

CERFACS
CFD - COMBUSTION



Sous la direction de Bénédicte CUENOT

SIMULATION NUMÉRIQUE DIRECTE DE LA COMBUSTION TURBULENTE AVEC NTMIX3D

Rapport de stage de fin de 2^{ème} année

Jean DASSÉ

Août 2006

Ref :

Département Hydraulique et Mécanique des fluides
ENSEEIH



Résumé

NTMIX3D est un puissant code utilisé pour simuler numériquement des écoulements réactifs en résolvant explicitement les équations de Navier-Stokes pour des géométries simples. Ce code est un outil performant de compréhension des mécanismes complexes dont le développement est directement limité par la puissance de calcul. Du fait des performances actuelles des calculateurs massivement parallèles, la simulation numérique directe est reconsidérée et NTMIX3D doit être mis à jour pour utiliser l'environnement logiciel nécessaire au traitement des résultats, la gestion du code et à la compilation des sources sur les nouvelles architectures.

Mots-clés : SND , NTMIX3D, CVS, Makefiles, Parallélisation

Abstract

NTMIX3D is a powerful code used to simulate reactive flows numerically by solving directly the Navier-Stokes equations for simple geometries. This code is a comprehension tool of the complex mechanisms in fluids mechanics whose development is directly limited by the computing power. Because of the current performances of massively parallel calculators, the direct numerical simulation is reconsidered and NTMIX3D must be updated to use the software environment necessary for the treatment of the results, management of the code and compilation of the sources on new architectures.

Keywords : DNS, NTMIX3D, CVS, Makefiles, parallelization

Contents

Introduction	1
Le CERFACS	1
Les objectifs du stage	3
La combustion turbulente et ses applications	3
1 Modélisation de la combustion turbulente	5
1.1 Les équations de base	5
1.2 Formulation adimensionnelle	7
1.3 Limite de stabilité	8
2 Implémentation numérique de NTMIX-3D sur calculateurs parallèles	9
2.1 Résolution des équations par la méthode des différences finies	9
2.1.1 Intégration temporelle : méthode de Runge Kutta d'ordre 3	9
2.1.2 Dérivation spatiale : schéma compact ou de Padé d'ordre 6	10
2.1.3 Résolution des équations par l'algorithme de Thomas	10
2.2 Conditions limites	10
2.3 Parallélisation du code par décomposition du domaine	11
2.4 Test de parallélisation de NTMIX-3D sur les calculateurs du CERFACS	12
3 Environnement de NTMIX-3D	15
3.1 Utilisation de CVS pour la gestion de NTMIX-3D	15
3.2 Compilation et installation du code	16
3.2.1 Présentation des compilateurs	16
3.2.2 Les quatres étapes de la compilation	16
3.2.3 Les Makefiles	17
Conclusions et perspectives	19
Annexes	20
Bibliographie	21

INTRODUCTION

1. Le CERFACS

Le CERFACS (*Centre Européen de Recherche et de Formation Avancée au Calcul Scientifique*) a été créé en 1987. C'est un centre de recherche de renommée internationale dans le domaine du calcul scientifique à hautes performances. Il héberge une centaine de personnes parmi lesquelles des chercheurs permanents, des ingénieurs, doctorants et post-doctorants de toutes nationalités, ainsi que de nombreux stagiaires et visiteurs.

Société civile entre EADS, SNECMA, EDF, Météo France et le CNES, le CERFACS travaille à la résolution par simulations numériques des problèmes scientifiques nécessitant le recours à des moyens de calcul à hautes performances. L'activité du laboratoire se divise en cinq branches :

- climatologie et environnement (prévision saisonnière, changements climatiques) ;
- PAE (Impact du trafic aérien sur l'environnement) ;
- électromagnétisme ;
- algorithmes numérique ;
- mécanique des fluides CFD (*Computational Fluid Dynamics*).

Cette dernière branche est divisée en deux équipes : *aérodynamique* et *combustion*. L'équipe *combustion* dans laquelle j'ai effectué mon stage oriente ses travaux autour de deux types de simulations :

- la LES (*Large Eddy Simulation* ou Simulation aux Grandes Échelles) qui est devenue la méthode de référence pour modéliser la turbulence. Elle est utilisée dans des géométries complexes pour étudier le mélange, les instabilités de combustion, la combustion turbulente gazeuse ou diphasique (gaz-liquide)... Le code de simulation aux grandes échelles AVBP développé au CERFACS comme un code industriel est un outil inégalé techniquement avec peu de concurrents au niveau international.
- la DNS (*Direct Numerical Simulation* ou Simulation Numérique Directe) plus précisément destinée à la recherche car se focalisant sur des géométries simples de faibles dimensions est un outil très puissant pour l'étude en profondeur des mécanismes. La DNS permet la résolution des équations de Navier Stokes en calculant explicitement toutes les échelles de turbulence. Les effets de combustion ainsi que toutes les instationnarités sont capturés par la simulation. Les codes utilisées au CERFACS dans le cas de la combustion turbulente sont NTMIX3D pour la configuration 3D avec chimie simple et NTMIX CHEMKIN pour des configurations 2D avec chimie complexe. La simulation numérique directe initialisée dans les années 1970 revient d'actualité dû à l'accroissement rapide des performances des calculateurs actuels permettant aujourd'hui des temps de calcul raisonnables¹. La DNS permet par exemple d'étudier les interactions flamme/turbulence avec chimie complexe, d'élaborer les lois de parois dans des configurations géométriques simples et de valider les modèles de turbulence ou de combustion utilisés dans les codes LES ou RANS dont elle est le complément indispensable.

¹Des calculs d'une durée d'un an peuvent être encore entrepris par exemple pour la modélisation de la turbulence homogène isotrope THI

Les avantages et les inconvénients des méthode LES et DNS comparées à la méthode RANS très présente au niveau industriel sont résumés dans le tableau 1.

La Figure 1 présente une simulation à l'aide de ces trois méthodes.

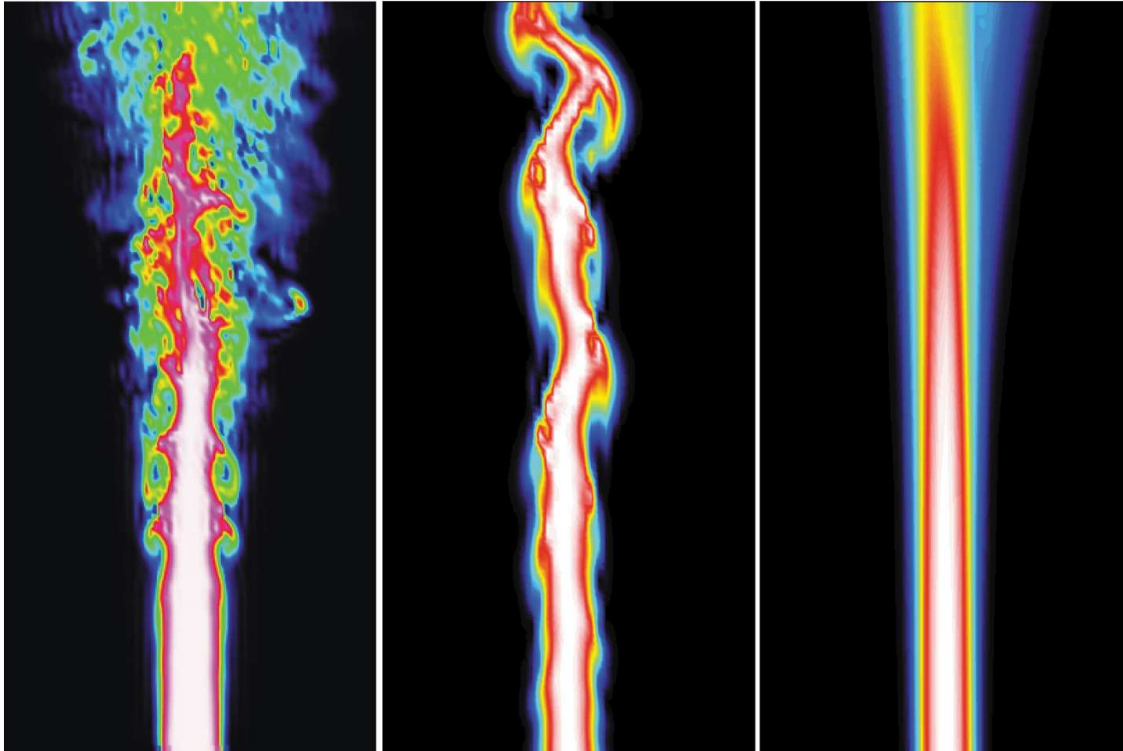


Figure 1: Exemples de simulations numériques : à gauche DNS (toutes les échelles sont résolues), au milieu LES (larges échelles résolues et faibles échelles modélisées), à droite RANS (toutes les échelles sont modélisées)

