



Rapport de Stage

Landa MCHAREK

Mars/Aout 2009

Intégration de la méthode $\mathcal{L.D.C}$ pour l'étude de la méthode du schéma en temps pararéel

Encadrante : Juliet RYAN

The French Aerospace Lab



Office National d'Études et Recherches Aéronautiques.

Remerciements

Je tiens particulièrement remercier Madame Juliet RYAN, ma tutrice, pour m'avoir bien encadrer durant mon stage passé à l'entreprise ONERA à Chatillon. Je la remercie pour l'aide et les consiels concernant les missions évoquées dans ce rapprot, qu'elle m'a apportée lors des différents suivis. Le temps qu'elle a consacré tout au long de cette période, sachant répondre à toutes mes interrogations et sans oublier sa participation au cheminement de ce rapport.

Je souhaite également adresser mes remerciements à Monsieur François Xavier ROUX, responsable du service Calcul Haute Performance, et tout le personnel du département de traitement de l'information et modélisation auquel j'étais rattachée, pour leur accueil, leur confiance et leur sympathie qu'ils m'ont raccordé dès mon arrivée à l'ONERA.

Enfin, je veux remercier Monsieur Olivier LAFITTE et tous les professeurs de la formation MACS, Mathmatiques appliquées et calcul scientifique, pour leur aide et leur encouragement durant ses trois années en école ingénieur Sup-Galilée.

Résumé

Dans cette article, une méthode de \mathcal{LDC} , Local Defect Correction, est présentée pour le problème instationnaire de convection-diffusion 1D et 2D. La méthode est adaptée avec le schéma en temps "pararéal" pour définir les conditions initiales sur les sous domaines en temps. Le problème est résolu par une grille globale grossière et uniforme et plusieurs grilles locales fines et uniformes. Les approximations grossières et fines sont améliorées itérativement. Résultats des exemples numériques illustre la précision et l'efficacité de cette méthode.

Mots clés : *local defect correction, schéma en temps "pararéal", solveur grille uniforme, grille uniforme grossière, grille uniforme fine, sous domaine.*

Abstract

In this article, a technique of LDC, Local Defect Correction is presented for unsteady convection-diffusion problem 1D and 2D. The method is adapted with the scheme of "parar  el" to define the initial condition on the subdomains of time. The problem is solved by means of a global uniform coarse grid and several of local uniform fine grid. The global and local approximation are improved iteratively. Results of numerical experiments demonstrates the accuracy and effectiveness of this method.

Keywords : *Local defect correction, "parar  el" scheme, uniform grid solver, uniform coarse grid, uniform fine grid, subdomain.*

<i>Remerciements</i>	2
<i>Introduction</i>	4
<i>Présentaion de l'ONERA</i>	7
1 Résolution d'une équation de convection-diffusion 1D	9
1.1 Position du problème	9
1.2 Existence et unicité d'une solution du problème	10
1.3 Formulation variationnelle	10
1.4 Résolution du problème	11
1.4.1 Discrétisation du problème en espace	11
1.4.2 Euler Explicite	13
1.4.3 Runge Kutta d'ordre 2,3 et 4	18
1.5 Méthode <i>LDC</i>	19
1.5.1 Problème (1) sur la grille grossière	20
1.5.2 L'itération de <i>LDC</i>	20
1.5.3 Propriétés de la méthode <i>LDC</i>	27
1.6 Applications Numériques	28
1.6.1 Exemple 1	28
1.6.2 Exemple 2	33
2 Problème de convection-diffusion en deux dimensions	37
2.1 Résolution numérique d'EDP	37
2.1.1 Discrétisation du problème en espace	38
2.1.2 Euler Explicite	43
Condition de stabilité du schéma Euler Explicite	44
2.1.3 Runge Kutta 4	45
2.2 Méthode <i>LDC</i>	46
2.3 Application Numérique	48
2.4 Conclusion	52
Conclusion	52
A Résolution d'EDP par un schéma en temps "pararéel"	55

1.1	Exemple d'un sous domaine avec une région de sécurité	24
1.2	Tracés des solutions approchées calculées par Euler explicite et la solution exacte, à $t_{N+1} = 1$	31
1.3	Solution exacte à $t_{N+1} = 1$ et solution numérique calculée par \mathcal{LDC}	32
1.4	Comparaison entre la méthode \mathcal{LDC} et solveur de grille uniforme	32
1.5	Solutions approchées par Runge Kutta 4 et solution exacte en fonction du pas d'espace à $t_{N+1} = 1$	34
1.6	Comparaison entre \mathcal{LDC} et solveur de grille uniforme en utilisant RK4	34
1.7	Comparaison entre \mathcal{LDC} et solveur de grille uniforme en utilisant Euler Explicite pour $\tau = 1$	35
1.8	Comparaison entre \mathcal{LDC} et solveur de grille uniforme en utilisant Euler Explicite pour $\tau = 5$	36
2.1	Solution exacte, solution approchée calculée par \mathcal{LDC} et celle calculée par le solveur de grille uniforme pour $\Delta t = 0.0625$, $\delta t = 0$	
2.2	Solution exacte, solution approchée calculée par \mathcal{LDC} et celle calculée par le solveur de grille uniforme pour $\Delta t = 0.03125$, $\delta t = 0$	

Les solutions d'équations aux dérivées partielles (EDP) sont souvent caractérisées, à chaque moment, par des régions où les pentes sont plus importantes par rapport à ceux du reste de domaine(où la solution présente un bon comportement).

Des exemples sont souvent présents dans le domaine de choc hydrodynamique, les processus de combustion, etc.

La méthode **LDC**, *Local Defect Correction*, a été introduite par Hackbusch [1] pour résoudre des EDP elliptiques avec les schémas implicites. Cette méthode a été étudiée avec les différentes techniques de discrétisation:

- Différences finis par Ferket et Reusken [2].
- Éléments finis par Wappler [3].
- Volumes finis par Anthonissen [4], Hof et Reusken [5].

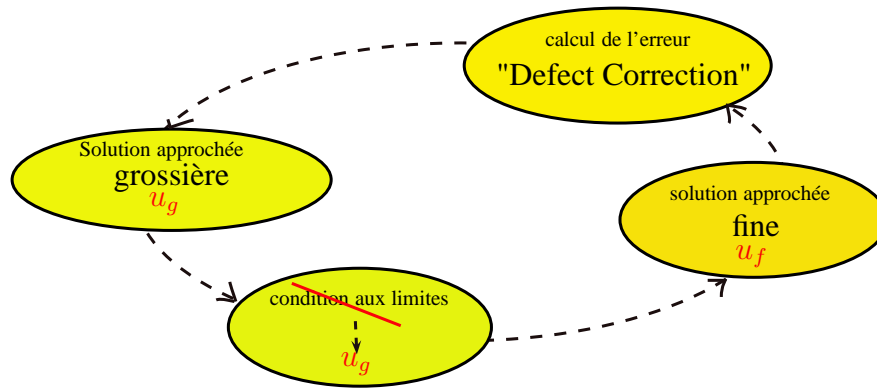
Cette méthode a été ensuite appliquée aux plusieurs types de grilles par Nefedov et Mattheij [6]. **LDC** a été étendue pour inclure plusieurs niveaux de raffinements et décomposition de domaine.

Ensuite, **LDC** a été généralisée pour résoudre les EDP paraboliques par Minero, Anthonissen et Mattheij [7].

Voici les étapes de cette méthode:

1. On résout sur la grille grossière une solution approchée.
2. Cette solution approchée est ensuite utilisée pour initialiser la solution approchée sur la grille fine comme condition aux limites là où la solution continue n'est pas définie ou moins lisse.
3. Une solution locale est ensuite calculée avec un pas de temps plus petit que celui adapté sur la grille grossière.
4. Dans la région où la solution continue est moins lisse, la solution approchée locale permet de calculer un estimateur de "defaut" ou l'erreur de la discrétisation sur la grille grossière.
5. L'erreur, ajouté au second membre du problème sur la grille grossière, permet de déterminer une approximation globale de la solution continue qui est supposée être plus précise(tant en espace qu'en temps) que celle calculée à l'itération d'avant.
6. Cette solution approchée grossière peut être de nouveau utilisée pour "re-actualiser" la condition aux limites locales et l'ensemble de la procédure peut être répété jusqu'à convergence.

Le schéma d'itération de la méthode **LDC** est représenté sur la figure suivante:



Pour chaque pas de temps, la méthode **LDC** est donc une méthode itérative dans laquelle une approximation globale et locale de la solution continue améliore l'une l'autre progressivement .

Avantage de cette méthode:

L'un des principaux avantages de **LDC** est que les grilles grossières et fines peuvent être des grilles uniformément structurées.

Autre méthode déjà utilisée:

La méthode **LDC** est similaire à la méthode **LUGR**, *Local Uniform Grid Refinement*, présentée et analysée par Trompet et Verwer [8].

Cependant, **LDC** se diffère de **LUGR**. Dans **LUGR** la solution fine n'améliore pas globalement la solution approchée sur la grille grossière par la correction de l'erreur "defect correction". Pour cette raison, **LDC**, comme elle a été étudiée par [7], se révèle être plus robuste que **LUGR**. Toutefois, à l'exception de rendre **LDC** plus robuste, la correction nécessite plus de calcul.

LDC a été utilisé aussi pour résoudre des problèmes elliptiques 1D et 2D en adaptant une méthode de raffinement local multigrille sur l'intervalle espace.

Le but de ce stage est d'adapter cette méthode, qui est étudié pour les problèmes elliptiques, pour résoudre le problème de convection-diffusion instationnaire et faire des raffinement local multigrille pour l'intervalle temps.

Pour se faire, nous utilisons la méthode du schéma en temps "pararéal". Cette dernière a été introduite par **LIONS**, **MADAY** et **TURINICI** [9].

Cette méthode, basée sur un schéma d'Euler, combine des résolutions grossières et des résolutions fines et indépendantes en temps en s'inspirant de ce qui est classique en espace. Elle a pour principale motivation les problèmes en temps réel, d'où la terminologie proposée de "pararéal".

L'idée du schéma en temps "pararéal" est d'utiliser la solution approchée calculée sur la grille grossière pour corriger l'initialisation de chaque sous domaine et aussi les points courants . Ainsi, La méthode **LDC** peut être vu comme une extension du pararéal.

Dans le premier chapitre, on étudie la méthode **LDC** dans le cas 1D avec les différents

types de schémas: Euler explicite, Runge Kutta 2, 3 et 4.

Dans le deuxième chapitre, on passe au cas 2D.

Présentation de l'ONERA

Pour compléter mes études d'ingénieur en formation MACS, Mathématiques appliquées et calcul scientifique, j'ai effectué un stage de 5 mois au service Calcul Haute Performance de l'entreprise *ONERA* à Châtillon.

L'*ONERA*, *Office National d'Etudes et de Recherche Aéronautique*, créée en 1946 est un établissement public scientifique et technique à caractère industriel et commercial (EPIC), placé sous la tutelle du ministère de la défense.

ONERA a été créée autour de six missions clés:

- Orienter et conduire les recherches dans le domaine aéronautique.
- Valoriser ces recherches pour l'industrie nationale et européenne.
- Réaliser et mettre en oeuvre les moyens d'expérimentation associés.
- Fournir à l'industrie des prestations et des expertises de haut niveau.
- Conduire des actions d'expertise au bénéfice de l'Etat.
- Former des chercheurs et des ingénieurs.

Chiffres clés en 2008

- **2048** des effectifs totaux hors stagiaires dont 1198 ingénieurs et cadres.
- **215** des doctorants et post-doctorants.
- **59** des thèses soutenues.
- **202** millions d'euros de budget dont:
 - 11 millions d'euros d'activités dans le cadre de projets européens.
 - 18,5 % réalisés à l'international (coopération + export).
 - 61 % de charges de personnel.

L'*ONERA* se distingue dans son environnement par son volume d'activités et par cinq particularités:

- La dualité calcul-expérience.
- La connaissance des applications industrielles.
- Un parc de moyens d'essais unique en Europe .
- La dualité civil-militaire:
 - 1/3 pour le civil
 - 1/3 pour la défense

- 1/3 en recherches duales
- La pluridisciplinarité des équipes.

Présentation de *DTIM* et *CHP*

J'ai effectué ce stage dans le département *DTIM*, Département *T*raitement de l'*I*nformation et *M*odélisation, l'un des 17 départements scientifiques de l'*ONERA*. Il fait partie de la branche *TIS*, *T*raitement de l'*I*nformation et *S*ystèmes.

Le *DTIM* a pour mission de conduire des études et des recherches débouchant sur l'élaboration de modèles et méthodes aptes à affronter la complexité de systèmes, de techniques et d'outils informatiques. Ces activités sont déclinées dans les domaines :

- La modélisation, la conception, la validation, la mise en oeuvre et l'exploitation optimale de systèmes complexes de traitement de l'information numérique et\ou symbolique.
- La modélisation numérique pour la simulation et le calcul à haute performance (parallèle, distribué).
- Le traitement d'images.

Le *DTIM* est organisé en 6 Unités de Recherches dont *CHP*, *C*alcul à *H*aute *P*erformance, où s'est déroulé ce stage.

Le *CHP* offre une expertise dans la modélisation, le développement de méthodes et leur mise en oeuvre. La simulation numérique de phénomènes physiques est un enjeu dans toutes les phases de conception et d'évaluation de systèmes aéronautiques ou spatiaux. Le *CHP* s'intéresse aux nouvelles techniques de calcul, tant sur le plan des matériels que des logiciels, et à l'utilisation ou à la réutilisation de nouveaux outils(maillages adaptatifs, couplage de codes). Cette activité conduit à la réalisation de logiciels robustes, portables et pérennes, ce qui est une nécessité face à l'évolution rapide des matériels.

1.1 Position du problème

Dans ce chapitre, nous considérons un problème de convection-diffusion évolutif.

Soit a et ν deux membres réels positifs et soit:

$$u_0 : x \in [0, E] \longrightarrow u_0(x) \in \mathbb{R}$$

$$f : (x, t) \in [0, E] * [0, T] \longrightarrow f(x, t) \in \mathbb{R}$$

deux fonctions données.

Dans ce chapitre, nous cherchons une fonction

$$u : (x, t) \in [0, E] * [0, T] \longrightarrow u(x, t) \in \mathbb{R}$$

telle que

$$(1) \left\{ \begin{array}{lcl} \frac{\partial u}{\partial t}(x, t) + a \frac{\partial u}{\partial x}(x, t) - \nu \frac{\partial^2 u}{\partial x^2}(x, t) & = & f(x, t) \quad \forall x \in]0, E[, \forall t \in [0, T] \quad (1.1) \\ u(x, 0) & = & u_0(x) \quad \forall x \in]0, E[\quad (1.2) \\ u(x, t) & = & g_D(t) \quad \forall t \in [0, T] , x = 0, x = E \quad (1.3) \end{array} \right.$$

L'équation (1,1) est une équation de convection-diffusion évolutive. Le deuxième terme de cette équation modélise le phénomène de convection et le troisième terme, le phénomène de diffusion. L'équation (1,2) est la condition initiale.

L'équation (1,3) est les conditions aux limites, avec

$$g_D : t \in \mathbb{R}^+ \longrightarrow g_D(t) \in \mathbb{R}$$

Une fonction définie sur le bord Dirichlet (en $x = 0$ et $x = E$).

1.2 Existence et unicité d'une solution du problème

On admet que le problème précédent a une unique solution régulière $u(x,t)$ si

$$u_0 \in \mathbb{C}^2([0, E])$$

1.3 Formulation variationnelle

Soit $\phi \in \mathbb{C}_0^1([0, E] * [0, +\infty[)$ une fonction test.

