

# **Listing des programmes**

## **Projet Numérique**

### **Différences Finies**

**MONTREUIL Yannis**  
**ATHARI Keivan**

# Table des matières

<b>I</b>	<b>Listing des fichiers</b>	<b>2</b>
<b>1</b>	<b>Maillage classique</b>	<b>2</b>
1.1	consistance.f90 . . . . .	2
1.2	convergence.f90 . . . . .	2
1.3	sor_classique.f90 . . . . .	2
<b>2</b>	<b>Maillage décalé</b>	<b>2</b>
2.1	difference.f90 . . . . .	2
2.2	convergence_decalle.f90 . . . . .	2
2.3	sor_decalle.f90 . . . . .	3
<b>II</b>	<b>Listing des programmes</b>	<b>3</b>
<b>3</b>	<b>Programme d'initialisation des paramètres</b>	<b>3</b>
3.1	parametre_mod . . . . .	3
3.2	initialisation . . . . .	3
3.3	initialisation_maillage . . . . .	3
3.4	initialisation_vecteurs . . . . .	3
<b>4</b>	<b>fonction récursive</b>	<b>3</b>
4.1	function phi . . . . .	3
4.2	function fi . . . . .	3
<b>5</b>	<b>Stockage des données</b>	<b>4</b>
5.1	ecriture . . . . .	4
5.2	ecriture2 . . . . .	4
<b>6</b>	<b>Méthodes de resolution</b>	<b>4</b>
6.1	Module LU . . . . .	4
6.1.1	initialisation_LU . . . . .	4
6.1.2	descente_LU . . . . .	4
6.1.3	remontee_LU . . . . .	4
6.2	Module SOR . . . . .	4
6.2.1	sor . . . . .	4
<b>7</b>	<b>Ordres et erreurs</b>	<b>4</b>
7.1	erreur . . . . .	4
7.2	ordre_methode . . . . .	5

<b>8 Fonctions Fortran</b>	<b>5</b>
8.1 CPU_time . . . . .	5

## Première partie

# Listing des fichiers

## 1 Maillage classique

### 1.1 consistance.f90

- 'trace\_ordre\_h\_const.gnu'
- script traçant const en fonction de h en lin/log, et sauvegarde au nom 'ordre\_dec\_h\_const.png'
- 'trace\_ordre\_h\_ninf.gnu'
- script traçant ninf en fonction de h en log/log, et sauvegarde au nom 'ordre\_dec\_h\_ninf.png'

### 1.2 convergence.f90

- 'trace\_conv\_clas\_LU.gnu'
- trace convergence via LU, et sauvegarde au nom 'Solutions\_LU\_dec.png'

### 1.3 sor\_classique.f90

- 'trace\_conv\_clas\_SOR.gnu'
- trace convergence via SOR, et sauvegarde au nom 'Solutions\_SOR\_dec.png'

## 2 Maillage décalé

### 2.1 difference.f90

- 'trace\_ordre2\_h\_const.gnu'
- script traçant const en fonction de h en lin/log, et sauvegarde au nom 'ordre\_h\_const.png'
- 'trace\_ordre2\_h\_ninf.gnu'
- script traçant ninf en fonction de h en log/log, et sauvegarde au nom 'ordre\_h\_ninf.png'

### 2.2 convergence\_decalle.f90

- 'trace\_conv\_dec\_LU.gnu'
- trace convergence via LU, et sauvegarde au nom 'Solutions\_LU.png'

### 2.3 sor\_decalle.f90

- 'trace\_conv\_dec\_SOR.gnu'
- trace convergence via SOR, et sauvegarde au nom 'Solutions\_SOR.png'

## Deuxième partie

# Listing des programmes

## 3 Programme d'initialisation des paramètres

### 3.1 parametre\_mod

- Permet d'initialiser les constantes de notre problème
- $nx$  en entrée

### 3.2 initialisation

- Initialise les coefficients des matrices calculé via la discrétisation de notre équation
- $h$  en entrée

### 3.3 initialisation\_maillage

- Initialise le maillage classique, on calcule  $x_i$
- $h$  en entrée

### 3.4 initialisation\_vecteurs

- Initialise les vecteurs de notre matrice tridiagonale permettant de résoudre notre équation discrète
- $x_i$  et  $h$  en entrée

## 4 fonction récursive

### 4.1 function phi

- Calcul la solution exacte de notre équation continu
- $x_i$  en entrée

### 4.2 function fi

- Calcul la fonction  $f(x_i)$
- $x_i$  en entrée

## 5 Stockage des données

### 5.1 `ecriture`

Permet de créer, d'ouvrir et écrire dans un fichier en fonction du nom du fichier. On écrit  $v_1, f_i, \phi_i, v_2$   
—  $v_1, v_2$  et  $nom_{fichier}$  en entrée

### 5.2 `ecriture2`

Permet de créer, d'ouvrir et écrire dans un fichier en fonction du nom du fichier. On écrit  $v_1$  et  $v_2$   
—  $v_1, v_2$  et  $nom_{fichier}$  en entrée

## 6 Méthodes de resolution

### 6.1 Module LU

#### 6.1.1 `initialisation_LU`

On initialise les vecteurs de la matrice LU simplifié  
—  $nx, A_w, A_p, A_e$  en entrée

#### 6.1.2 `descente_LU`

On effectue la descente  $Lv = b$   
—  $nx, b, L_p, L_w$  en entrée

#### 6.1.3 `remontee_LU`

On effectue la remontée  $Uu = v$ , on détermine alors la solution  $u$   
—  $nx, U, y$  en entrée

### 6.2 Module SOR

#### 6.2.1 `sor`

On effectue la résolution du schéma avec la méthode SOR, on calcule alors  $u_s$   
—  $nx, w, A_p, A_w, A_e, b, u_s^{(k+1)}, u_s, dU$  en entrée

## 7 Ordres et erreurs

### 7.1 `erreur`

Calcul l'erreur de consistance du schéma  
—  $x_i$  et  $h$  en entrée

## 7.2 ordre\_methode

On calcule les erreurs de convergence  $E_c$  et de consistance  $E_u$   
—  $nx, h, u, x_i$  en entrée

# 8 Fonctions Fortran

## 8.1 CPU\_time

Fonction interne à fortran. Permet de connaitre le temps d'exécution d'un programme.