Approche	Avantages	Inconvénients
RANS	<ul style="list-style-type: none"> - maillage « grossier » - simplifications géométriques (écoulements 2D, symétrie,...) - coûts numériques « réduits » 	<ul style="list-style-type: none"> - seulement les champs moyens - modèles requis
LES	<ul style="list-style-type: none"> - comportements instationnaires - impact réduit de la modélisation (par rapport au RANS) 	<ul style="list-style-type: none"> - modèles requis - simulation 3D requise - nécessite des codes précis - coût numérique
DNS	<ul style="list-style-type: none"> - pas de modèles nécessaire pour l'itération turbulence/combustion - outil d'étude des modèles 	<ul style="list-style-type: none"> - coût numérique prohibitif (maillage fin, codes précis) - limité à des problèmes académiques

Table 1: Avantages et inconvénients des méthodes RANS, LES et DNS pour les simulations numériques de la combustion turbulente

2. Les objectifs du stage

La priorité de ce stage a été portée sur l'outil informatique pour le calcul à hautes performances et son environnement appliqué au code de simulation numérique directe NTMIX3D. Plus précisément, les principaux objectifs de mon stage ont été les suivants :

- étude bibliographique sur la combustion turbulente et la simulation numérique directe [Poinsot and Veynante, 2005; Kuo, 1986; Borghi and Destriau, 1995] ;
- Familiarisation avec le système d'exploitation et les commandes Unix/Linux [Proux, 2005] ;
- initiation à CVS pour la gestion et le développement multi-utilisateurs de NTMIX3D [Purdy, 2004] ;
- prise en main du code de simulation numérique directe NTMIX3D et synthèse du guide utilisateur [Cuenot and Bédât, 1997; Helie et al., 2002] ;
- développement de Makefiles et de scripts Unix pour l'installation et la compilation du code ;
- initiation aux différents compilateurs (Intel, Portland, GCC) pour la portabilité de NTMIX3D sur les différents calculateurs présents au CERFACS et sur PC personnel Linux ;
- mise à jour des outils de post et de prétraitement (appelés « moulinettes ») et développement d'une moulinette de conversion du format binaire au format des logiciels de traitement 1D *Igor* et *Xmgrace* ;
- initiation au parallélisme d'un code et au MPI (*Message Passing Interface*) [Chergui et al., 2006] permettant la communication entre processeur et tests de performances de parallélisation ;
- cas test de validation du code sur des configurations simples (écoulement de Poiseuille 2D et flamme de prémélange laminaire 1D).
- apprentissage du langage Latex pour l'écriture du rapport [Oetiker et al., 2001] ;
- mise à jour du guide utilisateur de NTMIX3D ;

3. La combustion turbulente et ses applications

Il s'agit ici de présenter sommairement la combustion turbulente et quelques applications afin de caractériser le domaine d'étude de NTMIX3D.

La combustion est définie comme une réaction chimique globalement et généralement fortement exothermique, autonome, susceptible de s'accélérer brutalement et d'être accompagnée d'émission de rayonnements se situant globalement dans le visible, le proche UV et le proche IR et peut conduire à la formation d'une flamme. La flamme peut être soit prémélangée soit de diffusion. Les réactifs nommés généralement constituant la flamme de prémélange sont mélangés avant d'arriver dans la chambre de combustion contrairement à la flamme de diffusion où les réactifs sont introduits séparément. Les réactifs sont nommés généralement « fuel » (H_2 , CH_4 , essences C_nH_m ...) et « oxydant » (O_2 , air ...).

Ces deux types de flamme peuvent se former dans des écoulements laminaires ou turbulents ce qui permet de diviser l'étude de la combustion en quatre domaines selon le type de flamme (prémélangée ou de diffusion) et le niveau de turbulence (laminaire ou turbulent). Par exemple une bougie est une flamme de diffusion où la cire (carburant) remontant par capillarité à travers la mèche est séparée de l'arrivée d'air arrivant autour. L'écoulement autour de la flamme peut être supposé laminaire. Dans le cas d'une cuisinière à gaz, le carburant et l'oxydant sont prémélangés dans la « bouteille » et l'écoulement est aussi supposé laminaire autour de la flamme.

Dans le cas d'un écoulement turbulent réactif (combustion turbulente), le phénomène se complique par l'interaction de la combustion avec la turbulence. Le front de flamme, droit ou courbe dans le cas laminaire interagit avec les tourbillons turbulents d'où un plissement de ce front. Par l'augmentation de surface dû à ce plissement, les réactions chimiques peuvent augmenter ou être complètement inhibées ce qui peut conduire à une extinction de flamme. La vitesse et l'épaisseur d'une flamme turbulente sont bien supérieures au cas

de la flamme laminaire. Par exemple pour une flamme hydrocarbone/air à la pression atmosphérique, le front de flamme laminaire se déplace à une vitesse de l'ordre de 20 à 100 cm/s et l'épaisseur est de l'ordre de 0.1 mm alors que pour la flamme turbulente, la vitesse de flamme est de l'ordre de 10 m/s et l'épaisseur de quelques millimètres à quelques mètres.

Réciproquement, quand une flamme interagit avec un écoulement turbulent, celui-ci est modifié par la combustion à cause de l'accélération provoquée par le front de flamme ainsi que par le changement de viscosité associé à la hausse de température. Ce mécanisme seul peut générer de la turbulence ou laminariser un écoulement turbulent.

La combustion turbulente est le coeur de nombreux systèmes tels que les moteurs fusées, les réacteurs d'avion, les fours industriels, les moteurs à combustion interne . . . Un des plus simples foyers utilisant la combustion turbulente est le statoréacteur où du combustible est brûlé dans un écoulement permanent d'air à forte vitesse (environ 50 m/s). Le combustible gazeux ou liquide est souvent injecté suffisamment en amont de la zone de combustion pour que l'on puisse considérer que la combustion se fait dans un milieu gazeux où combustible et oxydant sont mélangés de façon pratiquement parfaite.

Dans un foyer de statoréacteur, l'écoulement est turbulent dû à la forte vitesse de l'écoulement entrant et à cause des gradients de vitesses aux parois ou d'obstacles, par exemple les injecteurs. La figure 2 présente le schéma de fonctionnement du statoréacteur.

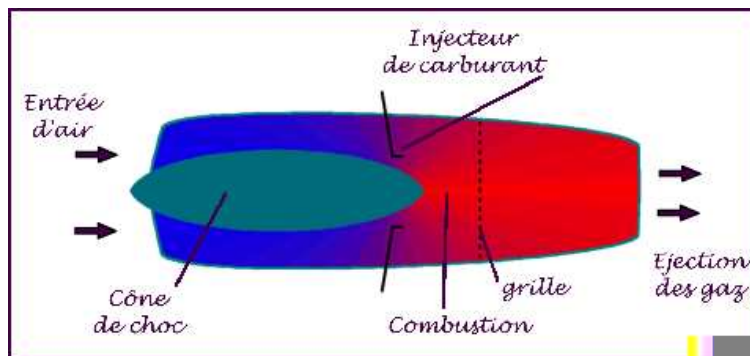


Figure 2: Schéma de principe du statoréacteur

Les moteurs à combustion interne et allumage commandé utilisent aussi la combustion turbulente par propagation d'une flamme de prémélange dans le milieu gazeux à partir d'une étincelle déclenchée au niveau de la bougie.