En multipliant l'équation (1.1) par ϕ , en intégrant sur $[0, E] * [0, +\infty[$ et si ϕ est solution classique, alors

$$\int \int \left(\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} \right) \phi \, dx \, dt = \underbrace{\int \int f \phi \, dx \, dt}_{\mathcal{L}(\phi)}$$

En utilisant la formule de Green suivante, sur un domaine Ω convexe, pour ψ une fonction test assez régulière et η la normale à Ω ,

$$\int_{\Omega} \operatorname{div} \psi = \int_{\partial \Omega} \psi \eta \, d\gamma$$

Or pour $\psi = uv$,

$$\begin{aligned} \int_{\Omega} \left(u \frac{\partial v}{\partial x_j} + \frac{\partial u}{\partial x_j} v \right) &= \int_{\partial \Omega} uv \eta_j \, d\gamma \\ \Rightarrow \int_{\Omega} u \frac{\partial v}{\partial x_j} &= - \int_{\Omega} \frac{\partial u}{\partial x_j} v + \int_{\partial \Omega} uv \eta_j \, d\gamma \end{aligned}$$

Nous notons K le support de ϕ , i.e $K = \operatorname{supp} \phi$, qui est un support compact alors nous obtenons,

$$\begin{aligned} - \int_K \frac{\partial \phi}{\partial t} u - a \int_K \frac{\partial \phi}{\partial x} u + \nu \int_K \frac{\partial \phi}{\partial x} \frac{\partial u}{\partial x} + \underbrace{\int_{\partial K} u \phi \eta_t}_{= - \int_0^E u_0(x) \phi(x,0) \, dx} + a \underbrace{\int_{\partial K} u \phi \eta_x}_0 - \nu \underbrace{\int_{\partial K} \frac{\partial u}{\partial x} \phi \eta_x}_0 &= \mathcal{L}(\phi) \end{aligned}$$

Or, par la formule de Green,

$$\int_K \frac{\partial \phi}{\partial x} \frac{\partial u}{\partial x} = - \int_K \frac{\partial^2 \phi}{\partial x^2} u + \underbrace{\int_{\partial K} \frac{\partial \phi}{\partial x} u \eta_x}_0$$

Ainsi la relation précédente devient,

$$\begin{aligned}
& - \int_K \frac{\partial \phi}{\partial t} u - a \int_K \frac{\partial \phi}{\partial x} u - \nu \int_K \frac{\partial^2 \phi}{\partial x^2} u - \int_0^E u(0(x)) \phi(x, 0) dx = \mathcal{L}(\phi) \\
& \implies \int_K \left(\frac{\partial \phi}{\partial t} + a \frac{\partial \phi}{\partial x} + \nu \frac{\partial^2 \phi}{\partial x^2} \right) u + \int_0^E u(0(x)) \phi(x, 0) dx = -\mathcal{L}(\phi)
\end{aligned}$$

Cette relation est vraie, $\forall \phi$ si u est solution classique.

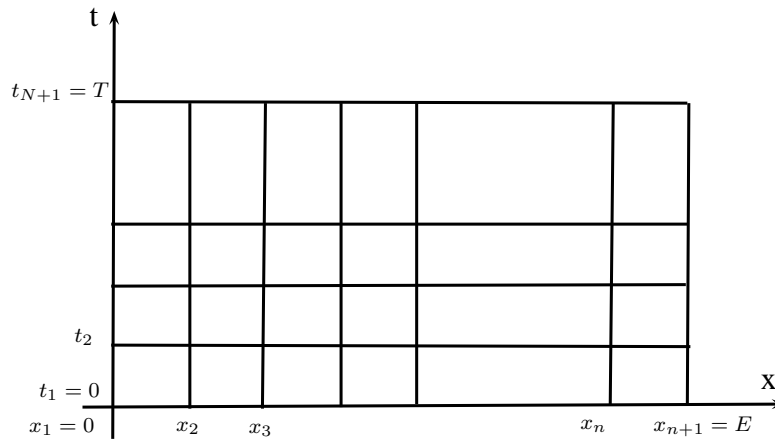
Nous disons que u est solution faible de l'équation si cette formule est vraie $\forall \phi \in \mathcal{C}_0^1([0, E] * [0, +\infty[)$ et $u \in \mathcal{L}_{localement}^\infty([0, E] * [0, +\infty[)$.

1.4 Résolution du problème

Soit N un entier positif, nous introduisons le pas temporel $\Delta t = \frac{T}{N}$ et les noeuds $t_j = (j-1)\Delta t$, avec $j = 1, \dots, N+1$.

Nous choisissons également, un pas spatial $\Delta x = \frac{E}{n}$, n est un entier positif nous introduisons les noeuds $x_i = (i-1)\Delta x$, avec $i = 1, \dots, n+1$.

Les points t_j et x_i sont représentés dans la grille ci-dessous:



1.4.1 Discrétisation du problème en espace

Afin de résoudre l'équation (1.1), nous allons d'abord réduire l'EDP à un système d'équations ou un système des ODE. Il y a trois méthodes qui sont, en général, utilisées pour discrétiser le domaine spatial:

- Différences finies
- Volumes finis
- Éléments finis

Dans la suite, nous utilisons la méthode de différences finies d'ordre 2, DF2, pour discrétiser le terme de diffusion et DF1 pour discrétiser le terme de convection.

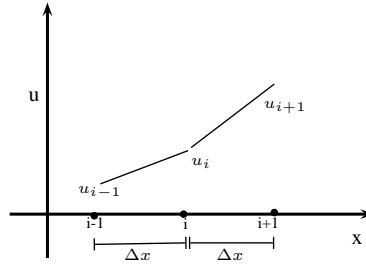
Ainsi, l'équation (1.1) est discrétisée en espace de la manière suivante, pour $i = 2 \dots n$

$$\frac{du_i}{dt} = -a \frac{\partial u}{\partial x}|_i + \nu \frac{\partial^2 u}{\partial x^2}|_i + f(t)|_i$$

Comme a est positif alors le terme de convection est approchée par une formule de différences finies rétrograde, c'est le cas de schéma décentré

$$a \frac{\partial u}{\partial x} \approx a \frac{u_i - u_{i-1}}{\Delta x}$$

Pour le terme de diffusion nous utilisons une interpolation lineaire de la manière suivante



Ainsi, nous pouvons estimer le second dérivé comme suit

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &\approx \frac{\frac{\partial u}{\partial x}|_{i+\frac{1}{2}} - \frac{\partial u}{\partial x}|_{i-\frac{1}{2}}}{\Delta x} \\ &\approx \frac{\frac{u_{i+1} - u_i}{\Delta x} - \frac{u_i - u_{i-1}}{\Delta x}}{\Delta x} \\ &\approx \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \end{aligned} \quad (1.4)$$

Et donc la discrétisation de (1.1) en espace est la suivante, pour $i = 2 \dots n$

$$\frac{du_i}{dt} = -a \frac{u_i - u_{i-1}}{\Delta x} + \nu \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + f(t)|_i$$

La discrétisation de (1.2) en espace est la suivante,

$$u_i(t=0) = u_{0i}$$

pour tout $i = 1 \dots n + 1$

1.4.2 Euler Explicite

Afin de résoudre l'équation (1.1), nous utilisons, en premier temps, un schéma d'Euler Explicite.

Soit $u(x_i, t_j)$ une valeur de la solution du problème au noeuds x_i et au temps t_j . Dans la suite, u_i^j est l'approximation d'une solution du problème (i.e $u_i^j \simeq u(x_i, t_j)$) La condition initiale (1.2) se traduit par :

$$u_i^1 = u_0^1, \text{ pour } i = 1, \dots, n+1$$

Les conditions aux limites (1.3) sont approchées par:

$$u_1^j = g_{D1}(t_j), \text{ pour } j = 1, \dots, N+1$$

$$u_{n+1}^j = g_{D2}(t_j), \text{ pour } j = 1, \dots, N+1$$

Avec g_{D1} et g_{D2} sont les conditions de Dirichlet en $x = 0$ et $x = E$, respectivement.

La discrétisation de $\frac{du_i}{dt}$ par Euler Explicite est,

$$\frac{du_i}{dt} \simeq \frac{u_i^{j+1} - u_i^j}{\Delta t}$$

Les valeurs u_i^j , $i = 1, \dots, n+1$ étant connues, les valeurs u_i^{j+1} , $i = 2, \dots, n$ sont calculées grâce au schéma d'Euler explicite progressif décentré.

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \underbrace{-a \frac{u_i^j - u_{i-1}^j}{\Delta x} + \nu \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2}}_{L(u_i^j)} + f_i^j \quad (1.4.1)$$

pour $i = 2, \dots, n$ et pour tout $j = 1, \dots, N$

Avec $f_i^j \simeq f(x_i, t_j)$

En conclusion, le schéma d'Euler Explicite est le suivant,

$$\begin{aligned} k1 &= L(u_i^j) + f_i^j \\ u_i^{j+1} &= u_i^j + \Delta t \, k1 \end{aligned}$$

Etude de la stabilité du schéma

Considérons le cas où $f = 0$, alors l'équation (1.4) devient:

$$u_i^{j+1} = \left(\frac{a\Delta t}{\Delta x} + \frac{\nu\Delta t}{\Delta x^2} \right) u_{i-1}^j + \left(1 - \frac{a\Delta t}{\Delta x} - \frac{2\nu\Delta t}{\Delta x^2} \right) u_i^j + \frac{\nu\Delta t}{\Delta x^2} u_{i+1}^j \quad (1.4.2)$$

Ce schéma est stable si le pas de temps Δt et le pas d'espace Δx satisfont

$$1 - \frac{a\Delta t}{\Delta x} - \frac{2\nu\Delta t}{\Delta x^2} \geq 0$$

Preuve:

Nous utilisons l'analyse de Von Neumann pour montrer la relation précédente.

Tout d'abord, nous posons, $\alpha = \frac{a\Delta t}{\Delta x}$ et $\beta = \frac{\nu\Delta t}{\Delta x^2}$
alors, nous avons,

$$u_i^{j+1} = (\alpha + \beta) u_{i-1}^j + (1 - \alpha - 2\beta) u_i^j + \beta u_{i+1}^j \quad (1.4.3)$$

Soit l'onde pure,

$$u_i^j = a_j e^{ikx_i}$$

En la remplaçant dans l'équation précédente, nous obtenons,

$$a_{j+1} e^{ikx_i} = a_j ((\alpha + \beta) e^{ikx_{i-1}} + (1 - \alpha - 2\beta) e^{ikx_i} + \beta e^{ikx_{i+1}})$$

\Rightarrow

$$\begin{aligned} \frac{a_{j+1}}{a_j} &= (\alpha + \beta) e^{-ik\Delta x} + (1 - \alpha - 2\beta) + \beta e^{ik\Delta x} \\ &= (1 - \alpha - 2\beta) + \beta \underbrace{(e^{ik\Delta x} + e^{-ik\Delta x})}_{2 \cos(k\Delta x)} + \alpha e^{-ik\Delta x} \\ &= (1 - \alpha - 2\beta \underbrace{(1 - \cos(k\Delta x))}_{2 \sin^2(\frac{k\Delta x}{2})}) + \alpha e^{-ik\Delta x} \\ &= (1 - \alpha - 4\beta \sin^2(\frac{k\Delta x}{2}) + \alpha \cos(k\Delta x)) - i\alpha \sin(k\Delta x) \\ &= (1 - \alpha(1 - \cos(k\Delta x)) - 4\beta \sin^2(\frac{k\Delta x}{2})) - i\alpha \sin(k\Delta x) \\ &= (1 - 2\alpha \sin^2(\frac{k\Delta x}{2}) - 4\beta \sin^2(\frac{k\Delta x}{2})) - i\alpha \sin(k\Delta x) \\ &= \underbrace{(1 - 2(\alpha + 2\beta) \sin^2(\frac{k\Delta x}{2})) - i\alpha \sin(k\Delta x)}_g \end{aligned}$$

Par Von Neumann, dès que le facteur d'amplification $|g| \leq 1$ alors le schéma est stable sinon divergence.

Or,

$$\begin{aligned} |g|^2 &= (1 - 2(\alpha + 2\beta) \sin^2(\frac{k\Delta x}{2}))^2 + \alpha^2 \sin^2(k\Delta x) \\ &= 1 + 4(\alpha + 2\beta)^2 \sin^4(\frac{k\Delta x}{2}) - 4(\alpha + 2\beta) \sin^2(\frac{k\Delta x}{2}) + \alpha^2 \sin^2(k\Delta x) \end{aligned}$$

Soit $X = \sin^2(\frac{k\Delta x}{2})$ alors, $\sin^2(k\Delta x) = 4 \sin^2(\frac{k\Delta x}{2}) \cos^2(\frac{k\Delta x}{2}) = 4 X (1 - X)$
Et donc,

$$|g|^2 = 1 + 4(\alpha + 2\beta)^2 X^2 - 4(\alpha + 2\beta) X + 4\alpha^2 X (1 - X)$$

Comme $|g| \leq 1$ et $0 \leq X < 1$, alors,

$$\begin{aligned} 1 + 4(\alpha + 2\beta)^2 X^2 - 4(\alpha + 2\beta) X + 4\alpha^2 X(1 - X) &\leq 1 \\ (\alpha + 2\beta)^2 X - (\alpha + 2\beta) + \alpha^2(1 - X) &\leq 0 \\ ((\alpha + 2\beta)^2 - \alpha^2) X - (\alpha + 2\beta) + \alpha^2 &\leq 0 \\ 4\beta(\beta + \alpha) X - (\alpha + 2\beta) + \alpha^2 &\leq 0 \end{aligned}$$

Or, $\beta + \alpha \geq 0$, alors le maximum est atteint en $X = 1$,

$$\begin{aligned} 4\beta(\beta + \alpha) - (\alpha + 2\beta) + \alpha^2 &\leq 0 \\ (\alpha + 2\beta)^2 - (\alpha + 2\beta) &\leq 0 \\ \underbrace{(\alpha + 2\beta)(\alpha + 2\beta - 1)}_{\geq 0} &\leq 0 \\ \implies \alpha + 2\beta - 1 &\leq 0 \\ \implies 1 - \alpha - 2\beta &\geq 0 \end{aligned}$$

Erreur de troncature / consistance

La solution continue vérifie le schéma à une perturbation près.

Nous définissons, l'erreur de troncature ou de consistance, l'erreur suivant:

$$T_i^j = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} + a \frac{u(x_i, t_j) - u(x_{i-1}, t_j)}{\Delta x} - \nu \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{\Delta x^2} - f(x_i, t_j)$$

Maintenant, nous utilisons un développement limité pour approcher chacune des ces expressions,

$$\begin{aligned} u(x_i, t_{j+1}) &\simeq u(x_i, t_j) + \Delta t \frac{\partial u}{\partial t}(x_i, t_j) + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t_j) + \mathcal{O}(\Delta t^3) \\ u(x_{i+1}, t_j) &\simeq u(x_i, t_j) + \Delta x \frac{\partial u}{\partial x}(x_i, t_j) + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t_j) + \frac{\Delta x^3}{6} \frac{\partial^3 u}{\partial x^3}(x_i, t_j) + \frac{\Delta x^4}{24} \frac{\partial^4 u}{\partial x^4}(x_i, t_j) + \mathcal{O}(\Delta x^5) \\ u(x_{i-1}, t_j) &\simeq u(x_i, t_j) - \Delta x \frac{\partial u}{\partial x}(x_i, t_j) + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t_j) - \frac{\Delta x^3}{6} \frac{\partial^3 u}{\partial x^3}(x_i, t_j) + \frac{\Delta x^4}{24} \frac{\partial^4 u}{\partial x^4}(x_i, t_j) + \mathcal{O}(\Delta x^5) \\ \implies \end{aligned}$$

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} = \frac{\partial u}{\partial t}(x_i, t_j) + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t_j)$$

$$\frac{u(x_i, t_j) - u(x_{i-1}, t_j)}{\Delta x} = \frac{\partial u}{\partial x}(x_i, t_j) - \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t_j) + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3}(x_i, t_j)$$

$$\frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{\Delta x^2} = \frac{\partial^2 u}{\partial x^2}(x_i, t_j) + \frac{\Delta x^2}{12} \frac{\partial^4 u}{\partial x^4}(x_i, t_j)$$

En les remplaçant dans l'expression de T_i^j , nous obtenons,

$$T_i^j = \cancel{\frac{\partial u}{\partial t}} + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} + \cancel{a \frac{\partial u}{\partial x}} - a \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} - \cancel{\nu \frac{\partial^2 u}{\partial x^2}} - \nu \frac{\Delta x^2}{12} \frac{\partial^4 u}{\partial x^4} - \cancel{f(x_i, t_j)}$$

$$\Rightarrow T_i^j = \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} - a \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} - \nu \frac{\Delta x^2}{12} \frac{\partial^4 u}{\partial x^4}$$

Par définition, le schéma est consistant si l'erreur de troncature tend vers 0 quand Δx et Δt tendent vers 0.

On remarque que,

$$\sup_{i,j} |T_i^j| = \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x)$$

Donc, le schéma est au moins d'ordre un en temps et en espace.

