur la couronne. Vous pouvez alors quitter ce mode en tapant `q`.
5. Définition des codes de CL par bord :
   cliquez sur le menu `Module / Physical groups / Add / Curve`
   - pour attribuer le code 10 au cercle intérieur : dans la fenêtre de dialogue, donnez un nom au bord considéré (par exemple, "cercle_int", ce nom ne sera pas utilisé par la suite), décochez la case `Automatic Numbering`, et donnez le numéro que vous voulez attribuer à ce bord (ici, on suggère 10). Puis cliquez sur le bord (comme à l'étape précédente, une bulle apparaît lorsque votre souris est bien positionnée sur le cercle, et vous pouvez alors cliquer), et validez en tapant `e`.
   - Même procédure pour attribuer le n° 11 au cercle extérieur (pour pouvoir utiliser deux CL différentes). Puis quittez ce mode en tapant `q`.
6. Définition du domaine complet :
   cliquez sur le menu `Module / Physical groups / Add / Surface` pour sélectionner le domaine (ici, la couronne) : dans la fenêtre de dialogue, il faut ici donner un nom et un code qui permette de choisir différentes CI pour différentes parties du domaine, même si cela ne sera pas utilisé dans notre cas. Par exemple, donnez le nom « couronne » et le code 100. Puis sélectionnez la couronne en cliquant sur une des lignes en trais tirets à l'intérieur du domaine (là encore, une bulle apparaît quand votre souris est correctement positionnée sur une de ces lignes), puis tapez `e`, en enfin tapez `q` pour quitter ce mode.

**Création du maillage**

1. cliquez sur `Mesh / 2D` : un premier maillage assez grossier, est automatiquement créé
2. cliquez sur `Mesh / Refine by splitting` une ou deux fois pour avoir un maillage plus fin

3. si les triangles vous semblent peu admissibles, cliquez sur `Mesh / Smooth 2D` puis à nouveau sur `Mesh / 2D`. N'hésitez pas à itérer ce processus plusieurs fois.
4. pour générer le fichier de maillage, allez dans le menu `File / export` : dans la fenêtre de dialogue, donnez un nom à votre fichier, par exemple `couronne.mesh`, et dans le menu déroulant en bas à droite de la fenêtre, remplacez `Guess From Extension(*.*)` par `Format: Mesh-INRIA Medit (*.mesh)`, puis cliquez sur `Enregistrez`
5. dans la nouvelle fenêtre de dialogue `MESH Options` qui s'ouvre, dans le menu déroulant `Element tag`, remplacez l'option par défaut `Elementary entity` par `Physical entity` (sinon, vous n'aurez pas les codes de CL sur les bords), puis validez avec `OK`. Vous avez alors créé le fichier de maillage `couronne.mesh` utilisable par votre code.

# D Aide au démarrage avec VisIT

On donne ici les premières commandes pour visualiser le maillage et le champ de température. N'hésitez pas à explorer le logiciel pour d'autres traitements de vos résultats.

1. lancez la commande `visit` dans un terminal
2. cliquez sur l'onglet `fichier`, puis dans le menu déroulant sur `ouvrir fichier`, puis cliquez sur la base de données `sortie_*.vtk` : cette base est la liste de vos fichier sorties (ici tous les noms sont supposés commencer par `sortie_`, suivi d'un numéro, puis de l'extension `.vtk` ;
3. dans le cadre `Tracés` de la fenêtre de contrôle, cliquez sur l'onglet `Ajouter`, puis cliquez sur `Maillage`, puis sur `mesh` ;
4. dans le même cadre `Tracés`, cliquez maintenant sur l'onglet `Tracer`, vous devez alors voir apparaître votre domaine de calcul et son maillage dans la fenêtre de droite ;
5. toujours dans le cadre `Tracés`, cliquez sur `Ajouter`, puis cliquez sur `Ajouter`, puis sur `Pseudo Couleur` puis sur `T`, qui est le nom du champ que vous voulez visualiser ;
6. dans le même cadre `Tracés`, cliquez maintenant sur l'onglet `Tracer`, vous devez alors voir le champ de température apparaître dans la fenêtre de droite, en plus du maillage, sous forme d'une carte de couleurs : chaque maille est colorée en fonction de la valeur de la température dans la maille, et le code couleur correspondant est affiché dans la petite barre sur le côté ;
7. ce que vous voyez là est le champ de température stocké dans le premier fichier de sortie : pour visualiser le champ à des temps ultérieurs, utilisez les boutons de contrôle de l'animation dans le cadre `Temps` juste au dessus du cadre `Tracés`.