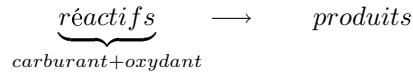
Dans les brûleurs à gaz industriels, un jet de combustible est souvent injecté au centre d'un écoulement d'air périphérique. Après allumage, la combustion se développe alors sous la forme d'une flamme de diffusion turbulente. Là encore, la turbulence joue un rôle très bénéfique pour mélanger les gaz en présence, oxydant et fuel mais aussi gaz brûlés. La turbulence est due en grande partie aux forts gradients de vitesse qui existent entre les vitesses d'arrivée de l'air et du gaz. [Borghini and Destriau, 1995]

Chapter 1

Modélisation de la combustion turbulente

1.1 Les équations de base

Le système d'équations pour un écoulement réactif dans le cas d'une réaction chimique unique irréversible :



est :

- Équation de continuité

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \quad (1.1)$$

- Équation de fraction de mélange

$$\frac{\partial (\rho Y_k)}{\partial t} + \frac{\partial}{\partial x_i} (\rho Y_k u_i) = \frac{\partial}{\partial x_j} \left(\rho \mathcal{D}_k \frac{\partial Y_k}{\partial x_j} \right) + \dot{\omega}_k \quad (1.2)$$

- Équation de mouvement (avec $i = 1, 2, 3$)

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_i u_j) + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} \quad (1.3)$$

- Équation de l'énergie totale

$$\frac{\partial \rho e_t}{\partial t} + \frac{\partial}{\partial x_i} [(\rho e_t + p) u_i] = \frac{\partial}{\partial x_j} (u_i \cdot \tau_{ij}) - \frac{\partial q_i}{\partial x_i} + Q \dot{\omega} \quad (1.4)$$

où :

$$e_t = \frac{1}{2} \cdot \sum_{k=1}^3 u_k^2 + e \quad (1.5)$$

est l'énergie totale somme de l'énergie cinétique et de l'énergie interne. L'énergie interne est reliée à la pression par $e = \int C_p dT - \frac{p}{\rho}$ [Poinsot and Veynante, 2005]. Avec l'hypothèse des gaz parfaits $P = \rho RT$, C_p peut s'écrire $C_p = \frac{\gamma r}{\gamma - 1}$ donc l'énergie interne s'écrit :

$$e = \frac{P}{\rho(1 - \gamma)} \quad (1.6)$$

τ_{ij} est le tenseur visqueux des contraintes visqueuses :

$$\tau_{ij} = \mu \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) \quad (1.7)$$

Le terme $\mathcal{D}_k \frac{\partial Y_k}{\partial x_i}$ de l'équation (1.2) découle de la loi de Fick qui s'écrit avec $V_{k,i}$ la vitesse de diffusion de l'espèce k suivant la direction i :

$$-V_{k,i} Y_k = \mathcal{D}_k \frac{\partial Y_k}{\partial x_i} \quad (1.8)$$

La chaleur de réaction massique Q dans l'équation (1.4) est modélisée par l'intermédiaire de $\alpha = (T_2 - T_1)/T_2$ appelé facteur de température [Williams, 1985] par :

$$Q = \frac{\alpha}{1 - \alpha} C_p T_1 \quad (1.9)$$

Pour une flamme de prémélange dans le cas d'une réaction unique et en partant de la loi d'Arrhenius le taux de réaction¹ du combustible (*Fuel reaction rate* $\dot{\omega}_F$) peut être exprimé comme suit [Poinot and Veynante, 2005]:

$$\dot{\omega}_F = B \rho Y_F \exp\left(-\frac{T_{a1}}{T}\right) = B \rho Y_F \exp\left(-\frac{\beta}{\alpha}\right) \exp\left(-\frac{\beta(1-\Theta)}{1-\alpha(1-\Theta)}\right) \quad (1.10)$$

donc :

$$\dot{\omega}_F = B \rho Y_F^1 (1 - \Theta) \exp\left(-\frac{\beta}{\alpha}\right) \exp\left(-\frac{\beta(1-\Theta)}{1-\alpha(1-\Theta)}\right) \quad (1.11)$$

en utilisant la température réduite $\Theta = (T - T_1)/(T - T_2) = 1 - Y$ avec $Y = Y_F/Y_F^1$. Les notations T_1 , T_2 et T_{a1} sont respectivement la température des gaz frais, la température de flamme adiabatique² pour un nombre de Lewis unité et la température d'activation³ relative à la réaction unique (indice 1). Les coefficients B et β sont appelés respectivement le facteur réduit préexponentiel et la température d'activation réduite $\beta = \alpha T_{a1}/T_2$.

Les variables indépendantes sont la masse volumique ρ , le mouvement ρu_i dans la direction x_i , la densité d'énergie totale e_t et la fraction massique Y_F . La densité de flux de chaleur q_i selon la direction x_i est donnée par la loi de Fourier :

$$q_i = -\lambda \frac{\partial T}{\partial x_i} \quad (1.12)$$

où la conductivité thermique λ et le coefficient de diffusion massique \mathcal{D}_k de l'espèce k sont obtenus directement à partir des nombres sans dimension de Prandtl Pr et de Schmidt Sc_k :

$$\lambda = \mu \frac{C_p}{Pr} \quad \text{et} \quad \mathcal{D}_k = \frac{\mu}{\rho Sc_k} \quad (1.13)$$

Les nombres de Prandtl Pr et de Schmidt Sc_k sont supposés constant. Par conséquent le nombre de Lewis Le_k est aussi constant par la relation :

$$Le_k = \frac{Sc_k}{Pr} = \frac{\lambda}{\rho C_p \mathcal{D}_k} \quad (1.14)$$

La viscosité cinématique est fonction de la température :

$$\mu = \mu_1 \left(\frac{T}{T_1} \right)^b \quad (1.15)$$

où b est une constante (pour de l'air, sa valeur est d'environ 0.76).

¹Taux de disparition d'une espèce chimique [Kuo, 1986]

²voir à ce sujet un calcul clair de cette température dans [Borghini and Destriau, 1995]

³Dans certains ouvrages la température d'activation T_a est remplacée par l'énergie d'activation $E_a = RT_a$

1.2 Formulation adimensionnelle

NTMIX3D résoud les équations de Navier Stokes dans leur forme adimensionnelle plus pratique pour l'implémentation numérique et l'interprétation des résultats. Les lois de similitude qui lient les variables adimensionnelles et dimensionnelles sont définies par :

$$u_i^+ \cdot a_{ref} = u_i \quad t^+ \cdot L/a_{ref} = t \quad x_i^+ \cdot L = x_i \quad T^+ \cdot T_{ref} = T \quad (1.16)$$

$$\rho^+ \cdot \rho_{ref} = \rho \quad p^+ \cdot p_{ref} = p \quad e_t^+ \cdot a_{ref}^2 \cdot \rho_{ref} = e_t \quad \nu^+ \cdot \nu_{ref} = \nu \quad (1.17)$$

$$\mu^+ \cdot \mu_{ref} = \mu \quad C_p^+ \cdot C_{p_{ref}} = C_p \quad (1.18)$$

où a est la vitesse du son.

Les valeurs de référence sont définis par rapport à l'état à « l'infini ». La seule condition imposée aux variables à « l'infini » est que l'équation d'état des gaz parfait doit être vérifiée. Pour une combustion de prémélange, l'état de référence est généralement l'état des gaz frais. En utilisant ces restrictions, il est possible de définir les valeurs de référence à « l'infini » par :

$$a_{ref} = a_\infty \quad T_{ref} = (\gamma - 1) \cdot T_\infty \quad \rho_{ref} = \rho_\infty \quad p_{ref} = \rho_\infty \cdot a_\infty^2 = \gamma \cdot p_\infty \quad (1.19)$$

$$\nu_{ref} = \nu_\infty \quad C_{p_{ref}} = C_{p_\infty} \quad (1.20)$$

Les nombres adimensionnels sont :

$$Pr = \left(\frac{C_p \cdot \mu}{\lambda} \right)_{ref} \quad (1.21)$$

$$Sc_k = \left(\frac{\nu}{D_k} \right)_{ref} \quad (1.22)$$

$$Le_k = \left(\frac{\lambda}{\rho C_p D_k} \right)_{ref} = \frac{D_t}{D_k} = \frac{Sc_k}{Pr} \quad (1.23)$$

$$Re = \left(\frac{aL}{\nu} \right)_{ref} \quad (1.24)$$

avec D_t la diffusivité thermique.