Maintenant, nous allons majorer l'erreur de troncature,

$$|T_i^j| \leq \frac{\Delta t}{2} \sup_{x,t} \left| \frac{\partial^2 u}{\partial t^2} \right| + a \frac{\Delta x}{2} \sup_{x,t} \left| \frac{\partial^2 u}{\partial x^2} \right| + \nu \frac{\Delta x^2}{12} \sup_{x,t} \left| \frac{\partial^4 u}{\partial x^4} \right|$$

Nous posons, $M_{\lambda\lambda} = \sup \{|u_{\lambda\lambda}(x, t)|; 0 \leq x \leq E, 0 \leq t \leq T\}$

Alors,

$$\begin{aligned} |T_i^j| &\leq \frac{\Delta t}{2} M_{tt} + a \frac{\Delta x}{2} M_{xx} + \nu \frac{\Delta x^2}{12} M_{xxxx} \\ \Rightarrow |T_i^j| &\leq \frac{\Delta t}{2} M_{tt} + \frac{\Delta x^2}{2} \left(\frac{a}{\Delta x} M_{xx} + \frac{\nu}{6} M_{xxxx} \right) \end{aligned}$$

Or en utilisant la condition de stabilité, nous en deduisons que,

$$\begin{aligned} 0 &\leq \frac{a}{\Delta x} \leq \frac{1}{\Delta t} - 2 \frac{\nu}{\Delta x^2} \\ \Rightarrow |T_i^j| &\leq \frac{\Delta t}{2} M_{tt} + \frac{\Delta x^2}{2} \left(\left(\frac{1}{\Delta t} - 2 \frac{\nu}{\Delta x^2} \right) M_{xx} + \frac{\nu}{6} M_{xxxx} \right) \end{aligned}$$

Etude de la convergence du schéma

On pose $e_i^j = u(x_i, t_j) - u_i^j$, l'erreur au point i et au temps j, alors l'erreur vérifie le schéma de différences finis perturbé par l'erreur de troncature T_i^j

L'erreur vérifie une équation du type:

$$\begin{cases} \frac{\partial e}{\partial t} &= -a \frac{\partial e}{\partial x} + \nu \frac{\partial^2 e}{\partial x^2} + \varepsilon \\ e(x, 0) &= 0 \end{cases}$$

En discrétisant ce schéma, nous obtenons le système suivant, pour $i = 2, \dots, n$ et pour $j = 1, \dots, N$:

$$\begin{cases} \frac{e_i^{j+1} - e_i^j}{\Delta t} &= -a \frac{e_i^j - e_{i-1}^j}{\Delta x} + \nu \frac{e_{i+1}^j - 2e_i^j + e_{i-1}^j}{\Delta x^2} + T_i^j \\ e_i^1 &= 0 \end{cases}$$

$$\Rightarrow e_i^{j+1} = (\alpha + \beta) e_{i-1}^j + (1 - \alpha - 2\beta) e_i^j + \beta e_{i+1}^j + \Delta t T_i^j$$

Pour α et β définient précédemment.

$$\Rightarrow |e_i^{j+1}| \leq (\alpha + \beta) |e_{i-1}^j| + |1 - \alpha - 2\beta| |e_i^j| + \beta |e_{i+1}^j| + \Delta t |T_i^j|$$

Par la condition de stabilité, nous avons,

$$|1 - \alpha - 2\beta| = 1 - \alpha - 2\beta$$

Et donc,

$$\begin{aligned} |e_i^{j+1}| &\leq (\alpha + \beta) \max_i |e_{i-1}^j| + (1 - \alpha - 2\beta) \max_i |e_i^j| + \beta \max_i |e_{i+1}^j| + \Delta t \max_i |T_i^j| \\ &\leq (\alpha + \beta + 1 - \alpha - 2\beta + \beta) \max_i |e_i^j| + \Delta t \max_i |T_i^j| \\ \Rightarrow \end{aligned}$$

$$\begin{aligned} \max_i |e_i^{j+1}| &\leq \cancel{\max_i |e_i^j|} + \Delta t \max_i |T_i^j| \\ \cancel{\max_i |e_i^j|} &\leq \cancel{\max_i |e_i^{j-1}|} + \Delta t \max_i |T_i^{j-1}| \\ &\vdots \\ &\vdots \\ &\vdots \\ \cancel{\max_i |e_i^1|} &\leq \max_i \underbrace{|e_i^0|}_0 + \Delta t \max_i |T_i^0| \\ \Rightarrow \max_i |e_i^j| &\leq \Delta t \sum_{k=0}^{j-1} \max_i |T_i^k| \\ \Rightarrow |u(x_i, t_j) - u_i^j| &\leq \Delta t \sum_{k=0}^{j-1} \max_i |T_i^k| \end{aligned}$$

Maintenant, nous utilisons la majoration de l'erreur de troncature, nous obtenons la majoration suivante:

$$\begin{aligned} |u(x_i, t_j) - u_i^j| &\leq \Delta t \sum_{k=0}^{j-1} \left(\frac{\Delta t}{2} M_{tt} + \frac{\Delta x^2}{2} \left(\left(\frac{1}{\Delta t} - 2 \frac{\nu}{\Delta x^2} \right) M_{xx} + \frac{\nu}{6} M_{xxx} \right) \right) \\ \Rightarrow |u(x_i, t_j) - u_i^j| &\leq \underbrace{\Delta t j}_{t_{j+1} \leq T} \left(\frac{\Delta t}{2} M_{tt} + \frac{\Delta x^2}{2} \left(\left(\frac{1}{\Delta t} - 2 \frac{\nu}{\Delta x^2} \right) M_{xx} + \frac{\nu}{6} M_{xxx} \right) \right) \\ \Rightarrow |u(x_i, t_j) - u_i^j| &\leq \frac{T}{2} \left(\Delta t M_{tt} + \Delta x^2 \left(\left(\frac{1}{\Delta t} - 2 \frac{\nu}{\Delta x^2} \right) M_{xx} + \frac{\nu}{6} M_{xxx} \right) \right) \end{aligned}$$

Nous en déduisons, que ce schéma converge (pour $\Delta t \rightarrow 0$ et $\Delta x \rightarrow 0$) si la condition de stabilité est vérifiée.

1.4.3 Runge Kutta d'ordre 2,3 et 4

Maintenant, nous utilisons un schéma de Runge Kutta pour discrétiser l'équation (1.1) en temps.

Pour le cas de Runge Kutta, les conditions aux limites (l'équation (1.3)) ainsi que la condition initiale (l'équation (1.2)) ne changent pas, nous utilisons les mêmes que celles introduites dans le paragraphe précédente, seule une différence par rapport au schéma d'Euler Explicite est la discrétisation de l'équation (1,1) qui devient plus précise.

Pour $i = 2.....n$ et pour tout $j = 1.....N$

Runge Kutta 2

$$\begin{aligned}k1 &= \Delta t (L(u_i^j) + f_i^j) \\k2 &= \Delta t (L(u_i^j + k1) + f_i(t_j + \Delta t)) \\u_i^{j+1} &= u_i^j + \Delta t \frac{k1 + k2}{2}\end{aligned}$$

Runge Kutta 3

$$\begin{aligned}k1 &= \Delta t (L(u_i^j) + f_i(t_j)) \\k2 &= L(u_i^j + \frac{\Delta t k1}{3}) + f_i(t_j + \frac{\Delta t}{3}) \\k3 &= L(u_i^j + \frac{2\Delta t k2}{3}) + f_i(t_j + \frac{2\Delta t}{3}) \\u_i^{j+1} &= u_i^j + \Delta t \frac{(k1 + 3 k3)}{4}\end{aligned}$$

Runge Kutta 4

$$\begin{aligned}k1 &= \Delta t (L(u_i^j) + f_i(t_j)) \\k2 &= \Delta t (L(u_i^j + \frac{k1}{2}) + f_i(t_j + \frac{\Delta t}{2})) \\k3 &= \Delta t (L(u_i^j + \frac{k2}{2}) + f_i(t_j + \frac{\Delta t}{2})) \\k4 &= \Delta t (L(u_i^j + k3) + f_i(t_j + \Delta t)) \\u_i^{j+1} &= u_i^j + \frac{k1 + 2 k2 + 2 k3 + k4}{6}\end{aligned}$$

1.5 Méthode \mathcal{LDC}

Dans cette section, nous étudions la méthode \mathcal{LDC} qui a été introduite par Hackbusch [1] pour résoudre les valeurs sur la frontière d'un problème elliptique.

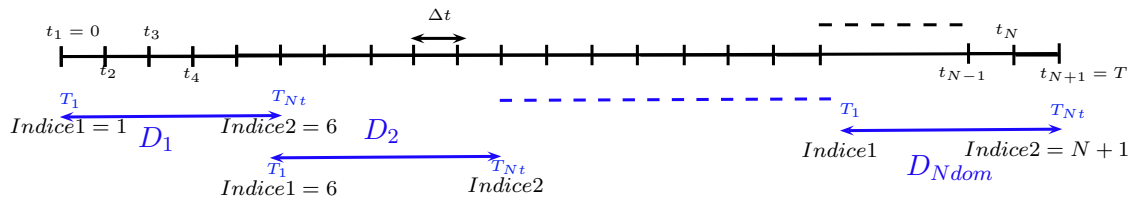
\mathcal{LDC} est une méthode de décomposition de domaine dans laquelle, le domaine spatial fin s'imbrique dans le domaine spatial grossier. Dans [7], Cette méthode est étendue pour inclure l'adaptativité, raffinement multilevel et décomposition du domaine.

Dans cette section, nous présentons la méthode de \mathcal{LDC} pour résoudre le problème de convection diffusion en temps.

Le problème (1) doit être discrétiser en temps et en espace afin d'être résolu numériquement. Pour cette raison, nous introduisons une grille grossière (Le pas de la grille est Δt). Nous introduisons aussi le pas spatial Δx . Nous décomposons le domaine temporel en N_{dom} sous domaines. Sur chaque sous domaine, une approximation de la solution continue sur la grille grossière calculée avec le pas de temps Δt peut-être pas assez suffisante pour représenter $u(x, t)$. afin de s'approcher de la solution continue, nous introduisons une grille uniforme fine (de pas $\delta t < \Delta t$) sur chaque sous domaine, qu'on la note D_{idom} pour $idom = 1...N_{dom}$. Sur chaque sous domaine, le pas d'espace est performé en utilisant un pas spatial $\delta x = \frac{\Delta x}{\tau}$, avec τ un entier ≥ 1 .

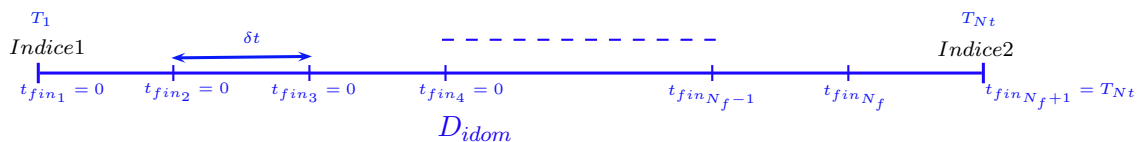
Dans la méthode \mathcal{LDC} , la solution locale fine a été utilisée pour améliorer une approximation de la solution grossière via l'erreur de la discrétisation "Defect correction".

Nous vous présentons le schéma suivant pour mieux comprendre la décomposition du domaine temporel:



Avec N_t est le nombre des points sur chaque sous domaine. $T_1...T_{N_t}$ sont les noeuds et Indice 1 et Indice 2 (ce sont les $j = 1...N + 1$ de la grille grossière) sont des entiers positifs qui représentent l'intervalle de chaque sous domaine.

Sur la figure suivante, nous présentons, pour un sous domaine quelconque par exemple D_{idom} , une décomposition de la grille fine:



N_f étant le nombre des points sur la grille fine. $t_{fin_1} \dots t_{fin_{N_f+1}}$ sont les noeuds de la grille fine.

1.5.1 Problème (1) sur la grille grossière

Afin de résoudre le problème (1), nous utilisons Différences finis d'ordre 2 pour discrétiser l'espace. Une approximation de la solution continue va être calculer, en premier temps, par Euler Explicite, comme a été expliqué précédemment, puis nous utilisons Runge Kutta 2, 3 et 4 afin d'obtenir plus de précision sur le schéma.

En utilisant l'une de schémas précédents, nous obtenons une approximation de la solution continue sur la grille grossière que nous notons u_g .

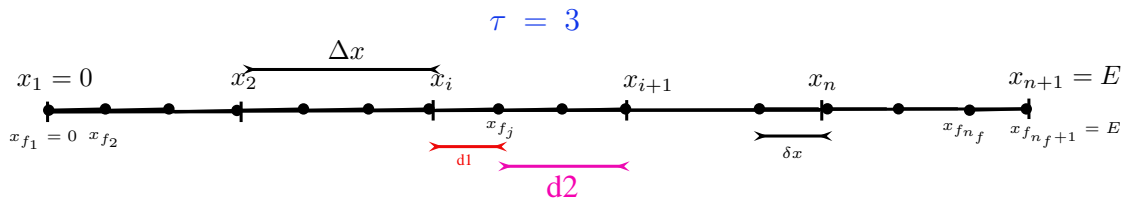
1.5.2 L'itération de \mathcal{LDC}

Problème (1) sur la grille fine

Afin de formuler un problème discret sur la grille fine, nous avons besoin de définir une condition artificielle en Indice 1 de chaque sous domaine. Nous pouvons prescrire des conditions de type Dirichlet, pour simplifier le problème au début, puis nous utilisons des conditions de type Robin.

Si nous voulons réaliser la discrétisation de l'espace avec un pas spatial $\delta x = \frac{\Delta x}{\tau}$ sur chaque domaine, nous avons, également, besoin de fournir des conditions aux limites, à tous les niveaux d'espace intermédiaire. A cet effet, nous utilisons un interpolateur linéaire.

Nous discrétisons le maillage espace comme suit, pour $\tau = 3$



Sur la grille fine, nous allons utiliser qu'Euler Explicite en temps et différences finis 2 en espace pour résoudre le problème (1).

En notant, u_f , une approximation de la solution continue, nous résolvons ainsi l'équation suivante,

$$\frac{u_{f_i}^{j+1} - u_{f_i}^j}{\delta t} = -a \frac{u_{f_i}^j - u_{f_{i-1}}^j}{\delta x} + \nu \frac{u_{f_{i+1}}^j - 2u_{f_i}^j + u_{f_{i-1}}^j}{\delta x^2} + f_i^j \quad (1.5.1)$$

Pour $j = 1 \dots N_f$ et pour $i = 2 \dots n_f$ avec $n_f = \tau n$.

1^{er} Cas: Condition de type Dirichlet en Indice 1

La procédure (1.5.1) est initialisée en utilisant l'approximation que nous avons obtenu sur la grille grossière, i.e u_g , en Indice 1 de la façon suivante, en utilisant l'interpolateur linéaire définie précédemment,

$$u_{f0} = \frac{d2 u_g(i, Indice1) + d1 u_g(i + 1, Indice1)}{\delta x \tau} \quad (1.5.2)$$

Pour tout $i = 1 \dots n$.

2^{me} Cas: Condition de type Robin en Indice 1

Dans ce cas, nous utilisons la formule de la condition Robin suivante:

$$u_f + \mu \frac{\partial u_f}{\partial t} = u_g + \mu \frac{\partial u_g}{\partial t}$$

Ce problème est bien posé si μ est un entier strictement positif.

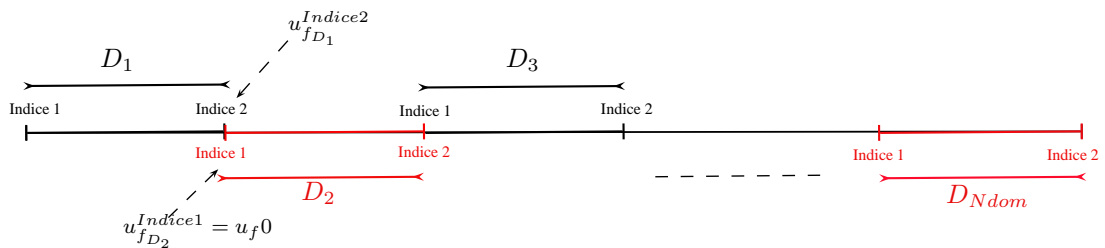
Pour le premier domaine, nous allons calculer la condition initiale comme dans l'équation (1.5.2). Pour les autres domaines, nous allons utiliser une discrétisation par Euler Explicite en temps, nous obtenons donc,

$$u_{f_i}^j + \mu \frac{u_{f_i}^j - u_{f_i}^{j-1}}{\delta t} = u_{g_k}^l + \mu \frac{u_{g_k}^l - u_{g_k}^{l-1}}{\Delta t}$$

Pour $i = 1 \dots n_f + 1$, $j = 1 \dots N_f + 1$, $k = 1 \dots n + 1$ et $l = Indice1 \dots Indice2$.

En particulier, en $j = 1$ et $l = Indice1$, nous ne connaissons pas la valeur de u_{f0} . Pour cette raison, nous allons utiliser l'approximation du domaine précédent par exemple sur D_2 nous allons utiliser la valeur de u_f sur D_1 en Indice 2. Nous obtenons donc l'équation suivante sur D_{idom} pour $idom = 2 \dots Ndom$:

$$u_{f_i}^1 + \mu \frac{u_{f_i}^1 - u_{f_i}^0}{\delta t} = u_{g_k}^{Indice1} + \mu \frac{u_{g_k}^{Indice1} - u_{g_k}^{Indice1-1}}{\Delta t}$$



$$\Rightarrow u_{f_{D_{idom}^i}}^1 + \mu \frac{u_{f_{D_{idom}^i}}^1 - u_{f_{D_{idom-1}^i}}^{Indice2-1}}{\delta t} = u_{g_k}^{Indice1} + \mu \frac{u_{g_k}^{Indice1} - u_{g_k}^{Indice2-1}}{\Delta t}$$

Pour $idom = 2...Ndom$.

L'intérêt d'utiliser Robin comme condition initiale, est d'accélérer la convergence de la méthode et surtout de diminuer le nombre d'itération de la procédure \mathcal{LDC} . Ce qui prouvera les résultats numériques dans la suite.

