

TPs : Volumes finis pour les problèmes elliptiques

F. Boyer

Institut de Mathématiques de Marseille
Aix-Marseille Université

- ① Résolution du Laplacien Dirichlet homogène (**fourni**) :
 - Prise en main, principes de mise en oeuvre.
 - Tracé des solutions exactes et approchées pour certains jeux de données.
 - Tracé des courbes d'erreur.
 - Réécriture du schéma en assemblant les termes **par interface**.
- ② Programmation des conditions de Dirichlet non-homogène
- ③ Programmation des conditions de Neumann non-homogène
- ④ Programmation des conditions de Robin (ou de Fourier)

- 3 bibliothèques de fonctions :
 - `donnees1D.sci` Différents jeux de données
 - `maillages1D.sci` Création de maillages, calculs de normes discrètes
 - `schemas1D.sci` Fonctions d'assemblage des schémas
- 2 programmes principaux :
 - `VF1D.sce` Calcul et tracé des solutions approchées/exactes
 - `courbes_erreur1D.sce` Tracé de courbes d'erreurs

JEUX DE DONNÉES

- Un jeu de données `donnees` est une **structure** qui contient
 - `donnees.nom` : le nom du cas test.
 - `donnees.source` : le terme source (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
 - `donnees.bordD` : les données au bord (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
 - `donnees.uexacte` : la solution exacte (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
 - `donnees.coeff_k` : le coeff de diffusion (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
- Certains contiennent d'autres informations...

JEUX DE DONNÉES

- Un jeu de données `donnees` est une **structure** qui contient
 - `donnees.nom` : le nom du cas test.
 - `donnees.source` : le terme source (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
 - `donnees.bordD` : les données au bord (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
 - `donnees.uexacte` : la solution exacte (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
 - `donnees.coeff_k` : le coeff de diffusion (une fonction $\mathbb{R} \rightarrow \mathbb{R}$).
- Certains contiennent d'autres informations...
- Liste de jeux de données : `cas_test=list(...,...,...)`
Pour tout entier `i` convenable `cas_test(i)` est un jeu de données.
- Pour rajouter un jeu de données :

```
cas_test($+1)=struct("nom",xxxx,...  
                    "uexacte",xxxx,...  
                    "bordD",xxxx,...  
                    "source",xxxx,...  
                    "coeff_k",xxxx);
```

MAILLAGES :

- Forme générique des fonctions de création

```
function [maillage] = maillage_XXXX(N);
```

Avec : `XXXX` \in {uniforme , alterne , aleatoire , stretch}

MAILLAGES :

- Forme générique des fonctions de création

```
function [maillage] = maillage_XXXX(N);
```

Avec : `XXXX` \in {uniforme , alterne , aleatoire , stretch}

- Chacune renvoie une structure `maillage` contenant

- `maillage.nom` : le nom du maillage.
- `maillage.nb_vol` : le nombre de volumes de contrôle.
- `maillage.centres` : coordonnées des centres

$$\text{maillage.centres}(i) = x_i, \quad 1 \leq i \leq \text{maillage.nb_vol}$$

- `maillage.sommets` : coordonnées des sommets

$$\text{maillage.sommets}(i) = x_{i-1/2}, \quad 1 \leq i \leq \text{maillage.nb_vol}+1$$

- `maillage.mes` : mesures des volumes de contrôles

$$\text{maillage.mes}(i) = h_i, \quad 1 \leq i \leq \text{maillage.nb_vol}$$

- `maillage.dist` : distances entre centres voisins

$$\text{maillage.dist}(i) = h_{i-1/2}, \quad 1 \leq i \leq \text{maillage.nb_vol}+1$$

EXEMPLES D'UTILISATION

On suppose `donnees` et `maillage` connus

- Evaluation du terme source au centre des mailles

```
source_centres=feval(maillage.centres,donnees.source);
```

EXEMPLES D'UTILISATION

On suppose `donnees` et `maillage` connus

- Evaluation du terme source au centre des mailles

```
source_centres=feval(maillage.centres,donnees.source);
```

- Evaluation du coefficient de diffusion aux interfaces

```
ck=feval(maillage.sommets,donnees.coeff_k);
```

EXEMPLES D'UTILISATION

On suppose `donnees` et `maillage` connus

- Evaluation du terme source au centre des mailles
`source_centres=feval(maillage.centres,donnees.source);`
- Evaluation du coefficient de diffusion aux interfaces
`ck=feval(maillage.sommets,donnees.coeff_k);`
- Evaluation de la solution exacte au centre des mailles, si elle est fournie
`if (isfield(donnees,'uexacte')) then`
`solexacte=feval(maillage.centres,donnees.uexacte);`
`end;`

FONCTIONS D'ASSEMBLAGE DES SCHÉMAS

- Syntaxe générale : $[A, b] = \text{vf_xxxx}(m, \text{donnees})$
 - Entrée : m est un maillage et donnees un jeu de données.
 - Sortie : A est la matrice du système et b le second membre.
- Est codée pour l'instant :