Ces nombres sont déterminés en fonction de l'état de référence ou peuvent être donnés directement si cet état de référence n'est pas défini. De plus, [Rutland, 1989] introduit le nombre de Damköler Da pour caractériser la flamme où s_L représente la vitesse de flamme laminaire :

$$Da = \frac{D_k}{s_L^2} B \exp\left(-\frac{\beta}{\alpha}\right) \quad (1.25)$$

En utilisant ces variables et nombres adimensionnels, le système d'équations (1.1) à (1.4) devient :

$$\frac{\partial \rho^+}{\partial t^+} + \frac{\partial}{\partial x_i^+} (\rho^+ u_i^+) = 0 \quad (1.26)$$

$$\begin{aligned} \frac{\partial \rho e_t^+}{\partial t^+} + \frac{\partial}{\partial x_i^+} [(\rho e_t^+ + p^+) u_i^+] &= \frac{1}{Re} \cdot \frac{\partial}{\partial x_j^+} (u_i^+ \cdot \tau_{ij}^+) - \frac{1}{Re \cdot Pr} \cdot C_p^+ \frac{\partial q_i^+}{\partial x_i^+} \\ &+ \rho^+ Y \alpha' T_\infty^+ \cdot Re Pr Da \cdot s_L^{+2} \cdot \exp\left(\frac{-\beta(1-\Theta)}{1-\alpha(1-\Theta)}\right) \end{aligned} \quad (1.27)$$

$$\frac{\partial \rho u_i^+}{\partial t^+} + \frac{\partial}{\partial x_j^+} (\rho u_i^+ u_j^+) + \frac{\partial p^+}{\partial x_i^+} = \frac{1}{Re} \cdot \frac{\partial \tau_{ij}^+}{\partial x_j^+} \quad (1.28)$$

$$\frac{\partial (\rho Y)}{\partial t^+} + \frac{\partial}{\partial x_i^+} (\rho^+ Y u_i^+) = \frac{1}{Re \cdot Sc} \cdot \frac{\partial}{\partial x_i^+} \left(\mu^+ \frac{\partial Y}{\partial x_i^+} \right) - \rho^+ Y Re Pr Da \cdot s_L^{+2} \cdot \exp\left(\frac{-\beta(1-\Theta)}{1-\alpha(1-\Theta)}\right) \quad (1.29)$$

avec

$$e_t^+ = \frac{1}{2} \cdot \sum_{k=1}^3 u_k^{+2} + \frac{p^+}{\rho^+ (\gamma - 1)} \quad (1.30)$$

$$p^+ = \rho^+ T^+ C_p^+ \frac{\gamma - 1}{\gamma} \quad (1.31)$$

$$\mu^+ = \mu_1^+ \cdot \left[\frac{T^+}{T_1^+} \cdot (\gamma - 1) \right]^b \quad (1.32)$$

$$\tau_{ij}^+ = \mu^+ \cdot \left(\frac{\partial u_i^+}{\partial x_j^+} + \frac{\partial u_j^+}{\partial x_i^+} - \frac{2}{3} \delta_{ij} \frac{\partial u_k^+}{\partial x_k^+} \right) \quad (1.33)$$

$$\alpha' = \frac{\alpha}{1 - \alpha} \quad (1.34)$$

Dans le cas d'une réaction unique irréversible, NTMIX3D introduit les variables $Damla_{fuel}$ pour l'équation (1.29) et $Damla$ pour l'équation (1.27) :

$$Damla_{fuel} = s_L^2 Pr Re Da \quad (1.35)$$

$$Damla = s_L^2 Pr Re Da \alpha' T_\infty^+ \quad (1.36)$$

1.3 Limite de stabilité

L'intervalle de temps maximal est donné par la limite de stabilité exprimée en général par le nombre de Courant-Friedrichs-Lewy CFL pour la partie convective et par le nombre de Fourier Fo pour la partie diffusive. Ces nombres peuvent être exprimés sous forme adimensionnelle (en configuration monodimensionnelle) :

$$CFL = \max \left[\left(|u^+| + \sqrt{C_p^+ \cdot (\gamma - 1) \cdot T^+} \right) \cdot \frac{\Delta t^+}{\Delta x^+} \right] \quad (1.37)$$

$$Fo = \max \left[\frac{\mu^+}{\rho^+} \cdot \frac{\Delta t^+}{\Delta x^{+2}} \cdot \frac{1}{Re Pr} \right] \quad (1.38)$$

Les valeurs critiques doivent être déterminées par l'utilisateur. En général, les valeurs $CFL < 0.6$ et $Fo < 0.1$ sont suffisant pour assurer la stabilité numérique. En plus de ces critères, la présence de réactions chimiques dans la phase gazeuse impose des limitations supplémentaires. Ceci est dû au fait que seule une raisonnable quantité d'espèces peut être consommée ou produite pendant un intervalle de temps sans perturber l'équilibre entre réaction et diffusion. La quantité pouvant être produite ou consommée est spécifiée dans le code par le nombre « chimique » et est déterminée de manière empirique. Un mauvais choix de ce nombre peut induire des instabilités ou annuler des réactions chimiques. L'intervalle de temps est déterminé par le plus sévère de tous ce critères.

Chapter 2

Implémentation numérique de NTMIX-3D sur calculateurs parallèles

2.1 Résolution des équations par la méthode des différences finies

Le système d'équations (1.26) à (1.29) est résolu dans NTMIX3D par la méthode des différences finies d'ordre élevé. La méthode des différences finies consiste à remplacer les dérivées partielles aux points du maillage par des développements de Taylor. Plutôt que de chercher $\Theta(x, y, z, t)$ pour Θ une variable primitive (T, ρ, u, v, w, e_t et Y) évoluant de façon continue, on préfère chercher $\Theta(x_i, y_j, z_k, t_n)$ évoluant discrètement. Deux types de dérivée partielle sont présentes dans les équations. Les dérivations temporelles et spatiales utilisent chacun une méthode de résolution différente.

2.1.1 Intégration temporelle : méthode de Runge Kutta d'ordre 3

Soit l'équation :

$$y(t + \Delta t) = y(t) + \Delta t \cdot \hat{f}(t, y(t)) \quad (2.1)$$

représentant l'avancement d'une fonction à intégrer y de t à $t + \Delta t$. $\hat{f}(t, y(t))$ représente la pente de la courbe sur l'intervalle Δt . L'idée principale de la méthode de Runge Kutta est que la valeur suivante $y(t + \Delta t)$ est approximée par la somme de la valeur actuelle $y(t)$ et du produit de la taille de l'intervalle Δt par la pente estimée par la dérivée temporelle de $y(t)$.

La méthode d'Euler ou de Runge kutta d'ordre 1 consiste à prendre :

$$\hat{f}(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) \quad (2.2)$$

Cependant, dans ce cas là, cette pente déterminée en début d'intervalle génère une erreur en puissance de Δt découlant de la décomposition en série de Taylor. Une méthode de Runge Kutta d'ordre 3, notée RK3, consiste à calculer plus précisément la valeur de la pente $\hat{f}(t, y(t))$ en podérant celle-ci en différents points de l'intervalle :

$$\begin{aligned} \hat{f}(t, y(t)) &= \frac{1}{4}(K_1) + \frac{3}{4}(K_3) \\ K_1 &= \frac{\partial f}{\partial t}(t, y(t)) \\ K_2 &= \frac{\partial f}{\partial t}\left(t + \frac{\Delta t}{3}, y + \frac{\Delta t}{3}K_1\right) \\ K_3 &= \frac{\partial f}{\partial t}\left(t + \frac{2\Delta t}{3}, y + \frac{2\Delta t}{3}K_2\right) \end{aligned} \quad (2.3)$$

On a donc :

- K_1 est la pente au début de l'intervalle ;

- K_2 est la pente au tiers de l'intervalle en utilisant K_1 pour calculer y ;
- K_3 est la pente aux 2/3 de l'intervalle, en utilisant la pente K_2 pour calculer y .

Dans la moyenne des trois pentes, un poids plus grand est donné à la pente calculée aux 2/3 de l'intervalle. La méthode RK3 est une méthode d'ordre 3, ce qui signifie que l'erreur commise à chaque étape est de l'ordre de Δt^4 , alors que l'erreur totale accumulé est de l'ordre de Δt^3 [Hirsch, 1990]. Ces formules sont aussi valables pour des fonctions à valeurs vectorielles.

2.1.2 Dérivation spatiale : schéma compact ou de Padé d'ordre 6

La dérivation spatiale est calculée dans NTMIX3D par le schéma de Padé (Schéma compact) d'ordre 6 dans toutes les directions qui a la propriété d'être moins dispersif mais aussi précis que les méthodes spectrales [Hilka et al., 1994]. Il s'écrit dans le cas d'une dérivation d'ordre 1 [Hirsch, 1990] :

$$3 \left(\frac{\partial u}{\partial x} \right)_{i+1} + 9 \left(\frac{\partial u}{\partial x} \right)_i + 3 \left(\frac{\partial u}{\partial x} \right)_{i-1} = \frac{1}{h} \cdot \left(\frac{1}{4} (u_{i+2} - u_{i-2}) + 7 (u_{i+1} - u_{i-1}) \right) \quad (2.4)$$

2.1.3 Résolution des équations par l'algorithme de Thomas

Les équations (1.26) à (1.29) intégrées spatialement par la relation (2.4) et temporellement par la relation (2.1) pour les différents points du domaine donne lieu à une méthode explicite en temps et implicite en espace. Le nouveau système d'équations discrétisées requiert la solution d'un système linéaire tridiagonal avec de multiples termes de droite (RHS). Cette solution est déterminée par une factorisation LU d'un système tridiagonal, mieux connues en CFD sous le nom de l'algorithme de Thomas qui est une version simplifiée de l'élimination de Gauss.