Les conditions aux limites (1.3) sont approchées sur la grille fine par:

$$u_{f_1}^j = g_{D1}(t_{fin_j}) , \text{ pour } j = 1, \dots, N_f + 1$$

$$u_{f_{n_f+1}}^j = g_{D2}(t_{fin_j}) , \text{ pour } j = 1, \dots, N_f + 1$$

Avant de passer à l'étape suivante qui est la projection de u_f sur la grille grossière, nous commençons tout d'abord par projeter cette approximation en espace, i.e $y_f \leftarrow u_f(1 : \tau : n_f + 1, :)$

Projection de y_f sur la grille grossière

L'approximation de la grille fine est, maintenant, utilisée pour améliorer, globalement, la solution sur la grille grossière pour chaque sous domaine.

La solution de la grille fine est considérée être plus précise que l'approximation de la grille grossière u_g car u_f est calculée avec un pas spatial $\delta x < \Delta x$ et avec un pas de temps $\delta t < \Delta t$.

La solution calculée sur la grille fine peut donc être utiliser pour approcher l'erreur de discrétisation locale ou "defect correction" sur D_{idom} pour $idom = 1...Ndom$.

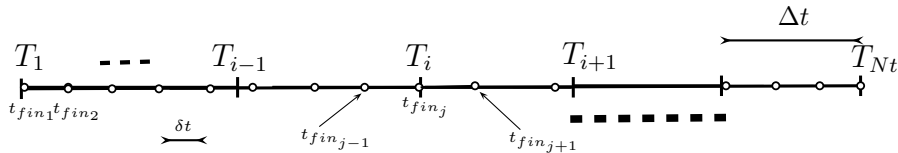
Nous commençons, d'abord, par vous présenter une restriction de l'opérateur de projection, R , de la grille fine sur la grille grossière de telle sorte que,

$$(R u)(x, t) = y_f(x, t)$$

Où x et t sont les noeuds grossiers spatiales et temporels, respectivement, sur chaque D_{idom} pour $idom = 1...Ndom$.

Il y a deux cas différents pour calculer cet interpolateur de projection linéaire, R . Ce calcul dépend, en effet, de l'endroit où se trouve les noeuds fins t_{fin_j} pour $j = 1...N_f+1$ entre les noeuds grossiers T_i et T_{i+1} pour $i = 1...Nt$

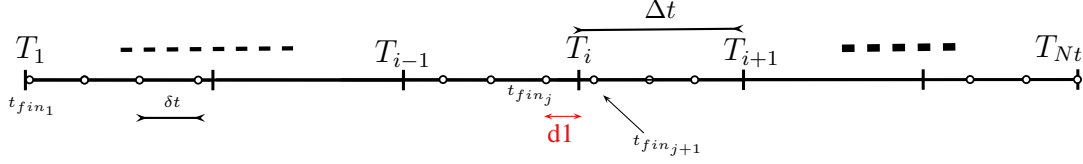
1^{er} Cas: $t_{fin_j} = T_i$



Dans ce cas, le noeud fin t_{fin_j} pour $j = 1....N_f + 1$ coïncide avec le noeud grossier T_i pour $i = 1....Nt$ et alors l'interpolateur de projection est:

$$(\mathbf{Ru})_{\mathbf{k}}^i = \mathbf{y}_{\mathbf{f}_k}^j, \text{ pour } k = 1....n+1, i = 1....Nt \text{ et } j = 1....N_f + 1$$

2^{me} Cas: $t_{fin_j} < T_i < t_{fin_{j+1}}$



Si nous notons $d2 = \delta t - d1$ alors R est calculé comme suit, pour $k = 1....n+1$, $i = 1....Nt$ et $j = 1....N_f + 1$

$$(\mathbf{Ru})_{\mathbf{k}}^i = \frac{d2 \mathbf{y}_{\mathbf{f}_k}^j + d1 \mathbf{y}_{\mathbf{f}_k}^{j+1}}{\delta t}$$

Estimation de l'erreur "Defect Correction"

La partie la plus cruciale de l'algorithme \mathcal{LDC} est comment la solution fine est utilisée pour améliorer l'approximation de la solution sur la grille grossière à travers une approximation de la discrétisation de l'erreur ou le "Defect".

Le defect "d" est défini, pour $i = 1....n+1$ et pour $j = 1....Nt - 1$, par:

$$d_i^j = \frac{u_i^{j+1} - u_i^j}{\Delta t} - L(u_i^j) - f_i(T_j) \quad (1.5.3)$$

Avec L est l'opérateur introduit précédemment.

Dans (1.5.2), nous avons utilisé la projection, sur chaque domaine, de la solution exacte continue u discrétisée dans le schéma (1.4.1).

Si nous savons les valeurs de "d" pour chaque sous domaine, nous pouvons les utiliser pour trouver une meilleure approximation de u sur la grille grossière. Cet objectif peut être atteint en rajoutant "d" dans le second membre du schéma (1.4.1). Toutefois, puisque nous ne connaissons pas la solution exacte de l'équation de convection diffusion, nous ne pouvons pas calculer les valeurs de "d" sur chaque D_{idom} pour $idom = 1...Ndom$. Ce que nous pouvons faire, cependant, est d'utiliser l'approximation fine, qui est plus précise, pour obtenir une estimation de "Local Defect" sur chaque sous domaine, i.e, pour $i = 1....n+1$

$$d_i^j := \begin{cases} \frac{Ru_i^{j+1} - Ru_i^j}{\Delta t} - L(Ru_i^j) - f_i(T_j) & \text{pour } j = 1....Nt - 1 \\ 0 & \text{pour } j = Nt \end{cases}$$

Des résultats précédents, sur la méthode \mathcal{LDC} , pour les problèmes stationnaires (voir [1] et [9]) montrent qu'il peut être convenant d'estimer le " Defect" pas en tout point de sous domaine mais dans un sous ensemble de D_{idom} , $idom = 1...N_{dom}$. En particulier, les noeuds situés à proximité de l'interface entre deux sous domaines doivent être exclu. Ce qui conduit à l'introduction d'un sous ensemble de D_{idom} , séparé de l'interface par une région appelée "région de sécurité". Plutôt que d'améliorer la précision de la méthode, le principal avantage de l'utilisation d'une région de sécurité est d'accélérer la convergence. Sur la figure précédente, la région de sécurité est celle en rouge. Ce qui explique pour quoi l'approximation "d", en $j = Nt$, est nulle.

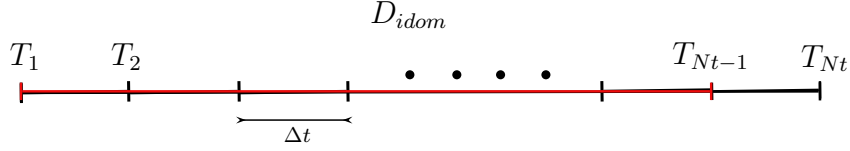


Figure 1.1: Exemple d'un sous domaine avec une **région de sécurité**

Le "Defect Correction" est maintenant calculé, sur chaque sous domaine. Nous rassemblons chacun de ces erreurs pour les ramener sur la grille grossière globale, que nous le notons P.

Finalement, l'estimateur P est rajouté au second membre de l'équation de convection-diffusion:

$$\frac{\partial u_g}{\partial t} = -a \frac{\partial u_g}{\partial x} + \nu \frac{\partial^2 u_g}{\partial x^2} + f + P \quad (1.5.4)$$

En discrétisant cette équation par Euler Explicite, nous obtenons,

$$\begin{cases} k1 &= L(u_{g_i}^j) + f_i^j + P_i^j \\ u_{g_i}^{j+1} &= u_{g_i}^j + \Delta t \, k1 \end{cases}$$

pour $i = 1.....n + 1$ et pour $j = 1.....N$.

Et par Runge Kutta 4, nous avons

$$\begin{cases} k1 &= \Delta t(L(u_{g_i}^j) + f_i(t_j) + P_i^j) \\ k2 &= \Delta t(L(u_{g_i}^j + \frac{k1}{2}) + f_i(t_j + \frac{\Delta t}{2}) + P_i^j) \\ k3 &= \Delta t(L(u_{g_i}^j + \frac{k2}{2}) + f_i(t_j + \frac{\Delta t}{2}) + P_i^j) \\ k4 &= \Delta t(L(u_{g_i}^j + k3) + f_i(t_j + \Delta t) + P_i^j) \\ u_{g_i}^{j+1} &= u_{g_i}^j + \frac{k1 + 2 \, k2 + 2 \, k3 + k4}{6} \end{cases}$$

La nouvelle solution grossière, à l'itération 1, peut être utilisée pour actualiser la condition initiale pour le nouveau problème fin en Indice 1 de chaque sous domaine, qui à son tour sera utilisée pour corriger l'approximation de la grille grossière. Pour chaque sous domaine, la méthode \mathcal{LDC} est donc une procédure itérative, elle est formalisé dans l'algorithme 1, et, comme elle a été étudiée pour les problèmes stationnaires (voir [10]), la convergence est très rapide.

Algorithme 1 Algorithme de \mathcal{LDC} pour l'équation de convection-diffusion

- **Initialisation:**

- ★ $a, \nu, Tol, ItMax,$
- ★ $T, N, \Delta t, t_1, \dots, t_{N+1}, Ndom,$
- ★ $E, n, \Delta x, x_1, \dots, x_{n+1}.$

- $u_{g0} \leftarrow u_{exacte}(x, t = 0)$

- Calculer une approximation u_g sur la grille grossière en résolvant le problème (1.4.1) soit par Euler Explicite, soit par Runge Kutta.

- $u_{gInitiale} \leftarrow u_g.$

- Critère-Arrêt $\leftarrow \|u_g\|$, $it \leftarrow 0$

- Diviser le pas spatial Δx par τ , $\delta x = \frac{\Delta x}{\tau}$ et définir n_f et x_f .

Tantque Critère- Arrêt $< Tol$ & $it < ItMax$ **faire**

- $u_{g1} \leftarrow u_g.$

Pour $idom \leftarrow 1 \dots Ndom$ **faire**

- Définir $T_1 \dots T_{Nt}$ sur chaque D_{idom} .
- Initialiser N_f, t_{fin} et δt .
- Calculer la condition initiale pour le problème fin comme dans (1.5.2).
- Calculer une approximation u_f par Euler Explicite en résolvant (1.5.1).
- $y_f \leftarrow u_f(1 : \tau : n_f + 1, :)$
- Projection de y_f sur la grille grossière et calcul de Ru.
- Utiliser Ru pour calculer une approximation de " Defect Correction", la discrétisation locale de l'erreur pour chaque D_{idom} .
- Assemblage de "d" dans P.

Fin pour idom

- Calculer une approximation grossière u_g plus précise en résolvant le problème modifié comme dans (1.5.4) par Euler Explicite ou Runge Kutta.

- Critère-Arrêt $\leftarrow \frac{\|u_g(:, Indice2) - u_{g1}(:, Indice2)\|}{\|u_{gInitiale}(:, Indice2)\|}$

- $it \leftarrow it + 1$

Fin Tantque

Dans l'algorithme 1, l'itération *Tantque*, est appelée l'itération \mathcal{LDC} . Chaque itération \mathcal{LDC} consiste à recalculer une approximation de la solution u sur chaque sous domaine. Pour une meilleure performance en résolvant le problème transitoire, il est, donc, souhaitable que seul un petit nombre d'itération \mathcal{LDC} (plus intéressant surtout si $it < N_{dom}$) soient nécessaires sur chaque D_{idom} , pour $idom = 1....N_{dom}$. Toutefois, comme c'est le cas pour le problème stationnaire, la convergence de l'itération \mathcal{LDC} est très rapide et, en général, exige qu'une seule itération.

Lors de la présentation de la méthode \mathcal{LDC} , nous avons utilisé Euler explicite pour discrétiser le temps sur la grille fine. Nous devons signaler que ce n'est pas restrictive et que d'autres schémas implicites pour la discrétisation du temps pourraient être appliquées aussi bien. D'ailleurs, nous ne sommes pas contraints d'utiliser le même schéma pour la grille grossière et fine, c'est pour cette raison, que nous avons utilisé Runge Kutta 2,3 et 4 sur la grille grossière et Euler Explicite sur la grille fine. Par contre, il est crucial pour l'efficacité de cette méthode, qu'un schéma d'Euler Explicite soit appliqué sur la grille fine.

L'une des caractéristiques les plus intéressantes de l'algorithme \mathcal{LDC} que nous proposons, c'est la possibilité de réaliser un pas d'espace sur la grille fine avec un pas $\delta x < \Delta x$. Il s'agit d'une fonctionnalité, pas une exigence de la méthode, il est bien possible de choisir le même pas spatial sur les deux grille grossière et fine. Nous nous attendons, toutefois, que la solution se caractérise par une valeur très élevée de la dérivée temporels, dans une région de sous domaine, a, dans cette même région, des gradients de l'espace relativement élevé aussi. Pour cette raison, l'utilisation d'un pas de l'espace Δx sur la grille fine peut-être pas suffisant pour représenter les phénomènes très rapide qui se produisent dans cette région. En outre, si nous avons effectuer le pas d'espace sur la grille fine avec le même pas que celui sur la grille grossière, la discrétisation de l'erreur "Defect Correction" sur la grille fine pourrait être largement dominé par la composante spatial, qui pourrait rendre l'effet de résoudre le problème, sur la grille fine avec un pas $\delta t < \Delta t$, inutile. Dans le Cas de \mathcal{LDC} le "Defect Correction", qui est utilisé pour calculer une approximation globale plus précise sur chaque sous domaine, améliore la solution, sur la grille grossière, non seulement dans la partie temporel de l'erreur mais aussi dans la composante spatiale. Dans la pratique, les paramètres Δt , δt , Δx et δx sont choisis sur la base du problème physique qui doit être résolu, et sur la discrétisation des schémas adoptés globalement et localement.

1.5.3 Propriétés de la méthode \mathcal{LDC}

Dans cette section, nous discutons de quelques propriétés de la technique \mathcal{LDC} pour le problème de convection-diffusion. Le lemme suivant montre que, une fois l'approximation, sur la grille grossière, ne change pas sur chaque interface entre deux domaines(en Indice 2), alors l'algorithme \mathcal{LDC} converge.

Lemme 1. Si $u_{git}^{Indice2} = u_{git-1}^{Indice2}$ pour une certaine itération it , alors l'itération \mathcal{LDC} converge et

$$u_{gl}^{Indice2} = u_{git}^{Indice2}, \text{ pour tout } l = it, it + 1....$$

Le théorème suivant affirme que, si l'itération \mathcal{LDC} converge, les approximations sur la grille fine et grossière coïncident sur les points communs entre les deux grilles.

Théorème 1. *Supposons que l'itération \mathcal{LDC} converge. Alors la projection de u_f sur la grille grossière, pour chaque sous domaine, est égale à u_g , à savoir pour $i = 1...n + 1$*

$$u_{g_i}^j = Ru_i^k, \text{ pour } j = \text{Indice1}...\text{Indice2}, k = 1....Nt$$

1.6 Applications Numériques

Dans cette section, nous présentons des exemples numériques. Pour chaque exemple, nous comparons les deux schémas utilisés sur la grille grossière (Euler Explicite et Runge Kutta 4). Nous comparons aussi la technique de \mathcal{LDC} avec celle du solveur de grille uniforme standard. Nous prouvons que \mathcal{LDC} est plus efficace que le solveur de la grille uniforme surtout si la solution approchée sur la grille grossière converge vers la solution analytique.

Afin de valider la méthode \mathcal{LDC} , nous avons, tout d'abord, commencer par les exemples basiques tels que,

$$u(x, t) = t, xt...$$

en utilisant Euler Explicite sur la grille grossière. Pour chacun de ces cas tests, nous avons obtenus la convergence de la solution approchée, sur la grille grossière, vers la solution analytique au bout d'une seule itération, c'est qui est normale car Euler Explicite est exacte pour ces types d'exemple.

1.6.1 Exemple 1

Pour cet exemple, nous présentons les résultats d'une expérience numérique visant à démontrer la robustesse de la méthode \mathcal{LDC} en la comparant à celle du solveur de grille uniforme. nous choisissons $E = 1$, $T = 1$ et nous résolvons le problème suivant:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} - 0.01 \frac{\partial^2 u}{\partial x^2} = f \quad (1.6.1)$$

La condition initiale, les conditions de Dirichlet sur la frontière (en $x = 0$ et $x = E$) et le terme source f sont choisis de telle sorte que la solution analytique exacte du problème est:

$$u(x, t) = \cos(\omega_0 x) \sin(\omega_1 t)$$

Où ω_0 et ω_1 sont deux réels que nous donnerons leur valeurs par la suite.

Le problème ci-dessus est résolu en utilisant la méthode \mathcal{LDC} avec différentes valeurs de Δt , Δx , δt , δx et τ (voir tableau 1.1).

La discrétisation du temps est réalisée en utilisant Euler Explicite sur la grille grossière. Nous fixons la valeur de N_{dom} à 3.

Comme mesure de précision de la solution numérique à $t_{N+1} = 1$ utilisant \mathcal{LDC} , nous calculons la norme infinie définie par:

$$\xi_{\infty} = \max(u_g^{N+1} - u(t_{N+1}))$$

Avec u_g^{N+1} est la solution approchée numériquement calculée à $t_{N+1} = 1$ et $u(t_{N+1})$ est la projection sur la grille grossière de la solution exacte à l'instant final.