- $\text{xxxx} = \text{laplacien_dirh}$:

$$-u'' = f, \text{ avec } u(0) = u(1) = 0.$$

- **Il reste à coder dans ce TP1 :**

- $\text{xxxx} = \text{laplacien_dirnh}$:

$$-u'' = f, \text{ avec } \begin{cases} u(0) = \text{bordD}(0) \\ u(1) = \text{bordD}(1) \end{cases}$$

- $\text{xxxx} = \text{diffusion_dirnh}$:

$$-\partial_x(k(x)\partial_x u) = f, \text{ avec } \begin{cases} u(0) = \text{bordD}(0) \\ u(1) = \text{bordD}(1) \end{cases}$$

- $\text{xxxx} = \text{diffusion_neunh}$:

$$-\partial_x(k(x)\partial_x u) = f, \text{ avec } \begin{cases} -k(0)u'(0) = \text{bordN}(0) \\ k(1)u'(1) = \text{bordN}(1) \end{cases}$$

- $\text{xxxx} = \text{diffusion_fournh}$:

$$-\partial_x(k(x)\partial_x u) = f, \text{ avec } \begin{cases} \text{coeffF} \times u(0) - k(0)u'(0) = \text{bordF}(0) \\ \text{coeffF} \times u(1) + k(1)u'(1) = \text{bordF}(1) \end{cases}$$

STRUCTURE DU PROGRAMME VF1D.sce

- Chargement des libraires de fonctions.
- Interrogation utilisateur :
 - Choix du cas test
 - Choix du problème à résoudre et de variantes du schémas.
 - Choix du maillage
 - Choix du nombre de volumes de contrôle
- Assemblage du schéma
- Résolution du système linéaire (par UMFPACK si Scilab \geq 5.2)
- Tracé de la solution approchée (et exacte si disponible) et évaluation de l'erreur.

STRUCTURE DU PROGRAMME `courbes_erreur1D.sce`

- Chargement des libraires de fonctions.
- Interrogation utilisateur :
 - Choix du cas test
 - Choix du problème à résoudre et de variantes du schémas.
 - Choix du maillage
 - Choix du nombre de volumes de contrôle du maillage grossier.
 - Choix de l'incrément du nombre de volumes de contrôle d'un maillage à l'autre.
 - Choix du nombre de volumes de contrôle maximal souhaité.
- Pour chaque taille de maillage :
 - Assemblage du schéma
 - Résolution du système linéaire (par UMFPACK si `Scilab` ≥ 5.2)
 - Evaluation et stockage de l'erreur dans différentes normes.
- Tracé des courbes d'erreur en fonction du pas du maillage en échelle logarithmique.

- ① Résolution du Laplacien Dirichlet homogène (**fourni**) :
`vf_laplacien_dirh`
 - Prise en main, principes de mise en oeuvre.
 - Tracé des solutions exactes et approchées pour certains jeux de données.
 - Tracé des courbes d'erreur.
 - Réécriture du schéma en assemblant les termes **par interface**.
- ② Programmation des conditions de Dirichlet non-homogène
`vf_laplacien_dirnh`
- ③ Ajout d'un coefficient de diffusion variable régulier $k(x)$:
`vf_diffusion_dirnh`
- ④ Programmation des conditions de Neumann non-homogène
`vf_diffusion_neunh`
- ⑤ Programmation des conditions de Fourier non-homogène
`vf_diffusion_fournh`

PROBLÈME DE DIFFUSION À COEFFICIENT DISCONTINU

- Programmation des différents choix du flux à l'interface :
 - Si `donnee.methode='exacte'` :

$$k_{i+1/2} = k(x_{i+1/2}).$$

- Si `donnee.methode='arithmetique'` :

$$k_{i+1/2} = \frac{k(x_i) + k(x_{i+1})}{2}.$$

- Si `donnee.methode='harmonique'` :

$$k_{i+1/2} = \frac{h_{i+1/2}k(x_i)k(x_{i+1})}{(x_{i+1} - x_{i+1/2})k(x_i) + (x_{i+1/2} - x_i)k(x_{i+1})}.$$

- Comparaison des ordres de convergence observés.

- ① Prise en main du format de maillage et des jeux de données utilisés (**fourni**).
- ② Programmation de VF4 :
 - Avec conditions aux limites de Dirichlet homogène
 - Avec conditions aux limites de Dirichlet non-homogène
 - Avec conditions aux limites mixtes Dirichlet / Neumann
- ③ Tests de convergence sur diverses familles de maillages et des solutions analytiques.
- ④ Utilité de la condition d'orthogonalité

- 3 bibliothèques de fonctions :

- `donnees2D.sci`

Différents jeux de données

- `maillages2D.sci`

Définition des maillages disponibles, chargement et manipulation.
Calculs de normes discrètes. Routines de tracé des fonctions discrètes.