Un système tridiagonal général peut être écrit :

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i \dots \quad (2.5)$$

où $a_1 = 0$ et $c_n = 0$. Sous forme matricielle, ce système peut s'écrire :

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \cdot & \\ & & \cdot & \cdot & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{bmatrix} \dots \quad (2.6)$$

Pour un tel système, la solution peut être obtenue en $O(n)$ opérations alors que la méthode de l'élimination Gaussienne en nécessite $O(n^3)$. Cette méthode s'appuie sur deux phases. La première consiste à éliminer la diagonale des a_i de la matrice (2.6) servant d'exemple par une combinaison linéaire puis par une substitution arrière (*backward substitution*), la solution est déterminée.

2.2 Conditions limites

Dans NTMIX3D, les conditions limites sont divisées en trois familles :

- Conditions limites géométriques ne modifiant pas les équations du phénomène qui englobent les conditions limites périodiques et symétriques.
- Conditions limites physiques modifiant les équations aux frontières comme les conditions limites non-réfléchissantes. La prise en compte de ces conditions limites se fait par une méthode des caractéristiques appelée NSCBC pour *Navier-Stokes Characteristic Boundary Conditions*. Au départ appliquée aux équations d'Euler, cette méthode permet de spécifier les conditions limites pour des systèmes hyperboliques en analysant les différentes ondes traversant les frontières [Poinsot and Lele, 1992; Poinsot and Veynante, 2005]. Cette condition limite réfléchissante permet d'absorber la majorité des ondes acoustiques et convectives dans le cas optimal d'un plan d'onde parallèle à la frontière.

- La troisième famille de conditions limites est celle qui nécessite un domaine de calcul propre qui peut être différent du domaine principal de calcul. Les surfaces réactives appartiennent à cette famille.

Le tableau 2.1 énumère les différentes conditions limites utilisées dans NTMIX3D :

Condition limite	Commentaires	Référence
symétrique		-2
pas de condition	utilisé dans le cas périodique	-1
non réfléchissante	méthode NSCBC	0
vitesse et température imposées	valeurs à « l'infini » de v_i et T , par défaut $Y = 1$	1
vitesse et densité imposées	valeurs à « l'infini » de v_i et ρ , par défaut $Y = 1$	2
mur non glissant isotherme	température imposée mais vitesse du mur et Y non imposées	3
mur glissant adiabatique	vitesse du mur normal à la surface de la CL et flux de chaleur imposé	4
mur non glissant adiabatique	idem CL 3 mais température non imposée	5
pression imposée	frontière réfléchissante par méthode NSCBC	6

Table 2.1: Différentes conditions limites et leur nom dans NTMIX3D

2.3 Parallélisation du code par décomposition du domaine

Le but de cette méthode est de résoudre localement le problème global [Hilka et al., 1994]. Le domaine global est divisé en sous domaine (voir figure 2.1) chacun traité indépendamment par un processeur mais communique avec les autres par l'intermédiaire de ses conditions limites. Cette méthode découle de la méthode de Schwarz et des multi-domaines [Stoessel and Baum, 1994].

Deux méthodes ont été développées. La première consiste à résoudre chaque sous-système avec les conditions limites de Dirichlet. Classiquement, la condition limite de Dirichlet est mise à jour à partir des solutions des autres sous-domaines et il y a itération jusqu'à convergence de la solution. La relation initiale de dérivation spatiale (2.4) est modifiée au niveau de la frontière de la manière suivante :

$$9 \left(\frac{\partial u}{\partial x} \right)_i + 3 \left(\frac{\partial u}{\partial t} \right)_{i-1} = \frac{1}{h} \cdot \left(\frac{1}{4} (u_{i+2} - u_{i-2}) + 7 (u_{i+1} - u_{i-1}) \right) - 3 \overline{\left(\frac{\partial u}{\partial x} \right)_{i+1}} \quad C' \text{ est ce type de condition qui est utilisé dans (2)}$$

où le dernier terme est une dérivation spatiale au niveau de la frontière approximé par un schéma centré du 6^{ème} ordre qui n'altère pas la précision du schéma global et qui nécessite une seule itération pour la convergence vers la solution globale. En utilisant cette méthode, il est possible de transformer une méthode en espace globalement implicite en une méthode localement implicite avec couplage explicite entre les sous-domaines. Cette méthode de parallélisation est de ce fait efficace et triviale. Cependant, comme les différents sous-domaines doivent être recouverts, cette méthode ne fonctionne bien qu'avec un grain de parallélisation moyen ou grossier¹. Sinon, le coût de l'évaluation des conditions limites et le traitement des points supplémentaires venant du recouvrement devient trop important par rapport au temps mis par un processeur pour calculer le domaine global. Prenons un exemple simple : un domaine de 128 points selon les 3 directions. Par défaut NTMIX3D nécessite un recouvrement de 3 points selon les 3 directions. Donc le domaine passe de 128^3 à $(128+2 \times 3)^3$ (voir figure 2.1) soit une augmentation du nombre de points de 14,7%.

Une deuxième méthode existe qui consiste cette fois à ne plus utiliser une condition de Dirichlet sur les frontières des sous-domaines mais des conditions limites non réfléchissantes issues des conditions NSCBC. Ce schéma non centré issu des conditions NSCBC est moins précis et nécessite des recouvrement plus grands

¹Pour la décomposition en sous-domaines, le grain de la parallélisation définit la taille de chaque domaine en fonction du domaine global. Un grain grossier signifie des sous-domaines de grande taille par rapport au domaine global

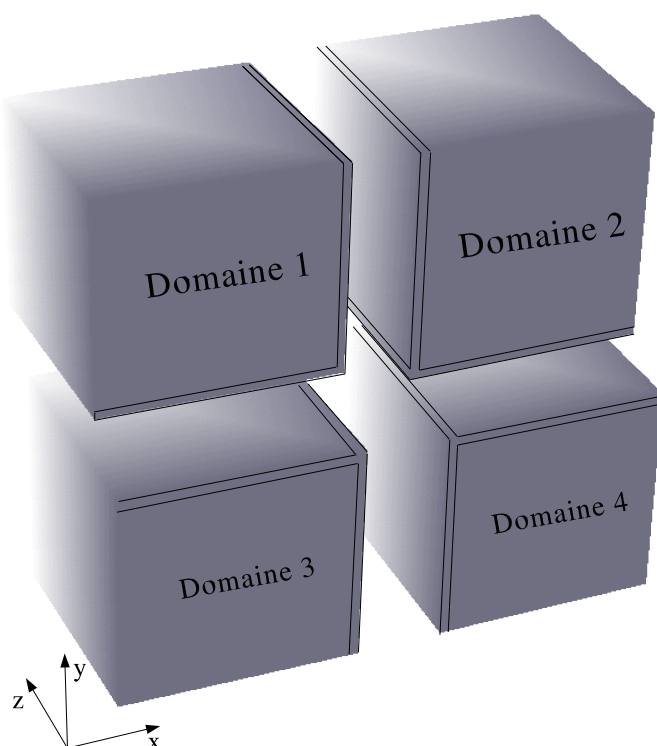


Figure 2.1: Décomposition du domaine global en quatre sous-domaines et mise en évidence du recouvrement nécessaire pour la communication entre sous-domaines.

mais permet de réduire l'impact du couplage entre sous-domaines. NTMIX3D utilise ce deuxième type de conditions limites entre sous-domaines.

2.4 Test de parallélisation de NTMIX-3D sur les calculateurs du CERFACS

Pour tester les performances de la parallélisation, ici par décomposition de domaine, il est nécessaire d'introduire la notion d'accélération plus connu dans ce domaine sous le terme anglais de *speedup* :

$$speedup = \frac{\text{Temps mis par 1 processeur pour calculer le domaine global}}{\text{Temps mis par un processeur pour calculer un des } n \text{ sous-domaines}} < n \quad (2.8)$$

où n est le nombre de sous-domaines chacun attribué à un processeur. Idéalement, ce *speedup* doit être inférieur au nombre de processeurs nécessaire au calcul du domaine complet. Considérant à priori les faibles performances de NTMIX3D dans des configurations faiblement parallélisées (faible nombre de processeurs), Les définitions suivantes sont nécessaires pour l'étude des performances :

$$speedup2 = \frac{\text{Temps mis par 1 processeur pour calculer la } \mathbf{moitié} \text{ du domaine}}{\text{Temps mis par un processeur pour calculer un des } n \text{ sous-domaines}} \times 2 < n \quad (2.9)$$

$$speedup4 = \frac{\text{Temps mis par 1 processeur pour calculer le } \mathbf{quart} \text{ du domaine}}{\text{Temps mis par un processeur pour calculer un des } n \text{ sous-domaines}} \times 4 < n \quad (2.10)$$

Des mesures des différents speedups ont été effectués sur les deux calculateurs du CERFACS : le Compaq alphaserver Imhotep et le CRAY XD1 Kali. Le sun grid Engine composé de tous les PC Linux du CERFACS en parallèles a aussi été testé. Les résultats sont repertoriés aux tableaux 2.2 à 2.4 et aux figures 2.2 à 2.3.