Pour la grille uniforme, nous utilisons un pas du temps très petit que nous le notons δt_{unif} et un pas spatial δx_{unif} très petit aussi. Les valeurs de δt_{unif} et δx_{unif} sont choisies de telle manière qu'elles sont égaux aux plus petites valeurs de pas temporel et spatial sur la grille fine.

Comme comparaison, nous résolvons le problème (1.8.1) en utilisant un pas d'espace et un pas de temps, sur la grille uniforme, qui sont égaux à:

$$\delta t_{unif} = \delta t = \frac{T_{Nt} - T_1}{N_f}$$

$$\delta x_{unif} = \delta x = \frac{\Delta x}{\tau}$$

Nous attendons à ce que les résultats obtenues avec la méthode \mathcal{LDC} soient à peu près de même précision que la solution approchée résolue par le problème (1.8.1) sur la grille uniforme.

Pour chacun de ces calculs, les pas spatiaux sont choisis de telle sorte que l'erreur d'espace ont le même ordre que ceux du temps. Sur la grille uniforme, aussi, nous mesurons la norme infinie.

Résultats dans le tableau 1.1 montrent que, dans tous les cas que nous considérons, la méthode \mathcal{LDC} est en mesure d'atteindre une meilleure précision de la norme infinie que celle de la grille uniforme et pour un nombre d'itération qui est inférieur à 10.

A partir du tableau 1.1, nous pouvons voir que dans notre exemple, quand nous utilisons la méthode \mathcal{LDC} , nous avons obtenue une norme infinie qui est inférieure à celle calculée sur la grille uniforme. Ce gain augmente quand nous raffinons le pas du temps et aussi le pas d'espace.

Table 1.1: Comparaison entre \mathcal{LDC} et solveur de grille uniforme

Pas du temps et Pas d'espace					ξ_∞	
Δt	Δx	τ	δt_{unif}	δx_{unif}	\mathcal{LDC}	Grille Uniforme
0.04	0.2	3	$6.67 e^{-3}$	$6.6 e^{-2}$	0.0641	0.0797
		5	$4 e^{-3}$	$4 e^{-2}$	0.0493	0.0554
		7	$1.4 e^{-3}$	$2.86 e^{-2}$	0.0396	0.0433
0.01	0.1	3	$1.6 e^{-3}$	$3.3 e^{-2}$	0.0471	0.0494
		5	$1 e^{-3}$	$2 e^{-2}$	0.0309	0.0323
		7	$5.1 e^{-4}$	$1.428 e^{-2}$	0.0231	0.0241
$2.5 e^{-3}$	0.05	3	$3.4 e^{-4}$	$1.67 e^{-2}$	0.0282	0.0281

Ces valeurs ont été pris pour ω_0 et ω_1 égaux à 25 et 50, respectivement, et pour $N_f = 200$. Pour chacun de ces cas, nous avons obtenue le calcul de la norme infinie au bout de 4 ou 5 itérations.

Ce tableau montre que la méthode \mathcal{LDC} permet d'obtenir, presque, une même précision que le solveur de la grille uniforme. Bien entendu, Cette méthode effectue moins de calcul que la méthode de la grille uniforme.

Pour conclure, \mathcal{LDC} est une méthode plus efficace que celle de la grille uniforme, qui utilise le même pas du temps et d'espace que la méthode \mathcal{LDC} sur la grille fine.

Par la suite, nous posons que $\omega_0 = \omega_1 = 50$. Nous choisissons un pas d'espace assez grand sur la grille grossière, $\Delta x = \frac{1}{5}$. Nous fixons la valeur de la Condition de stabilité "CFL", Courant-Friedrichs-Lewy, sur la grille grossière, à 1 et nous calculons le pas du temps de la manière suivante:

$$(1.8.2) \quad \begin{cases} dt1 = \frac{CFL \Delta x}{a} \\ dt2 = \frac{CFL \Delta x^2}{\nu} \\ \Delta t = \min(dt1, dt2) \end{cases}$$

Et sur la grille fine, nous fixons $\tau = 15$ pour qu'il y est convergence de la solution approchée vers la solution exacte. Le nombre de domaine est $N_{dom} = 2$. La CFL sur cette grille est égale à 0.25 et nous calculons δt par (1.8.2) mais en remplaçant l'ancienne CFL par 0.25 et Δx par δx .

Sur les figures suivantes, nous présentons les solutions approchées, sur la grille grossière, calculées en utilisant la méthode \mathcal{LDC} et le solveur de la grille uniforme ainsi que la solution exacte à l'instant $t_{N+1} = 1$ et en fonction de pas d'espace.

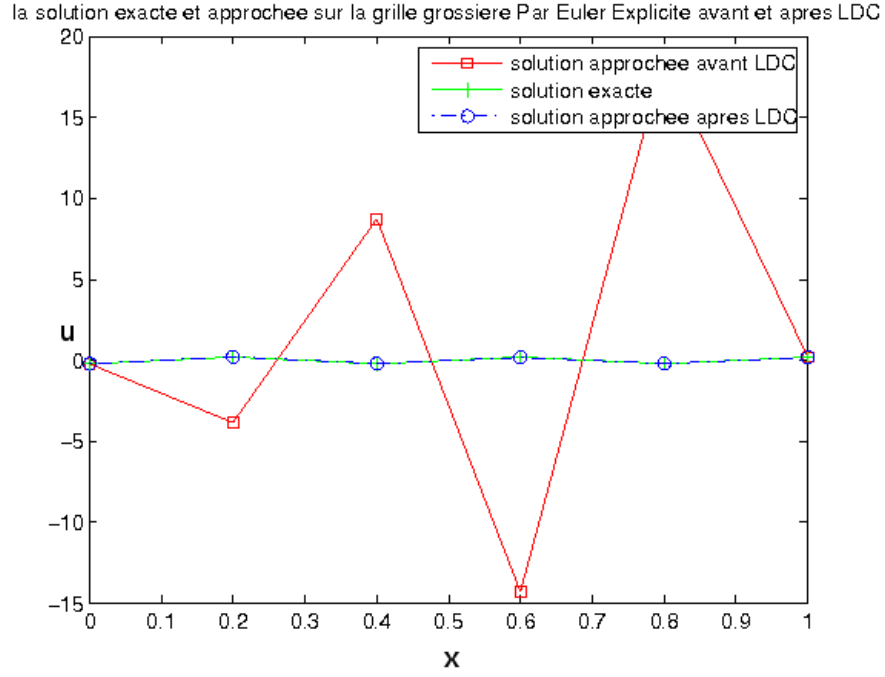


Figure 1.2: Tracés des solutions approchées calculées par Euler explicite et la solution exacte, à $t_{N+1} = 1$

Sur cette figure, nous comparons la solution approchée sur la grille grossière, calculée par Euler Explicite, colorée en rouge, avant que nous utilisons la méthode \mathcal{LDC} (à l'itération 0), à celle calculée après application de la méthode \mathcal{LDC} , qui est colorée en bleu. Nous remarquons que la solution approchée calculée par \mathcal{LDC} converge vers la solution exacte, qui est en vert, alors que celle calculée à l'itération 0 ne converge pas.

Pour mieux observer la convergence de cette solution approchée obtenue en utilisant l'algorithme \mathcal{LDC} vers la solution analytique du problème, nous traçons sur la figure suivante que ces deux solutions et nous observons qu'elles sont confondues ou presque. Il y a eu convergence au bout de 2 itérations.

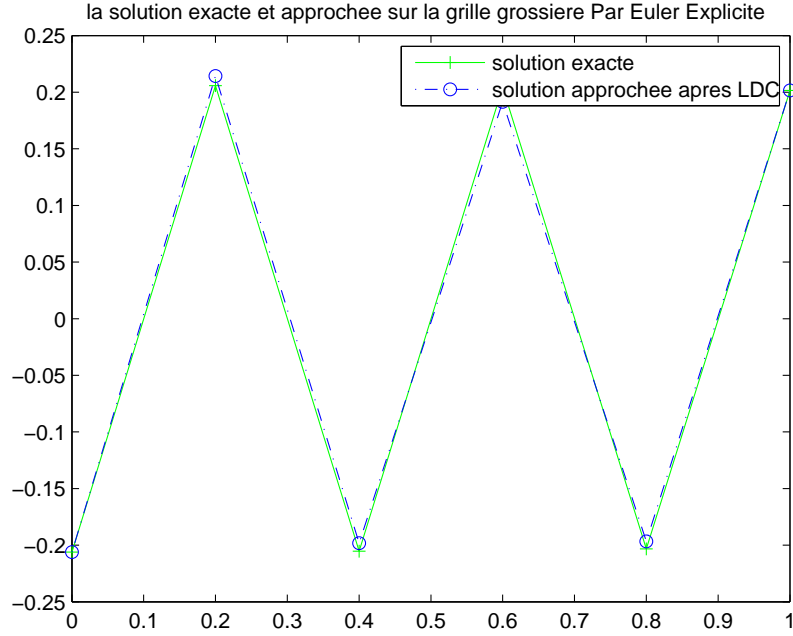


Figure 1.3: Solution exacte à $t_{N+1} = 1$ et solution numérique calculée par \mathcal{LDC}

Maintenant, nous comparons la solution approchée calculée par la méthode \mathcal{LDC} avec celle calculée sur une grille uniforme. Cette dernière a un pas du temps et un pas d'espace égaux au plus petit de pas du temps et d'espace de la grille fine. Nous présentons sur la figure suivante, le tracé des solutions approchées calculées par \mathcal{LDC} et le solveur de la grille uniforme ainsi que la solution exacte.

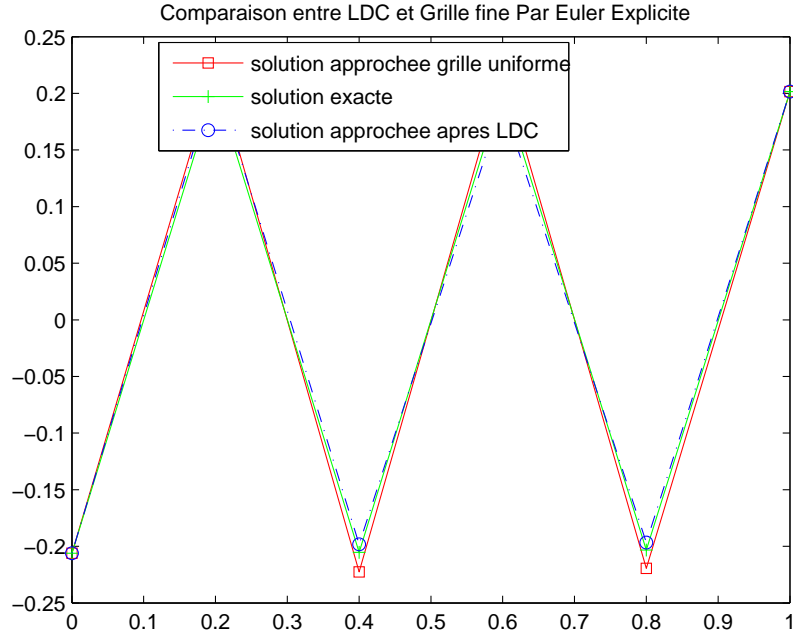


Figure 1.4: Comparaison entre la méthode \mathcal{LDC} et solveur de grille uniforme

Nous observons sur cette figure que la solution approchée calculée par la méthode \mathcal{LDC} (en bleu) est la plus proche de la solution analytique (en vert) que la solution approchée,

projetée sur la grille grossière, calculée sur la grille uniforme (en rouge). Ce qui prouve que \mathcal{LDC} est une méthode très efficace.

1.6.2 Exemple 2

Dans cette section, nous présentons une expérience numérique qui a pour objectif de comparer entre le schéma d'Euler Explicite et le schéma de Runge Kutta 4. Nous résolvons le problème (1.7.1) mais cette fois ci le terme source, la condition initiale et les conditions de Dirichlet sont calculés à partir de la solution analytique exacte:

$$u(x, t) = \cos(x) \exp(2t)$$

Les valeurs des paramètres sur la grille grossière, sont:

$$\Delta x = \frac{1}{5}$$

$$\Delta t = 5 e^{-2}$$

Pour le nombre du domaine N_{dom} , nous l'avons fixé à 2.

Nous commençons, tout d'abord, avec le schéma de Runge Kutta 4 sur la grille grossière. Nous avons obtenue la convergence de la solution approchée, calculée par \mathcal{LDC} , vers la solution exacte analytique au bout d'une seule itération sachant que nous avons calculé le pas du temps fin sur la grille fine comme précédemment, i.e en utilisant le système (1.8.2), pour une CFL qui est égale à 0.25 et pour $\delta x = \Delta x$.

Sur la figure 1.5, nous traçons les solutions approchées, une calculée en utilisant Runge Kutta 4 sur la grille grossière (celle en rouge) à l'itération 0 et l'autre en utilisant l'algorithme \mathcal{LDC} (celle en bleu), et aussi la solution exacte. Nous observons, sur ce graphe, que la solution approchée par \mathcal{LDC} est confondue avec la solution exacte alors que l'autre reste loin.

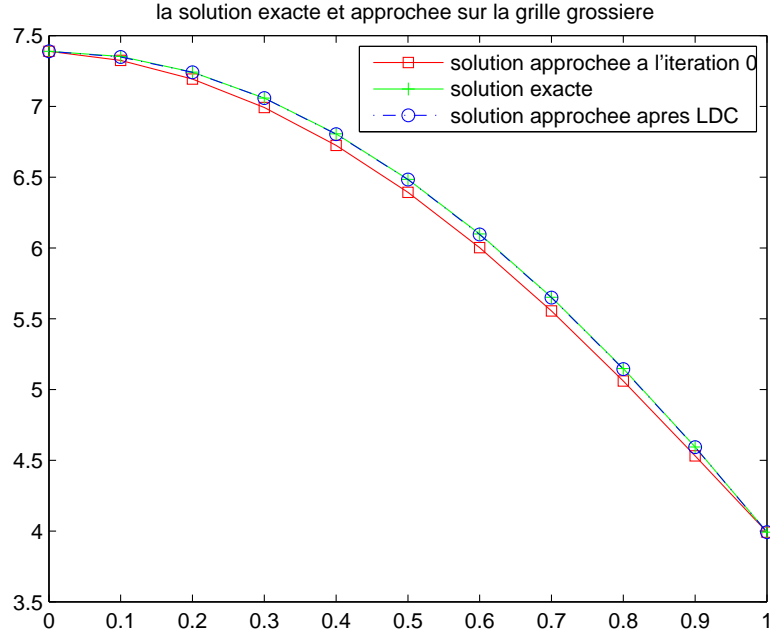


Figure 1.5: Solutions approchées par Runge Kutta 4 et solution exacte en fonction du pas d'espace à $t_{N+1} = 1$

Puis, nous comparons aussi la solution approchée obtenue en résolvant le problème (1.7.1) par \mathcal{LDC} à celle de solveur de grille uniforme et de nouveau, nous remarquons que \mathcal{LDC} converge vers la solution analytique avant la solution approchée calculée sur la grille uniforme (avec un pas du temps $\delta t_{unif} = 2.5 e^{-2}$).

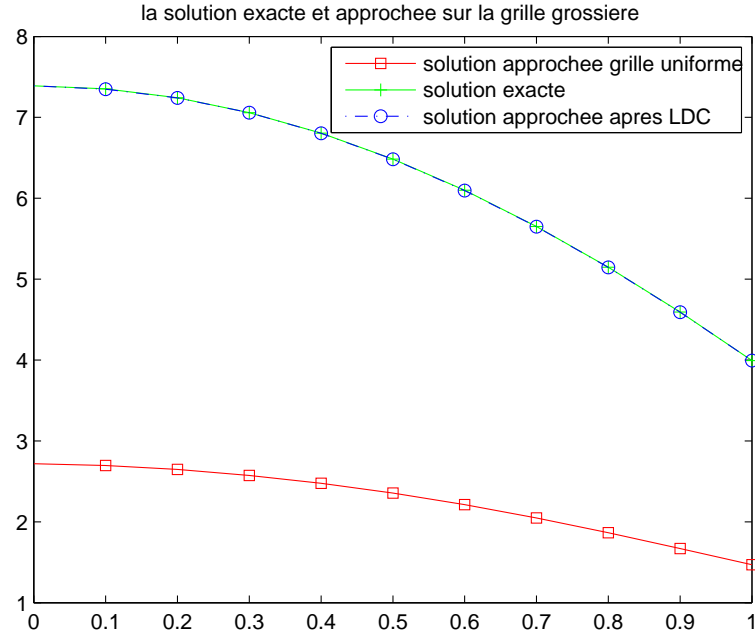


Figure 1.6: Comparaison entre \mathcal{LDC} et solveur de grille uniforme en utilisant RK4

Pour cette exemple, nous avons calculé la norme infinie pour chacun de ces cas et nous avons obtenu:

1. Méthode \mathcal{LDC} : $\xi_\infty = 0.0035$
2. Solveur de grille uniforme : $\xi_\infty = 0.0971$

Maintenant, pour les mêmes paramètres sur la grille grossière et la grille fine, nous avons tracé les solutions approchées calculées en utilisant Euler Explicite, sur la grille grossière et sur la grille uniforme, ainsi que la solution exacte. voici c'est que nous avons obtenue.