- `schemas2D.sci`

Fonctions d'assemblage des schémas

- 2 programmes principaux :

- `VF2D.sce`

Calcul et tracé des solutions approchées/exactes

- `courbes_erreur2D.sce`

Tracé de courbes d'erreurs

LES MAILLAGES 2D :

- Stockés dans 3 fichiers (**fournis**) :
 - **xxxx_sommets**
 - **xxxx_centres**
 - **xxxx_aretes**
- Sont chargés dans Scilab par la commande

```
[maillage]=lecture_maillage(xxxx);
```

- La variable `maillage` ainsi obtenue est une structure qui contient
 - `maillage.nom` : un nom qui décrit le maillage
 - `maillage.nb_vol` : nombre de volumes de contrôle
 - `maillage.nb_som` : nombre de sommets
 - `maillage.nb_are` : nombre d'arêtes
 - `maillage.sommets` : une matrice de taille `nb_som x 2`
 - `maillage.centres` : une matrice de taille `nb_vol x 2`
 - `maillage.arettes` : une matrice de taille `nb_are x 17`

- Sommets : coordonnées du sommet numéro i

`maillage.sommets(i,_X)`, `maillage.sommets(i,_Y)`

- Centres : coordonnées du centre du volume j

`maillage.centres(j,_X)`, `maillage.centres(j,_Y)`

- Arêtes : Pour l'arête numéro k

- Numéros des deux sommets

`maillage.arettes(k,_DEB)`, `maillage.arettes(k,_FIN)`

- Numéros des deux volumes κ et \mathcal{L}

`maillage.arettes(k,_K)`, `maillage.arettes(k,_L)`

- Mesure de l'arête $|\sigma|$ et distance $d_{\kappa\mathcal{L}}$

`maillage.arettes(k,_MES)`, `maillage.arettes(k,_DKL)`

- Mesures des quarts de diamant

`maillage.arettes(k,_MES_K_DEB)`, ...

- Label (≥ 0 à l'intérieur, $= -1$ bord Dirichlet, < -1 bord Neumann)

`maillage.arettes(k,_LABEL)`.

AUTRES FONCTIONS UTILES :

- Pour évaluer une fonction sur les sommets ou sur les centres

```
eval_fonction(m.centres,f)
```

```
eval_fonction(m.sommets,f)
```

- Pour dessiner une fonction constante par mailles u^T

```
trace_fonction(m,u,nwin,ep_maillage)
```

- m est le maillage
 - u un vecteur colonne de taille $m.nb_vol$ contenant les valeurs
 - $nwin$ est le numéro de la fenêtre graphique où l'on veut tracer
 - $ep_maillage$ est l'épaisseur du trait pour dessiner le maillage (0 si on ne veut pas afficher le maillage).
- Pour calculer la norme L^2 de u^T

```
norme_L2(m,u)
```

- Pour calculer une norme H_0^1 discrète de u^T

```
norme_H1(m,u)
```

LES JEUX DE DONNÉES :

Structure générale similaire au cas 1D

- Liste de jeux de données : `cas_test=list(...,...,...)`
- Chaque variable de cette liste est une structure `donnees`
 - `donnees.nom`
 - `donnees.source`
 - `donnees.bordD`
 - `donnees.bordN`
 - `donnees.coeff_k`
 - `donnees.uexacte`
 - `donnees.maillages` : donne (éventuellement) une liste de noms de maillages spécifiquement utilisables pour ce cas test.
 - + d'autres paramètres éventuellement utiles pour le calcul.

FONCTIONS D'ASSEMBLAGE DES SCHÉMAS

- Syntaxe générale : $[A, b] = \text{const_schema_xxxx}(m, \text{donnees})$
 - Entrée : m est un maillage et donnees un jeu de données.
 - Sortie : A est la matrice du système et b le second membre.
- **Deux schémas prévus mais ne sont pas codés :**
 - **xxxx=VF4**
 - **xxxx=DDFV**

STRUCTURE DU PROGRAMME VF2D.sce

- Chargement des libraires de fonctions.
- Interrogation utilisateur :
 - Choix du cas test
 - Choix du maillage (la liste des maillages proposée peut dépendre du cas test)
 - S'il s'agit d'une famille de maillages : choix du niveau de raffinement.
 - Choix du schéma à utiliser (VF4 ou DDFV).
- Lecture du maillage
- Assemblage du schéma
- Résolution du système linéaire (par UMFPACK)
- Tracé de la solution approchée (et exacte si disponible) et évaluation de l'erreur.
- Eventuellement : tracé de la fonction de courant.

STRUCTURE DU PROGRAMME `courbes_erreur2D.sce`

- Chargement des libraires de fonctions.
- Interrogation utilisateur :
 - Choix du cas test
 - Choix d'une **famille** de maillages
 - Choix des niveaux de raffinement min et max souhaités
 - Choix du schéma à utiliser (VF4 ou DDFV).
- Pour chaque maillage choisi :
 - Lecture du maillage
 - Assemblage du schéma
 - Résolution du système linéaire (par UMFPACK)
 - Evaluation et stockage de l'erreur.
- Tracé des courbes d'erreur en fonction du pas du maillage en échelle logarithmique.