Nombre de procs	Temps CPU	Speedup	Speedup2	Speedup4
1	5985			
2	2163	2.8		
4	646	9.3	6.7	
6	377	15.9	11.5	6.8
8	330	18	13	7.8
12	215	27.8	20.1	12
16	166	36	26.1	15.6

Table 2.2: Mesure des speedups d'Imhotep pour 128^3 points

Nombre de procs	Temps CPU par proc	Speedup	Speedup2	Speedup4
1	2369			
2	748	3.2		
4	286	8.3	5.2	
6	198	11.2	7.6	5.8
8	151	15.7	9.9	7.6
12	105	22.6	14.2	10.9
16	83	28.5	18	13.8

Table 2.3: Mesure des speedups de Kali pour 128^3 points

Nombre de procs	Temps CPU par proc	Speedup1	Speedup4
1	2416		
4	526	4.6	
6	391	6.2	5.4
8	286	8.4	7.4

Table 2.4: Mesure des speedups de Linux Sun Grid Engine pour 128^3 points

Pour les trois serveurs, les valeurs calculées des *speedup* et *speedup2* ne sont pas satisfaisantes car ne devant pas dépasser la valeur du nombre de processeurs ou de sous-domaine. En effet il n'est pas normal que par exemple un domaine global soit calculé plus rapidement avec 4 processeurs qu'avec un seul (voir Fig.2.3 à droite). Ce qui est demandé à la parallélisation par décomposition de domaines est qu'un quart du domaine global soit calculé au maximum quatre fois plus rapidement que le domaine global entier par un processeur.

Seules les valeurs de *speedup4* sont satisfaisantes. La définition (2.10) servira donc à quantifier les performances de parallélisation par décomposition de domaines avec NTMIX3D.

Les valeurs excessives des *speedup* et *speedup2* sont induites par un temps excessif de calcul des sous-domaines dans des décompositions inférieures à 4 sous-domaines. En effet lors de ces décompositions en faible nombre de sous-domaines, le temps de sortie des fichiers résultats alors de grandes tailles vers la mémoire physique du système s'opère pendant un temps supérieur que pour des domaines plus petits. Par décomposition successive du domaine global, les sous-domaines se divisent et l'accès mémoire diminue. De plus, il serait bien de vérifier si le temps CPU donné dans le fichier résultat de NTMIX3D ne considère que les parties parallélisables en évitant les parties incompressibles d'initialisation.

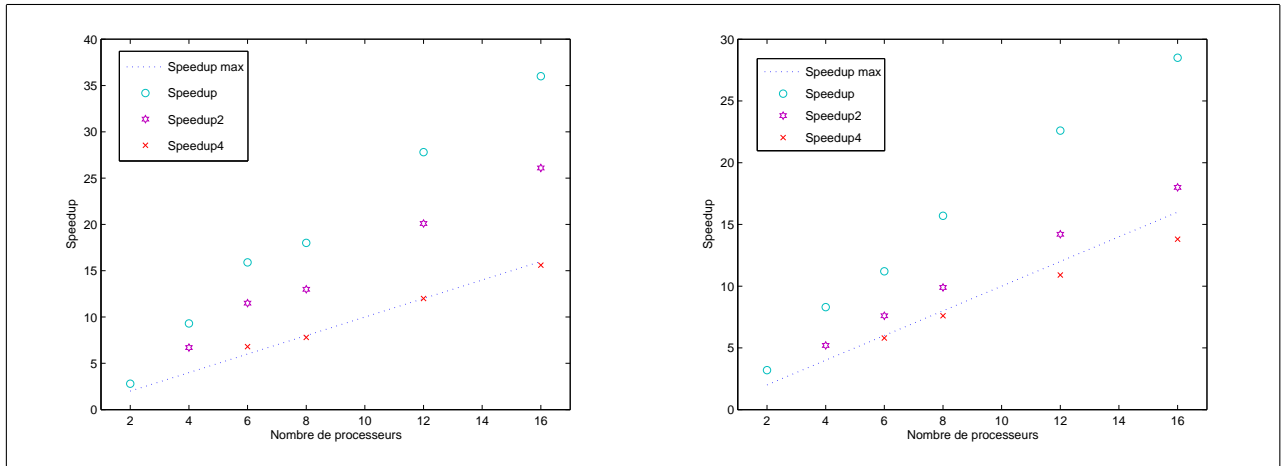


Figure 2.2: *Speedups* des calculateurs Imhotep (à gauche) et kali (à droite)

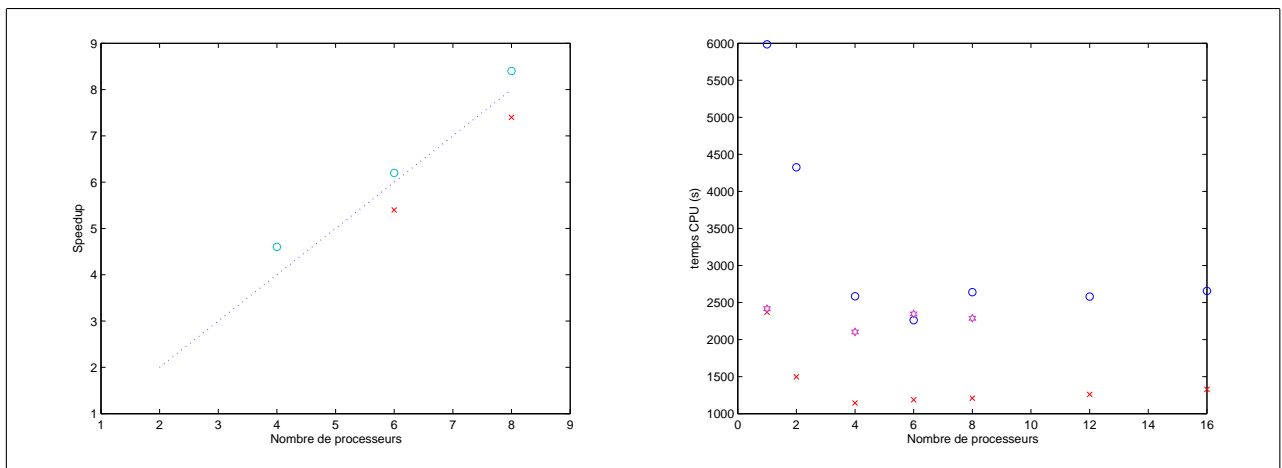


Figure 2.3: *Speedup* des PC Linux en parallèles Sun Grid Engine (à gauche) et comparatif du temps CPU des trois calculateurs (à droite)

Chapter 3

Environnement de NTMIX-3D

*I*l est présenté dans ce chapitre les outils extérieurs au code de simulation numérique directe NTMIX3D pour sa gestion, sa compilation automatisée et au traitement de ses données. Ces outils peuvent être appliqués à tout projet informatique.

3.1 Utilisation de CVS pour la gestion de NTMIX-3D

Le système de gestion de versions de fichiers CVS (*Concurrent Version System*) permet de conserver la trace et l'historique des modifications successives effectuées sur un projet. CVS aide à gérer le développement d'un projet effectué en parallèle par plusieurs utilisateurs en identifiant les zones de conflit pour lesquelles un arbitrage humain est requis. Un projet sous CVS est organisé sous forme de modules, représentant l'arborescence des fichiers. CVS peut fonctionner en accès direct sur un système de fichiers réparti comme NFS¹ ou en mode client/serveur autorisant les développements répartis sur plusieurs sites comme dans le cas de NTMIX3D (développement entre le CERFACS et l'IFP).

Cette section est une présentation de CVS du côté utilisateur, c'est à dire que CVS est déjà installé sur le serveur et un répertoire contenant le projet en cours a été initialisé. Pour des renseignements sur l'administration de CVS et pour des approfondissements côté utilisateur, le lecteur pourra consulter [Purdy, 2004].