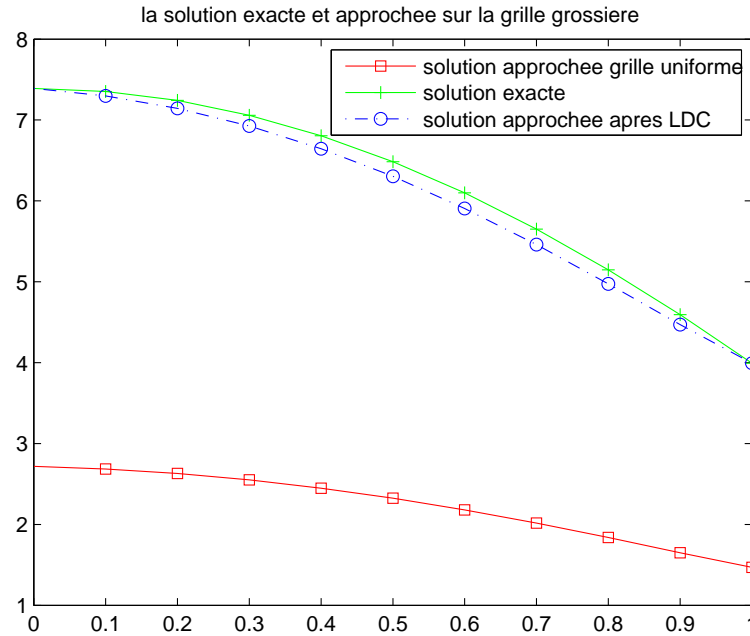


Figure 1.7: Comparaison entre \mathcal{LDC} et solveur de grille uniforme en utilisant Euler Explicite pour $\tau = 1$

Nous remarquons, que les solutions approchées calculées par les deux méthodes sont très loin de la solution exacte. Ce qui prouve que le schéma de Runge Kutta 4 est plus précis que le schéma d'Euler Explicite. Pour ce type d'exemple, il y a eu convergence de la solution approchée par \mathcal{LDC} vers la solution exacte au bout de 2 itérations et en augmentant $\tau = 5$ au lieu de 1. Alors que la solution approchée sur la grille uniforme ne converge toujours pas vers la solution analytique du problème à l'instant final. La figure 1.8, prouve ce que nous venons d'interpréter.

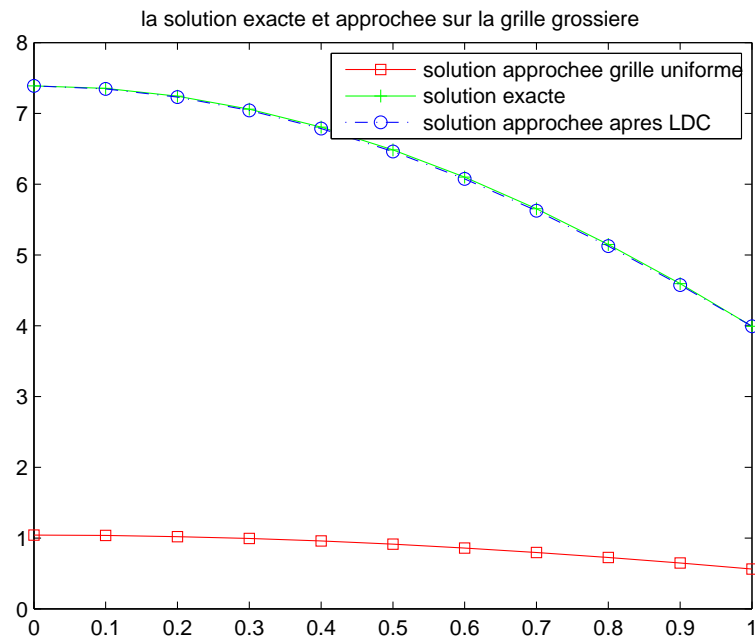


Figure 1.8: Comparaison entre \mathcal{LDC} et solveur de grille uniforme en utilisant Euler Explicite pour $\tau = 5$

2.1 Résolution numérique d'EDP

Retournons le problème de convection diffusion résolu dans le chapitre 1 (Cas 1D) et regardons, cette fois, ce problème en deux dimensions.

L'équation de convection diffusion que nous cherchons à résoudre est la suivante:

$$\frac{\partial u}{\partial t} + \vec{a} \cdot \nabla u - \nu \nabla^2 u = f$$

Où \vec{a} est la vitesse d'advection, ν est le terme de la viscosité et f est le terme source.

En deux dimensions, cela peut être écrit comme suit:

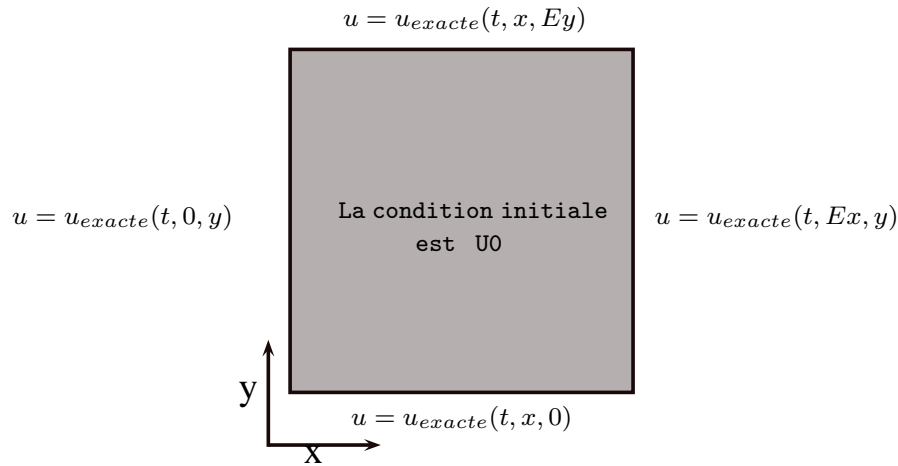
$$\frac{\partial u}{\partial t} + \begin{pmatrix} a_x \\ a_y \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} - \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f$$

\Rightarrow

$$\frac{\partial u}{\partial t} + \left(a_x \frac{\partial u}{\partial x} + a_y \frac{\partial u}{\partial y} \right) - \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f \quad (2.1.1)$$

Avec a_x , a_y et ν sont des réels positifs.

Nous allons résoudre ce problème sur le domaine suivant,



2.1.1 Discrétisation du problème en espace

Afin de résoudre cette EDP numériquement, nous devons, d'abord, réduire l'EDP soit en un système d'équations (éventuellement non-linéaire), ou en un système d'équations différentielles, EDO, qui peut être une marche en avant dans le temps. Pour ce faire, le domaine de la solution doit être discrétiser. Nous allons changer le problème de convection-diffusion 2D en un système d'équations linéaires en utilisant les différences finies. Bien évidemment, ils existent d'autres méthodes pour discrétiser le domaine spatial tels que la méthode de volume finis et éléments finis.

Dans la suite, nous utilisons la méthode des lignes. C'est une technique de résolution des équations aux dérivées partielles (EDP), dans laquelle tous, mais une seule dimension est discrétisée.

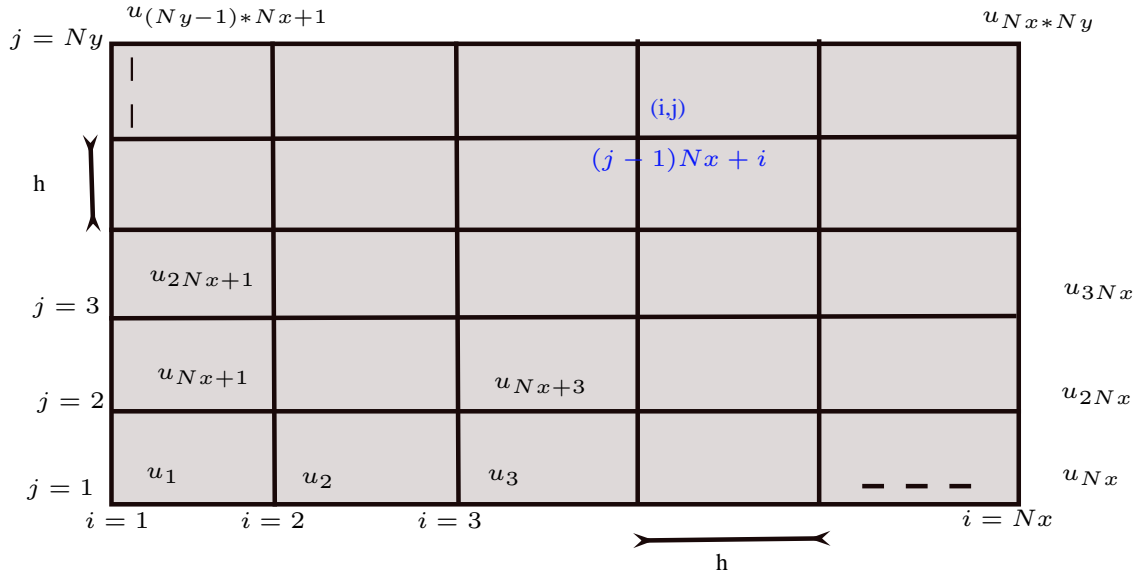
Dans la méthode des lignes, nous discrétisons le domaine spatial du problème de convection-diffusion 2D, pour produire un ensemble des EDO. Pour ce faire, nous utilisons les différences finies d'ordre 2, DF2.

Pour procéder, nous plaçons une grille sur notre domaine de solution. Pour simplifier, nous utiliserons une grille spatiale régulière et à distance égale, i.e $\Delta x = \Delta y = h$.

Nous notons:

- N_x : Le nombre de points sur la grille dans la direction de x .
- N_y : Le nombre de points sur la grille dans la direction de y .
- CFL : Condition de stabilité de Courant-Friedrichs-Lewy.
- Δt : Le pas du temps sur la grille grossière.
- N_t : Le nombre de points sur la grille du temps.
- N_{dom} : Le nombre de domaine sur l'intervalle temps.
- δt : le pas du temps sur la grille fine.

A noter que le pas du temps, sur la grille grossière ou sur la grille fine, est calculé en utilisant la formule (1.8.2).



L'indice du noeud général (i,j) devient

$$k = (j - 1) Nx + i$$

De sorte que,

$$u_{i,j} = u_k$$

Nous pouvons maintenant discrétiser (2.1.1) pour trouver une EDO de la façon suivante:

$$\frac{du_{i,j}}{dt} = -(a_x \frac{\partial u}{\partial x}|_{i,j} + a_y \frac{\partial u}{\partial y}|_{i,j}) + \nu(\frac{\partial^2 u}{\partial x^2}|_{i,j} + \frac{\partial^2 u}{\partial y^2}|_{i,j}) + f_{i,j}(t) \quad (2.1.2)$$

Pour tout $i = 1....Nx$ et pour tout $j = 1....Ny$

Nous avons besoin d'estimer le gradient et Laplace au noeud k. Ceux-ci peuvent être obtenus à partir de différences finies. Nous pouvons supposer un interpolateur linéaire, comme nous l'avons expliqué dans le chapitre 1.

Nous estimons la dérivée première en utilisant la formule de différences finies rétrograde, car a_x et a_y sont des entiers positifs, i.e:

$$\frac{\partial u}{\partial x} \simeq \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

Ensuite, nous estimons la dérivée seconde par la formule (1.4) de telle sorte que:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

Si nous faisons une approximation pareille des dérivées première et seconde dans la direction y, nous pouvons écrire l'équation (2.1.2) au noeud k, comme, pour $i = 2....Nx - 1$ et pour $j = 2....Ny - 1$

$$\begin{aligned} \frac{du_{i,j}}{dt} = & -(a_x \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + a_y \frac{u_{i,j} - u_{i,j-1}}{\Delta y}) + \nu \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \\ & + \nu \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} + f_{i,j}(t) \end{aligned} \quad (2.1.3)$$

Or nous avons pour

$$\begin{aligned} \{i+1, j\} & \rightsquigarrow \overbrace{(j-1)Nx + i}^k + 1 \rightsquigarrow k+1 \\ \{i, j+1\} & \rightsquigarrow (j+1 - 1)Nx + i = \overbrace{(j-1)Nx + i}^k + Nx \rightsquigarrow k+Nx \end{aligned}$$

Ainsi, nous obtenons:

$$\begin{aligned} \{i, j\} & \longleftrightarrow k \\ \{i+1, j\} & \longleftrightarrow k+1 \\ \{i-1, j\} & \longleftrightarrow k-1 \\ \{i, j+1\} & \longleftrightarrow k+Nx \\ \{i, j-1\} & \longleftrightarrow k-Nx \end{aligned}$$

Et donc en terme d'indice des noeuds, l'équation (2.1.3) devient, sachant que nous avons supposer que la grille spatiale est régulière ($\Delta x = \Delta y = h$),

$$\begin{aligned} \frac{du_k}{dt} = & -\frac{1}{h}(a_x(u_k - u_{k-1}) + a_y(u_k - u_{k-Nx})) + \frac{\nu}{h^2}(u_{k+1} - 2u_k + u_{k-1}) \\ & + \frac{\nu}{h^2}(u_{k+Nx} - 2u_k + u_{k-Nx}) + f_k(t) \end{aligned} \quad (2.1.4)$$

\Rightarrow

$$\begin{aligned} \frac{du_k}{dt} = & \overbrace{\left(\frac{a_y}{h} + \frac{\nu}{h^2}\right) u_{k-Nx}}^{e3} + \overbrace{\left(\frac{a_x}{h} + \frac{\nu}{h^2}\right) u_{k-1}}^{e2} \\ & - \overbrace{\left(\frac{a_x + a_y}{h} + \frac{4\nu}{h^2}\right) u_k}^{e1} + \frac{\nu}{h^2} u_{k+1} + \frac{\nu}{h^2} u_{k+Nx} + f_k(t) \end{aligned} \quad (2.1.5)$$

\Rightarrow

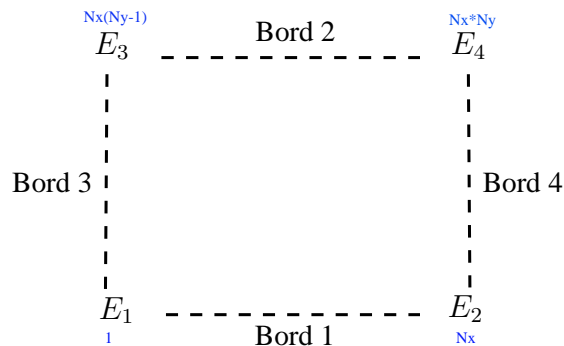
$$\frac{du_k}{dt} = e3 u_{k-Nx} + e2 u_{k-1} - e1 u_k + \frac{\nu}{h^2} u_{k+1} + \frac{\nu}{h^2} u_{k+Nx} + f_k(t) \quad (2.1.6)$$

Il y a deux façons de procéder pour tenir compte des noeuds sur la frontière,

1. Inclure les équations sur la frontière comme un ensemble d'équations algébriques qui doivent être résolus à coté du système des EDO produit par (2.1.4).
2. Eliminer les équations sur la frontière de l'équation donnée par (2.1.4) et les rajouter à la fin de la résolution.

Ici nous allons utiliser la deuxième option. Toutefois, il faut noter que les solveurs explicite ou implicite sur Matlab peuvent être utilisés pour résoudre des systèmes qui contiennent à la fois les EDO et aussi les équations algébriques.

Ainsi, pour éliminer les conditions de Dirichlet sur la frontière, nous notons:



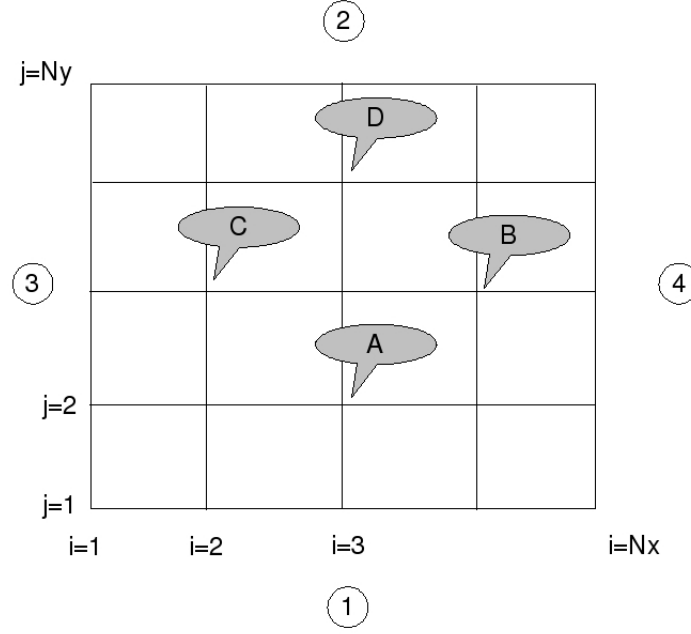
Nous donnons quelque définition,

- 1) $u_{d1} = u_{exacte}(t, x, 0)$
- 2) $u_{d2} = u_{exacte}(t, x, Ey)$
- 3) $u_{d3} = u_{exacte}(t, 0, y)$
- 4) $u_{d4} = u_{exacte}(t, Ex, y)$

Et donc en terme d'indice des noeuds, nous avons,

- 1) Les noeuds sur le Bord 1 $\longrightarrow u_{k-Nx} = u_{d1}$
- 2) Les noeuds sur le Bord 2 $\longrightarrow u_{k+Nx} = u_{d2}$
- 3) Les noeuds sur le Bord 3 $\longrightarrow u_{k-1} = u_{d3}$
- 4) Les noeuds sur le Bord 4 $\longrightarrow u_{k+1} = u_{d4}$

Prenons l'exemple suivant pour mieux comprendre les conditions de Dirichlet sur les bords 1, 2, 3 et 4:



Et donc par exemple, pour,

★ le noeud A:

$$\frac{du_k}{dt} = e2 u_{k-1} - e1 u_k + \frac{\nu}{h^2} u_{k+1} + \frac{\nu}{h^2} u_{k+Nx} + f_k(t) + e3 u_{d1}$$

★ le noeud B:

$$\frac{du_k}{dt} = e3 u_{k-Nx} + e2 u_{k-1} - e1 u_k + \frac{\nu}{h^2} u_{k+Nx} + f_k(t) + \frac{\nu}{h^2} u_{d4}$$

★ le noeud C:

$$\frac{du_k}{dt} = e3 u_{k-Nx} - e1 u_k + \frac{\nu}{h^2} u_{k+1} + \frac{\nu}{h^2} u_{k+Nx} + f_k(t) + e2 u_{d3}$$

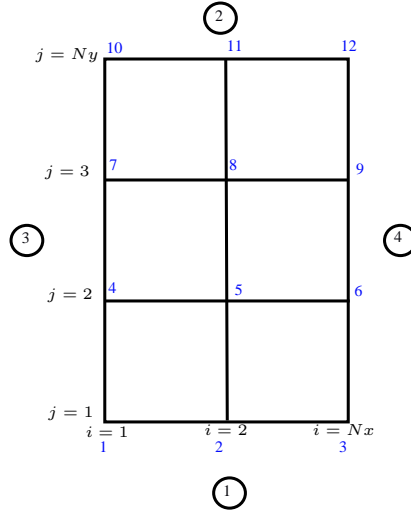
★ le noeud D:

$$\frac{du_k}{dt} = e3 u_{k-Nx} + e2 u_{k-1} - e1 u_k + \frac{\nu}{h^2} u_{k+1} + f_k(t) + \frac{\nu}{h^2} u_{d2}$$

Ainsi, l'équation (2.1.6) peut être écrit sous la forme

$$\frac{d\underline{u}}{dt} = M\underline{u} + \underline{b} + \underline{F} \quad (2.1.7)$$

Avec M est une matrice de taille $N_x \times N_y$, b est un vecteur de même dimension que M (b est nul pour les noeuds internes mais contient les conditions de Dirichlet pour les noeuds qui sont à coté de la frontière, comme A, B, etc...) et F est un vecteur de taille $N_x \times N_y$, il contient les valeurs du terme source f en chaque noeud. Ceci est écrit sous forme matricielle dans le cas où $N_x = 3$ et $N_y = 4$,



Pour cet exemple, nous écrivons la matrice M sous la forme:

$$M_3 \left\{ \begin{array}{c} \overbrace{\begin{bmatrix} -e1 & \frac{\nu}{h^2} & 0 \\ e2 & -e1 & \frac{\nu}{h^2} \\ 0 & e2 & -e1 \end{bmatrix}}^{M_1} \quad \overbrace{\begin{bmatrix} \frac{\nu}{h^2} & \text{elliptical} \\ \text{elliptical} & \frac{\nu}{h^2} \end{bmatrix}}^{M_2} \quad \begin{bmatrix} \text{elliptical} \end{bmatrix} \\ \begin{bmatrix} e3 & \text{elliptical} \\ \text{elliptical} & e3 \end{bmatrix} \quad \begin{bmatrix} M_1 \end{bmatrix} \quad \begin{bmatrix} M_2 \end{bmatrix} \\ \begin{bmatrix} \text{elliptical} \end{bmatrix} \quad \begin{bmatrix} M_3 \end{bmatrix} \quad \begin{bmatrix} M_1 \end{bmatrix} \end{array} \right\}$$

Ainsi, le vecteur b s'écrit sous la forme:

$$\begin{bmatrix} e3 u_{d1} + e2 u_{d3} \\ e3 u_{d1} + \frac{\nu}{h^2} u_{d4} \\ 0 \\ \frac{\nu}{h^2} u_{d4} \\ e2 u_{d3} + \frac{\nu}{h^2} u_{d2} \\ \frac{\nu}{h^2} (u_{d2} + u_{d4}) \end{bmatrix}$$

2.1.2 Euler Explicite

Pour résoudre ce problème en temps, nous choisissons, tout d'abord, le schéma d'Euler Explicite.

Nous initialisons la condition initiale,

$$u_k^1 = u0_k$$

pour $k = (j-1) * Nx + i$, pour $i = 1...Nx$ et $j = 1...Ny$

Ainsi, l'équation (2.1.7) s'écrit, pour tout $n = 1 Nt - 1$

$$\frac{\underline{u}^{n+1} - \underline{u}^n}{\Delta t} = M\underline{u}^n + \underline{b}^n + \underline{F}^n$$

\Rightarrow

$$\underline{u}^{n+1} = \underline{u}^n + \Delta t \underbrace{(M\underline{u}^n + \underline{b}^n + \underline{F}^n)}_{L(\underline{u}^n)} \quad (2.1.8)$$

Et donc, nous écrivons (2.1.8) résolue par Euler Explicite comme suit

$$\begin{array}{lcl} k1 & = & L(u_k^n) + F_k^n \\ u_k^{n+1} & = & u_k^n + \Delta t \ k1 \end{array}$$

Condition de stabilité du schéma Euler Explicite

Pour étudier la stabilité de ce schéma, nous posons $f \equiv 0$.

En appliquant le schéma d'Euler Explicite en temps pour l'équation (2.1.5), pour $n = 1...Nt-1$, pour $k = (j-1)Nx + i$, pour $i = 2...Nx-1$ et pour $j = 2...Ny-1$, nous obtenons,

$$u_k^{n+1} = u_k^n + \Delta t \left(\left(\frac{a_y}{h} + \frac{\nu}{h^2} \right) u_{k-Nx}^n + \left(\frac{a_x}{h} + \frac{\nu}{h^2} \right) u_{k-1}^n - \left(\frac{a_x + a_y}{h} + \frac{4\nu}{h^2} \right) u_k^n + \frac{\nu}{h^2} u_{k+1}^n + \frac{\nu}{h^2} u_{k+Nx}^n \right)$$

\Rightarrow

$$\begin{aligned} u_k^{n+1} = & \underbrace{\Delta t \left(\frac{a_y}{h} + \frac{\nu}{h^2} \right)}_{>0} u_{k-Nx}^n + \underbrace{\Delta t \left(\frac{a_x}{h} + \frac{\nu}{h^2} \right)}_{>0} u_{k-1}^n + \\ & \left(1 - \Delta t \left(\frac{a_x + a_y}{h} + \frac{4\nu}{h^2} \right) \right) u_k^n + \underbrace{\frac{\nu \Delta t}{h^2}}_{>0} u_{k+1}^n + \underbrace{\frac{\nu \Delta t}{h^2}}_{>0} u_{k+Nx}^n \end{aligned} \quad (2.1.9)$$

Par le critère de positivité, nous en deduisons que ce schéma est stable, si la condition suivante est vérifiée:

$$1 - \frac{(a_x + a_y)\Delta t}{h} - \frac{4\nu\Delta t}{h^2} > 0$$

\Rightarrow

$$\frac{(a_x + a_y)}{h} + \frac{4\nu}{h^2} < \frac{1}{\Delta t}$$

Dans la suite nous posons,

$$dt1 = \frac{h}{a_x + a_y}$$

$$dt2 = \frac{h^2}{4\nu}$$

Et nous calculons le pas du temps de la manière suivante:

$$\Delta t = \min(dt1, dt2)$$

2.1.3 Runge Kutta 4

Nous utilisons Runge Kutta 4 sur la grille grossière pour améliorer la précision du schéma.

Pour le problème de convection-diffusion en 2D le schéma de Runge Kutta 4 est le suivant, pour $n = 1 \dots Nt - 1$ et pour $k = (j - 1)Nx + i$, $i = 2 \dots Nx - 1$, $j = 2 \dots Ny - 1$

$$\begin{aligned} k1 &= \Delta t (L(u_k^n) + F_k(t_n)) \\ k2 &= \Delta t (L(u_k^n + \frac{k1}{2}) + F_k(t_n + \frac{\Delta t}{2})) \\ k3 &= \Delta t (L(u_k^n + \frac{k2}{2}) + F_k(t_n + \frac{\Delta t}{2})) \\ k4 &= \Delta t (L(u_k^n + k3) + F_k(t_n + \Delta t)) \\ u_k^{n+1} &= u_k^n + \frac{k1 + 2 k2 + 2 k3 + k4}{6} \end{aligned}$$

2.2 Méthode \mathcal{LDC}

Pour cette section, nous utilisons la méthode \mathcal{LDC} , que nous avons définis dans la section (1.5) du chapitre 1.

Sur le domaine de temps, nous utilisons la même discrétisation en sous domaine(le nombre total est N_{dom}) que dans le chapitre précédent(page 20). Nous notons N le nombre de points sur chaque D_{idom} , $idom = 1....N_{dom}$ et par $t_0 = T_{Indice1} = T_1, T_2, ..., T_{N-1}, T_N = T_{Indice2} = t_1$ les noeuds grossiers sur chaque domaine temporel.

Sur chaque sous domaine temporel, nous initialisons une grille fine de pas $\delta t < \Delta t$ et de nombre de points Nt_f . Nous notons les noeuds de la grille fine par $t_0 = t_{f1}, t_{f2}, ..., t_{f_{Nt_f+1}} = t_1$

1^{ère} étape: Résolution de (2.1.1) sur la grille grossière

Afin de résoudre le problème (2.1.1) sur la grille grossière, de pas Δt , nous utilisons un schéma de type Euler Explicite ou Runge Kutta 4 comme décrit précédemment, page 44 et 46. Ainsi, nous obtenons une solution approchée que nous notons u_g .

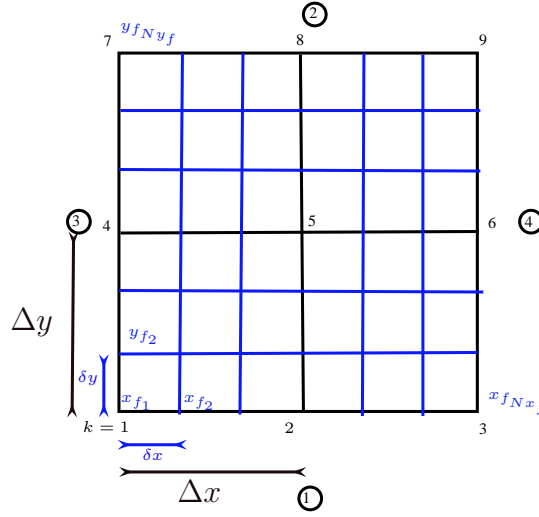
2^{ème} étape: Méthode \mathcal{LDC}

Résolution de (2.1.1) sur la grille fine

Sur la grille fine, nous discrétisons le domaine en temps comme dans 1.5. Nous utilisons u_g pour initialiser une solution approchée(que nous notons u_f , calculée par le schéma d'Euler Explicite en temps et par différences finis 2 en espace) en Indice 1 de chaque sous domaine en appliquant la condition de Dirichlet. i.e $u_f(t_0(x, y)) \approx u_g(T_{Indice1}, x, y)$.

Nous modifions la discrétisation du domaine d'espace pour améliorer la solution approchée locale. Nous supposons que le pas d'espace fin, suivant la direction de x et y , est régulier, i.e $\delta x = \delta y = h_f$ puisque $\delta x = \frac{\Delta x}{\tau} = \frac{h}{\tau}$ et $\delta y = \frac{\Delta y}{\tau} = \frac{h}{\tau}$, τ est un entier strictement positif.

Sur la figure suivante, nous présentons un exemple de discrétisation du domaine spatial, pour $Nx = Ny = 3$ et $\tau = 3$



Nous notons Nx_f et Ny_f les nombres des points sur la grille d'espace fine suivant la direction de x et y , respectivement. Les bords 1, 2, 3 et 4 sont définis sur la grille spatiale fine par

- Bord 1 : $1 \longrightarrow Nx_f$.
- Bord 2 : $Nx_f * (Ny_f - 1) + 1 \longrightarrow Nx_f * Ny_f$.
- Bord 3 : $1 \longrightarrow Nx_f * (Ny_f - 1) + 1$ par pas de Nx_f .
- Bord 4 : $Nx_f \longrightarrow Nx_f * Ny_f$ par pas de Nx_f .

Nous commençons, tout d'abord, par transformer le vecteur colonne, qui correspond à la solution approchée calculée sur la grille de temps grossière u_g en Indice 1 de chaque domaine, en une matrice V_g de taille (Nx, Ny) de la manière suivante:

$$V_g(i, j) = u_g(k, Indice1) ; \text{ pour } i = 1 \dots Nx ; j = 1 \dots Ny \text{ et } k = (j - 1) * Nx + i$$

Nous utilisons une fonction qui permet d'interpoler, linéairement, entre les points de données $(x, y, x_f \text{ et } y_f)$. Cette fonction retourne les valeurs interpolées aux points correspondants (x_f, y_f) des éléments de la matrice V_g données par les abscisses (x, y) . Les valeurs de l'interpolation sont stockées dans une matrice de dimension (Nx_f, Ny_f) que nous notons Z_i . Voici l'entête de cette fonction:

$$Z_i \longleftarrow Interp(x, y, V_g, x_f, y_f)$$

Finalement, nous transformons cette matrice en un vecteur colonne de dimension $Nx_f * Ny_f$ pour initialiser la condition initiale sur la grille de temps fine et utiliser (2.1.9) pour calculer une solution approchée locale, U_f .

$$u_f0(k) = Z_i(i, j) ; \text{ pour } i = 1 \dots Nx_f ; j = 1 \dots Ny_f \text{ et } k = (j - 1) * Nx_f + i$$

Puis nous calculons une approximation de la solution continue sur la grille locale de pas $dx, dy < h$ et $\delta t < \Delta t$. Nous utilisons le schéma d'Euler Explicite comme dans (2.1.8) sur cette grille fine pour trouver U_f qui est supposée être plus précise que u_g .

Avant de projeter U_f sur la grille temporelle grossière, nous commençons, d'abord, par la projeter sur la grille spatiale grossière. Pour cela, nous définissons une fonction qui permet de trouver les noeuds grossiers à partir de la grille d'espace fine

$$\text{IndEsGro} \leftarrow (j-1) * Nx_f + i ; i = 1....Nx_f ; j = 1....Ny_f \text{ par pas de } \tau$$

Et donc,

$$u_f \leftarrow U_f(\text{IndEsGro}, n) ; n = 1....Nt_f$$

Pour la projection de u_f sur la grille de temps grossière, nous utilisons un interpolateur linéaire Ru comme nous l'avons défini précédemment(voir pages 23-24).

Estimation de l'erreur " Defect correction "

L'itération \mathcal{LDC} est basée sur le calcul de l'erreur "defect correction" et comment cette erreur va être utiliser pour améliorer la solution approchée sur la grille grossière(en temps et en espace) en utilisant celle calculée sur la grille fine. Nous avons expliquer cette étape de l'algorithme plus en détail(voir page 24) . Par contre, il y a une modification concernant le calcul de "Defect correction ". En effet, le calcul de "d" devient dans le cas 2D

$$\underline{d}^n := \begin{cases} \underline{Ru}^{n+1} - (Id + \Delta t M) \underline{Ru}^n - \Delta t \underline{b}^n - \Delta t \underline{F}^n & \text{pour } n = 1....Nt - 1 \\ 0 & \text{pour } n = Nt \end{cases}$$

Une fois le calcul de "d" est fait sur chaque sous domaine, nous rassemblons ces erreurs, en les stockant dans une matrice P de dimension(Nx*Ny, Nt), afin de les ramener sur la grille grossière globale. L'estimateur P est rajouté au second membre de l'équation (2.1.1) de la façon suivante:

$$\frac{\partial u_g}{\partial t} = -(a_x \frac{\partial u_g}{\partial x} + a_y \frac{\partial u_g}{\partial y}) + \nu (\frac{\partial^2 u_g}{\partial x^2} + \frac{\partial^2 u_g}{\partial y^2}) + f + P$$

A l'itération 1, nous avons obtenu une nouvelle solution approchée sur la grille grossière globale en utilisant celle calculée sur la grille locale. Cette nouvelle approximation de la solution continue est supposée être plus précise que celle obtenue à l'itération 0. La nouvelle solution grossière peut être utilisée pour actualiser la condition initiale en Indice 1 du problème fin pour chaque sous domaine, qui à son tour sera utiliser pour corriger l'approximation grossière de u et ainsi de suite jusqu'à convergence de l'algorithme \mathcal{LDC} .