Pour télécharger en interne une version de NTMIX3D, il faut premièrement localiser le dépôt CVS nommé le « CVSROOT » qui est le répertoire racine où le code est archivé avec toutes ses modifications successives. Dans le cas de NTMIX3D, en interne au cerfacs, il faudra taper :

```
#> setenv CVSROOT machine:/chemin/du/répertoire/NTMIX_V2.X/CVS
```

En externe, il est possible de déclarer la variable CVSROOT par la commande `ssh` ou `rsh` pour des utilisateurs ayant un accès total au serveur. Dans le cas d'un utilisateur ou d'un développeur anonyme², un serveur CVS public peut être mis en place [Purdy, 2004].

Ensuite, le répertoire de travail qui va accueillir le projet doit être créé. Les fichiers de ce répertoire pourront être modifiés sans pour autant modifier le répertoire du dépôt CVSROOT. Pour télécharger la dernière version du projet, la commande « `cvs checkout` » permettra de télécharger dans le répertoire de travail une copie du projet contenu dans le dépôt. Pour NTMIX3D, le téléchargement de la dernière version du code se fait simplement par :

```
#> cvs checkout NTMIX
```

Une fois les fichiers du répertoire de travail modifiés, le dépôt pourra être mis à jour en ne conservant que les différences entre chaque version (gain de place disque) par la commande « `cvs commit` » suivi du

¹Système de fichiers accessible depuis toutes les machines d'un réseau permettant de centraliser l'accès mémoire pour les utilisateurs

²Comme c'est le cas pour les logiciels *open-source*

nom de la modification apportée :

```
#> cvs commit -m "<modifications apportées>" NTMIX
```

La partie « commentaires » des sources de NTMIX3D intègre le mot-clé « \$log » qui va être substitué par le nom de l'étiquette pour conserver un historique des modifications à l'intérieur du code.

Toute modification d'édition d'un fichier sera considérée automatiquement par CVS pour une mise à jour du dépôt. Cependant toutes les opérations de manipulation de fichiers ou répertoires devront être explicitement indiquées à CVS. L'effacement d'un fichier sera indiqué à CVS par un « `cvs remove` » suivi du fichier à effacer. L'ajout d'un fichier ou répertoire se fera par un « `cvs add` ». La modification du nom est une combinaison d'un « `cvs remove` » suivi d'un « `cvs add` ».

La commande « `cvs diff` » est aussi très utile pour revenir à une version antérieure par comparaison des modifications entre deux versions.

Dans le cas d'un développement en parallèle, la commande « `cvs update` » permettra de mettre à jour son répertoire de travail en fusionnant toutes les modifications du dépôt effectuées par tous les développeurs du projet.

3.2 Compilation et installation du code

Après avoir téléchargé le code et effectué des modifications éventuelles, il est nécessaire de compiler les sources du code pour avoir un exécutable.

3.2.1 Présentation des compilateurs

La compilation est le travail réalisé par un compilateur qui consiste à transformer un code source lisible par un humain appelé code de haut niveau (langages Fortran, C, C++, Java, Python, Ada...) en un fichier binaire exécutable par une machine. Le principal compilateur gratuit existant sur le marché est le compilateur GCC pour *GNU Compiler Collection* fourni en standard sur le système d'exploitation Unix. En faisant référence aux compilateurs de chaque langage, on parle de GCC³ pour le compilateur du langage C, G++ celui du langage C++ et Gfortran pour le langage fortran qui remplacent les anciens compilateurs CC, f77, f90. Il existe aussi des compilateurs payants beaucoup plus complets utilisés dans le cas de NTMIX3D pour le calcul parallèle. Ces compilateurs sont de la marque Intel⁴, Portland, Compaq ...

3.2.2 Les quatre étapes de la compilation

Lors de la compilation d'un code, quatre opérations sont effectuées : Le passage au préprocesseur (*preprocessing*), la compilation en langage assembleur (*compiling*), la conversion du langage assembleur en code machine (*assembling*) et l'édition des liens (*linking*).

Dans cette partie, il est pris comme exemple le compilateur « `gcc` » sur un fichier écrit en langage C.

Passage au préprocesseur

Dans le code, certaines parties sont introduites par le caractère # qui permet d'indiquer au préprocesseur une directive qui va subir une substitution textuelle. Ces directives sont :

- Les macros « `#define` » qui permettent l'utilisation d'une abréviation autant de fois qu'on le souhaite par la suite. Celle-ci s'écrit en majuscule et désigne la plupart du temps une constante (par exemple la constante PI sera définie par une macro).
- Les « `#include` » appellent des fichiers pouvant être partagés par plusieurs fichiers sources nommés les fichiers en-tête d'extension « `.inc` » ou « `.h` ».

³Même appellation entre la collection de compilateur et le compilateur C

⁴Seulement la toute dernière version est payante

- Les conditionnelles « `#if`, `#ifdef`, `#ifndef`, `#elif`, `#else` » qui permettent lors du passage du préprocesseur de cibler les parties du code à conserver et celles à effacer selon l'architecture sur lequel l'exécutable sera lancé fonction des constantes définies dans des macros.
- Les commentaires vont être effacés lors du preprocessing.

Compilation en langage assembleur

Le langage assembleur ou langage d'assemblage, dit aussi assembleur ou abrégé ASM, est le langage de programmation lisible pour un humain le plus proche du langage machine utilisé par le microprocesseur. Il y a correspondance directe entre code assembleur et langage machine contrairement au langage de haut niveau. Cette phase d'assemblage prendra en considération l'architecture du système pour s'en affranchir dans les prochaines étapes de compilation.

Conversion du langage assembleur en langage machine

Le code assembleur (qui est lisible) est transformé en *code machine* (langage binaire). Le fichier obtenu est un « fichier objet » de suffixe « `.o` ».

Édition de liens

L'édition de liens a pour but de réunir le ou les fichiers objets et les routines contenues dans les bibliothèques, pour produire le programme complet exécutable. Elle va résoudre les références entre les différents modules de code (appels de sous-programmes, références à des variables définies dans d'autres modules...), réunir les fichiers objets et les fonctions des bibliothèques (comme par exemple `write` ou `read`).

3.2.3 Les Makefiles

Comme on vient de le voir précédemment, deux fichiers essentiels peuvent découler d'un fichier source : les « fichiers objets » et les « exécutables ». lorsque le fichier source est modifié, les fichiers objets et les exécutables deviennent obsolètes. Lorsqu'un fichier est composé de plusieurs fichiers sources⁵, il est nécessaire de pouvoir gérer les dépendances entre ces fichiers de façon à ne recompiler que ce qui est nécessaire. C'est l'utilitaire « `make` » qui permet de gérer les dépendances, elles sont décrites dans un fichier Makefile. La règle de base explicite d'un Makefile est de la forme suivante :

```
cible: dependance
<tab>  commande
        commande..
```

La première ligne indique en général que le fichier cible dépend des fichiers donnés dans la liste de dépendances. La ou les lignes qui suivent décrivent les commandes à effectuer si le fichier cible n'existe pas, ou si un des fichiers de dépendances est plus récent que le fichier cible c'est à dire si un fichier dépendance a été modifié sans que le fichier cible n'ait été recompilé. Ces commandes peuvent être toutes commandes du Shell. Si l'un des fichiers de dépendances est lui-même cible d'une dépendance, alors il est traité auparavant. Enfin, la cible n'est pas obligatoirement un nom de fichier, mais peut être une simple étiquette (nettoyage par exemple).

La figure 3.1 présente le Makefile de NTMIX3D . A la fin de ce Makefile, est introduite une « règle des suffixes » qui est en fait une règle implicite permettant de compiler indifféremment tout fichier source en se servant du suffixe de ces fichiers. Le fichier de dépendance inclus à la fin de Makefile est du type :

```
Objet1.o: sources.f sources.c includes.h includes.inc ...
Objet2.o: sources.f sources.c includes.h includes.inc ...
...
```

où n'apparaît pas de commande. En fait chaque ligne ne concerne que la dépendance à la cible et la compilation s'effectue automatiquement par règles implicites.