2.3 Application Numérique

Afin de valider la méthode \mathcal{LDC} , nous avons commencé par des exemples basiques tels que,

$$u(t, x, y) = t, xy...$$

en utilisant Euler Explicite sur la grille grossière. Pour chacun de ces cas tests, nous avons obtenu la convergence de la solution approchée, sur la grille grossière, vers la solution analytique au bout d'une seule itération.

Dans cette section, nous présentons les résultats d'un exemple numérique en 2D. Nous choisissons $Ex = Ey = 1$, $T = 1$ et nous résolvons l'équation de convection-diffusion suivante:

$$\frac{\partial u}{\partial t} + \nabla u = \Delta u + f \quad (2.3.1)$$

La condition initiale, les conditions de Dirichlet sur la frontière(sur les bords 1, 2, 3 et 4) et le terme source f sont choisis de telle sorte que la solution analytique exacte du problème est:

$$u(t, x, y) = 3 - \tanh(25(x - t) + 5(y - 1))$$

Le problème est résolu par \mathcal{LDC} avec différentes valeurs de h , h_f , Δt et δt . la discrétisation d'espace est effectuée en utilisant différences finis 2 globalement et localement. Le schéma d'Euler Explicite est utilisé pour la discrétisation du domaine de temps à la fois sur la grille grossière et fine.

A titre de comparaison, nous résolvons aussi le problème (2.3.1) en utilisant une grille globale unique et uniforme avec un pas d'espace $h_{unif} = h_f$ et un pas de temps $\delta t_{unif} = \delta t$. A l'instant final, nous mesurons l'erreur maximale ξ_∞ de l'approximation numérique à l'égard de la solution exacte comme dans le cas 1D et la norme \mathcal{L}^2 définie par:

$$er2 := \sum_{k=1}^{Nx*Ny} \frac{|u_{gk} - u_k|^2}{|u_k|^2} \quad ; \quad \text{Norme } \mathcal{L}^2 := \sqrt{\frac{er2}{Nx * Ny}}$$

Le tableau suivant montre que \mathcal{LDC} est plus précise que le solveur de la grille uniforme. \mathcal{LDC} effectue un calcul moins coûteux que le solveur de la grille uniforme puisque la solution approchée fine corrige la solution grossière.

Table 2.1: Comparaison entre \mathcal{LDC} et solveur de grille uniforme

Pas du temps		Pas d'espace		ξ_∞		Norme \mathcal{L}^2	
Δt	δt	h	h_f	\mathcal{LDC}	Grille Uniforme	\mathcal{LDC}	Grille Uniforme
0.0625	0.03125	0.5	h	0	0.0563	$3.48 e^{-1}$	1.7
	0.007		$\frac{h}{3}$	$6.5 e^{-4}$	0.6907	$3.46 e^{-1}$	$8.98 e^{-1}$
0.03125	0.0007	0.5	$\frac{h}{3}$	0.0535	0.1656	$2.248461 e^{-1}$	1.543
	$2.5 e^{-4}$		$\frac{h}{5}$	0.0523	0.1715	$2.244585 e^{-1}$	$8.35 e^{-1}$

Ces valeurs ont été pris pour un nombre de sous domaine fixé à $Ndom = 5$.

Sur la figure (2.1), nous traçons la solution analytique ainsi que la solution approchée calculée par la méthode \mathcal{LDC} à l'instant final en $t_{Nt} = 1$, en fonction de pas d'espace

c.à.d en fonction de x et y que nous les comparons à la solution approchée calculée par le solveur de grille fine, qui a été tracée en fonction de pas d'espace fin (x_f et y_f) au temps $t_{Nt_f} = 1$. Toutes ces solutions ont été calculées par le schéma d'Euler Explicite en temps.

Nous avons tracé ces graphes pour les valeurs suivantes de:

- Pas de temps grossier et fin égaux à 0.0625 et 0.007 respectivement,
- Pas d'espace grossier $h = 0.5$ et fin égal à $\frac{h}{3}$,
- $N_{dom} = 5$

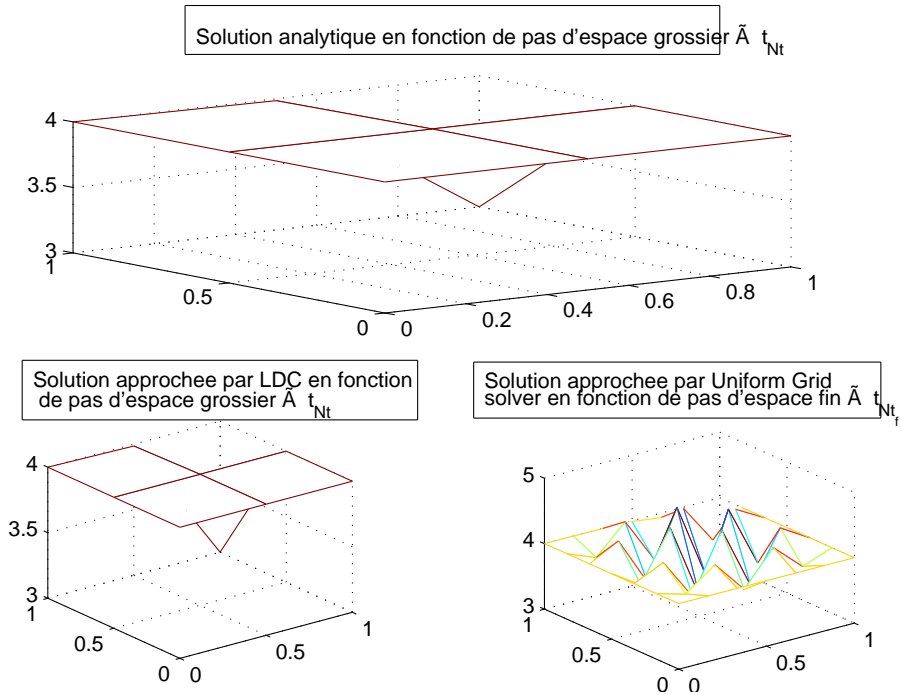


Figure 2.1: Solution exacte, solution approchée calculée par \mathcal{LDC} et celle calculée par le solveur de grille uniforme pour $\Delta t = 0.0625$, $\delta t = 0.007$, $h = 0.5$ et $\tau = 3$.

Nous remarquons sur cette figure que la solution approchée calculée par la méthode \mathcal{LDC} est la plus proche de la solution exacte que la solution calculée sur la grille uniforme.

Sur la figure suivante, nous vous montrons que \mathcal{LDC} est plus précise que solveur de la grille uniforme. Nous traçons les solutions approchées calculées par \mathcal{LDC} et sur la grille uniforme pour $\Delta t = 0.03125$, $\delta t = 0.0007$, $h = 0.5$ et $\tau = 3$.

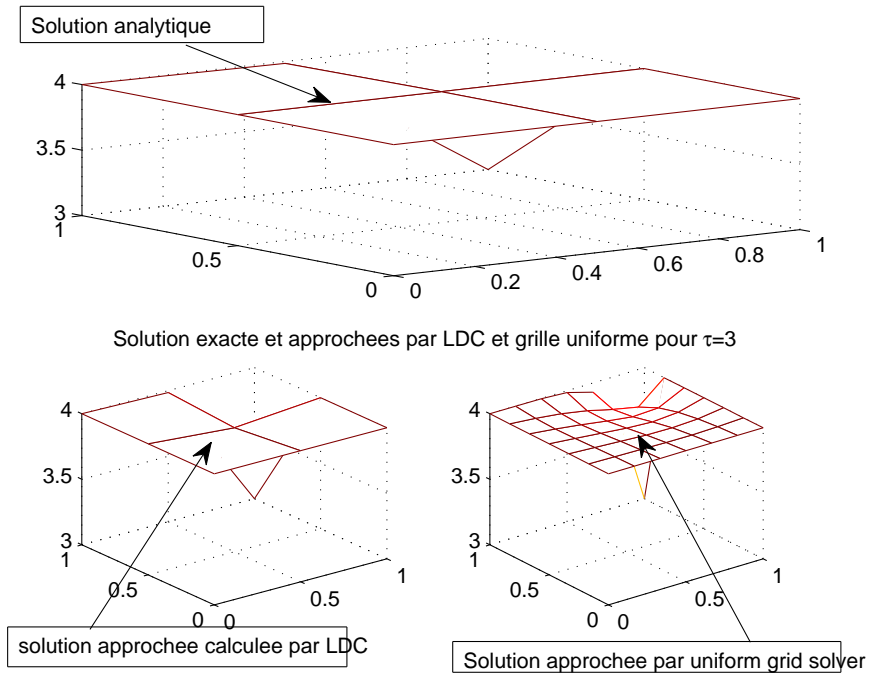


Figure 2.2: Solution exacte, solution approchée calculée par \mathcal{LDC} et celle calculée par le solveur de grille uniforme pour $\Delta t = 0.03125$, $\delta t = 0.0007$, $h = 0.5$ et $\tau = 3$.

2.4 Conclusion

Ce stage effectué au sein de l'ONERA m'a permis d'appréhender mon futur métier d'ingénieur calcul scientifique et d'avoir une idée sur la la vie professionnelle ainsi que de m'imprégner de l'organisation d'un service calcul en entreprise.

Le calcul haute performance occupe ici une place importante dans la phase de simulation. C'est un élément essentiel à la compétitivité des entreprises et un outil indispensable pour les grands challenges scientifiques. Ainsi, elle permet de concevoir plus rapidement des produits mieux adaptés aux besoins des clients mais aussi d'analyser et de comprendre des phénomènes complexes.

Dans cet article, nous avons présenté la méthode de \mathcal{LDC} pour le problème de convection-diffusion 1D et 2D. En général, \mathcal{LDC} est un processus itératif approprié pour résoudre, efficacement, les problèmes caractérisés à chaque pas temps par une région de grande activité qui couvre une petite partie du domaine physique. Nous avons adapté cette méthode, qui a été utilisée pour discrétiser le domaine en espace, pour étudier le problème de convection-diffusion en temps en se basant de la méthode du schéma en temps "pararéel" pour avoir une idée sur la condition initiale sur chaque sous domaine.

D'abord, le problème est résolu sur une grille grossière en temps, l'approximation calculée sur la grille globale définit une condition initiale artificielle pour le problème local fin sur chaque sous domaine, où une solution plus précise peut être calculée en utilisant un pas de temps et d'espace très petit. la solution locale fine est alors utilisée pour améliorer la première approximation grossière sur la grille globale par le "Defect Correction". La nouvelle solution grossière peut à son tour définir une nouvelle condition initiale pour chaque sous domaine pour le problème de la grille fine(en temps et en espace). Le processus peut être répéter jusqu'à convergence, qui est en général très rapide.

Les résultats des expériences numériques montrent que \mathcal{LDC} peut atteindre une précision meilleure que le solveur de grille uniforme dont le pas de temps et d'espace coïncident avec ceux utilisés pour le problème local de l'algorithme \mathcal{LDC} . En même temps, \mathcal{LDC} garantit un moindre coût de calcul que le solveur grille uniforme.

Lors de ce stage, j'ai eu l'opportunité de réaliser une quinzaine d'études et des tests très variés pour trouver le critère d'arrêt le plus efficace pour assurer la convergence .

Ces études m'ont permis d'appréhender un certain nombre de difficultés comme les problèmes de convergence des schémas (instabilité dans le cas de schéma de Runge Kutta 4) ainsi que de déterminer le domaine physique pour lequel \mathcal{LDC} est optimale ...

- [1] Wolfgang Hackbusch
Local Defect Correction method and domain decomposition techniques. In K. Böhm and H. J. Stetter, editors, Defect Correction Methods. Theory and Applications, Computing, suppl. 5(1984), pages 89-113. Springer, Wien New York
- [2] P.J.J. Ferket et A.A. Reusken
- A finite difference discretization method on composite grids. Computing, 56:343-369,1996
- Further analysis of the local defect correction method. Computing, 56:117-139,1996
- [3] J. U. Wappler
Die lokale Defektkorrekturmethode zur adaptiven Diskretisierung elliptischer Differentialgleichungen mit finiten Elementen. PhD thesis, Christian- Albrechts- Universität, Kiel, 1999, In German.
- [4] M. J. H. Anthonissen
Local defect correction technique: analysis and application to combustion. PhD thesis, Eindhoven University of technology, Eindhoven, 2001
- [5] M. J. H. Anthonissen, B. van't Hof et A.A. Reusken
A finite volume scheme for solving elliptic boundary value problems on composite grids. Computing, 61:285-305,1998.
- [6] V. Nefedov et R.M.M. Mattheij
Local defect correction with different grid types. Numerical Methods for Partial Differential Equations, 18:454-468,2002
- [7] R. Minero, M. J. H. Anthonissen, et R. M. M. Mattheij.
A Local Defect Correction technique for time-dependent problems.
Numerical Methods for Partial Differential Equations, 2006. Early View.
- [8] R. A. Trompet et J. G. Verwer
- Analysis of the implicit Euler local uniform grid refinement method.
SIAM journal of Scientific Computing, 14:259-278, 1993.
- Runge Kutta methods and local uniform grid refinement.
Mathematics of Computation, 60:591-616, 1993.
- [9] Lions, J.-L., Maday, Y. and Turinici, G.
Résolution d'EDP par un schéma en temps "pararéel".
CRAS Sér I Math. v332 i7. 661-668.

- [10] M. J. H. Anthonissen, et R. M. M. Mattheij et J. H. M. ten Thije Boonkamp.
Convergence analysis of a Local Deffect Correction method for diffusion equation.
Numer Math 95 (2003), 401-425.

APPENDIX A

RÉSOLUTION D'EDP PAR UN SCHÉMA EN TEMPS "PARARÉEL"

On considère une équation d'évolution aux dérivées partielles (EDP)

$$\frac{\partial u}{\partial t} + Au = f \text{ dans un intervalle de temps } [0, T]$$

avec des conditions initiales $u(t=0) = u_0$ et des conditions aux limites qu'il est inutile de préciser ici.

Nous proposons dans cette note un schéma en temps compatible avec une résolution en parallèle et avec une précision souhaitée. Pour cela, on choisit un entier N représentant le nombre de pas de temps et on note $\Delta t = \frac{T}{N}$ le pas de temps et $T^n = n\Delta t$ pour $n = 0, \dots, N$ et on résout, dans l'intervalle $[T^n, T^{n+1}]$ l'équation

$$\frac{\partial u_n}{\partial t} + Au_n = f_n, \quad f_n = f|_{[T^n, T^{n+1}]}$$

Pour commencer, on expose l'idée sur l'exemple simple d'une équation différentielle linéaire

$$\begin{cases} \frac{dy}{dt}(t) &= -ay(t) \quad \text{sur } [0, T] \\ y(t=0) &= y_0 \end{cases}$$

On considère le schéma d'Euler implicite

$$\begin{cases} \frac{Y^{n+1} - Y^n}{\Delta t} + aY^{n+1} = 0 \\ Y^0 = y_0 \end{cases}$$

puis on utilise les valeurs précédemment calculées pour résoudre de façon exacte, sur chaque intervalle de temps $[T^n, T^{n+1}]$

$$\begin{cases} \frac{dy^n}{dt}(t) &= -ay^n(t) \quad \text{sur } [T^n, T^{n+1}] \\ y^n(t=T^n) &= Y^n \end{cases}$$

On propose maintenant une procédure itérative pour améliorer la précision de ce schéma.

On pose donc $Y_1^n = Y^n$ et, défini sur $[T^n, T^{n+1}]$, $y_1^n(t) = y^n(t)$. Puis, supposant connus Y_k^n et $y_k^n(t)$ sur $]T^n, T^{n+1}]$

(i) on introduit les sauts $S_k^n = y_k^{n-1}(T^n) - Y_k^n$

(ii) puis on propage les sauts

$$\begin{cases} \frac{\delta_k^{n+1} - \delta_k^n}{\Delta t} + a\delta_k^{n+1} = \frac{S_k^n}{\Delta t} \\ \delta_k^0 = 0 \end{cases}$$

(iii) ensuite on pose $Y_{k+1}^n = y_k^{n-1}(T^n) + \delta_k^n$ et on résout de façon exacte, et en parallèle

$$\begin{cases} \frac{dy_{k+1}^n}{dt}(t) = -ay_{k+1}^n(t) & \text{sur } [T^n, T^{n+1}] \\ y_{k+1}^n(t = T^n) = Y_{k+1}^n \end{cases}$$