⁵NTMIX3D est composé d'une soixantaine de fichiers sources fortran, d'une trentaine d'*includes* et d'un fichier source C servant pour l'allocation dynamique de la mémoire

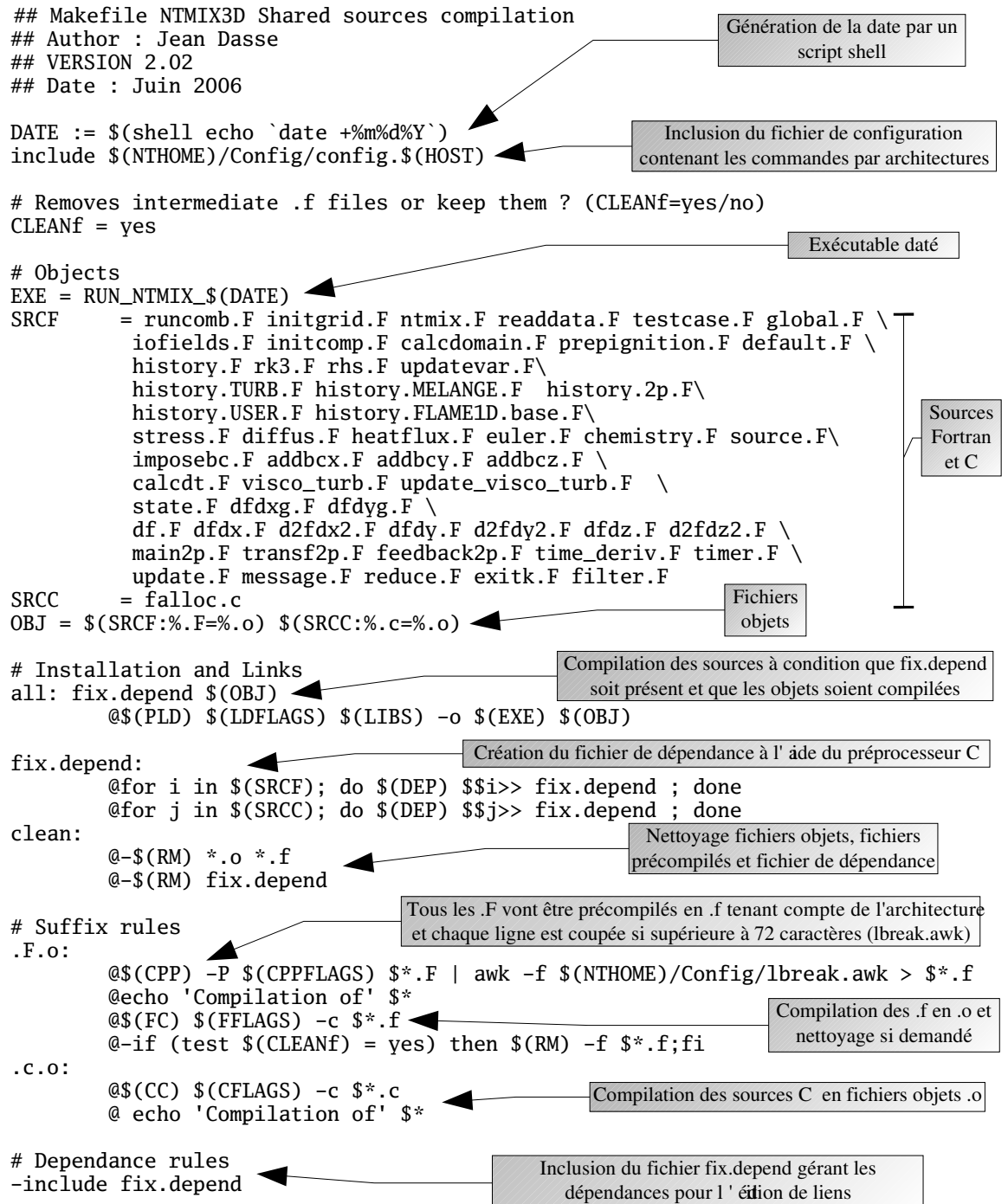


Figure 3.1: Makefile de compilation des sources de NTMIX3D et explication des différents modules

CONCLUSIONS ET PERSPECTIVES

Ce stage passé au CERFACS m'a permis d'aborder le domaine de la simulation numérique directe à l'aide du code NTMIX3D pouvant être appliqué à la combustion turbulente très présente dans les systèmes de propulsion ou dans les fours industriels. Mon travail a consisté au développement informatique du code et de son environnement : gestion de projet avec CVS, compilation et installation à l'aide des Makefiles, performances de calculateurs parallèles. Ces éléments sont à la base de tout code informatique à vocation scientifique et peuvent constituer une bonne introduction au domaine du numérique.

Dans un premier temps, mon travail s'est focalisé sur l'installation et la compilation de NTMIX3D. Les Makefiles ont donc été simplifiés et modifiés pour la prise en compte des nouveaux calculateurs parallèles du CERFACS ayant chacun des architectures différentes. Une configuration par architecture a aussi été remplacé par une configuration par hôte facilitant l'utilisation.

Dans un deuxième temps, une mise à jour des moulinettes a été effectuée pour le post-traitement des résultats donnés par NTMIX3D sur les logiciels Tecplot et Ensignht pour la 2D/3D, Igor et Xmgrace pour la 1D. La moulinette de prétraitement servant à initialiser une flamme 1D a aussi été validée conforme. L'architecture du code a dû être reconsidérée pour que la compilation de ces différentes moulinettes s'effectue à partir du Makefile principal.

Afin de valider ces modifications sur ces nouvelles architectures et pour débiter avec NTMIX3D, des configurations simples ont été testées par l'intermédiaire de cas-test de validation. Deux configurations simples (flamme 1D laminaire, écoulement de poiseuille) ont été testées avec succès et les résultats concordent avec les résultats théoriques pour ces deux configurations. De plus, la parallélisation sur les calculateurs scalaires du CERFACS a été testée pour vérifier la conformité avec les calculs effectués précédemment par [Hilka et al., 1994] sur des machines vectorielles. Les performances de parallélisation se sont avérées convaincantes mais des approfondissements pourraient être faits sur le temps de calcul excessif pour un domaine global divisé en moins de quatre sous-domaines.

Une mise à jour plus profonde de NTMIX3D passerait par la traduction du code en Fortran 90 ce qui éviterait le recours au langage C pour l'allocation dynamique de la mémoire qui peut être source d'erreur d'interaction entre codes sur certaines architectures. Cependant la refonte totale en Fortran 90 n'est pas jugée pour l'instant prioritaire car le temps nécessaire à celle-ci peut s'avérer inutile au vu des performances actuelles satisfaisantes du code. De plus, dans NTMIX3D, toute la partie de précompilation permettant d'adapter le code à l'architecture se base sur des appels d'architectures obsolètes et la mise à jour du script d'installation (Makefile) s'appuie sur ces anciennes architectures. Même si le code est correctement compilé sur ces différentes architectures, il faudrait néanmoins reprendre chacun des sous-programmes pour remplacer les appels des architectures obsolètes par les nouvelles architectures pour ainsi valider les options de compilation. De nouveaux cas-tests pourraient être aussi mis en oeuvre pour des écoulements diphasique ce qui validerait le code dans cette configuration.

Bibliography

- R. Borghi and M. Destriau. *La combustion et les flammes*. Éditions Technip, 1995.
- J. Chergui, I. Dupays, D. Girou, S. Requena, and P. Wautelet. *Message Passing Interface MPI-1 MPI-2*. http://www.idris.fr/data/cours/parallel/mpi/choix_doc.html, Juin 2006.
- B. Cuenot and A. Bédard, Corjon. *NTMIX3D, user's guide*, 1997.
- J. Helie, B. Bedat, B. Cuenot, and R. Paoli. *The NTMIX-3D On-line Users Guide*. <http://www.cerfacs.fr/ntmix/>, 2002.
- M. Hilka, D. Veynante, A. Stoessel, and M. Baum. *Parallel direct numerical simulation of turbulent premixed flames*. ISATA, 1994.
- Hirsch. *Numerical computation of internal and external flows*, volume 1. John Wiley and sons, 1990.
- K. K. Kuo. *Principles of combustion*. John Wiley, 1986.
- T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *Une courte (?) introduction \tilde{A} $\LaTeX 2_{\epsilon}$* . <http://www.laas.fr/matthieu/cours/latex2e/>, Novembre 2001.
- T. Poinso and S. Lele. Boundary conditions for direct simulations of compressible viscous flows. *J. Comp. Physics*, vol.101(1):104–129, 1992.
- T. Poinso and D. Veynante. *Theoretical and Numerical Combustion*. Edwards, 2005.
- Jean-Philippe Proux. *Cours Unix utilisateur avancé*. http://www.idris.fr/data/cours/unix/user/choix_doc.html, Janvier 2005.
- Gregor N. Purdy. *CVS précis et concis*. Seconde édition. O'Reilly, 2004.
- C.J. Rutland. *Effect of strain, vorticity and turbulence on premixed flames*. PhD thesis, Stanford, 1989.
- A. Stoessel and M. Baum. Direct numerical simulation of 2d turbulent combustion using domain decomposition methods. In *SCS High Performance Computing Conference*, La Jolla, 1994. crct CONF.
- F. A. Williams. *Combustion theory*. Addison-Wesley Publishing Co., 1985.