Performance de calcul parallele sur KSR1 pour la resolution numerique des equations d'Euler en regime hypersonique hors equilibre thermo-chimique avec une methode de type volumes-finis

Fadi El Dabaghi, M.C. Druget

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Performances de calcul parallèle sur KSR1 pour la résolution numérique des équations d'Euler en régime hypersonique hors équilibre thermo-chimique, avec une méthode de type volumes-finis

Fadi EL DABAGHI - Marie-Claude DRUGUET

N° 2112
Novembre 1993

PROGRAMME 6

Calcul scientifique,
modélisation et
logiciels numériques

Performances de Calcul Parallèle sur KSR1 pour la Résolution Numérique des Equations d'Euler en Régime Hypersonique Hors Equilibre Thermo-Chimique, avec une méthode de type Volumes-Finis.

Fadi El Dabaghi† Marie-Claude Druguet‡

Résumé

Dans un cadre de calcul parallèle intensif, ce travail concerne la simulation numérique d'un modèle d'écoulements à haute enthalpie. Ce modèle est gouverné par un système de lois de conservation, comprenant les équations d'Euler compressibles et des équations d'évolution de composants chimiques et d'énergies de vibration hors équilibre. La méthode de solution adoptée se base sur une approche de type Volumes Finis, avec schéma explicite du 1er ordre en temps, combinée aux flux de Van-Leer. Malgré une formulation axisymétrique simplifiant notablement le problème 3D initial, la résolution numérique nécessite encore des temps *CPU* importants. Ainsi, et afin de réduire les temps d'exploitation d'un tel solveur, la parallélisation sur la KSR1 est entreprise partiellement sur la partie la plus consommatrice en temps, à savoir l'étape de résolution du déséquilibre chimique; trois cas tests, de tailles croissantes en calcul, ont fait l'objet de cette démarche en utilisant différentes stratégies de parallélisation. L'analyse comparative des résultats correspondants, illustrées par des courbes de performance, montre d'une part les excellents speed-up obtenus et d'autre part indique comment améliorer, sinon au moins garder ces performances lors des applications industrielles de très grande taille.

Mots-Clefs *Axisymétrique, Calcul Parallèle, Déséquilibre Chimique et Vibrationnel, Equations d'Euler, Hypersonique, Flux de Van-Leer, KSR, Volumes Finis, Speed-Up.*

† INRIA - Projet MENUSIN, B.P. 105 Rocquencourt, 78153 Le Chesnay Cedex, France.

‡ IUSTI-URA CNRS 1168, Marseille, France - European Space Agency, France - Dept. of Maths, Univ. of Houston, Texas 77204-3476

Parallel Performances on KSR1 for the Numerical Resolution of Euler Equations in Thermo-Chemical Nonequilibrium Hypersonic Regime, with a Finite Volumes method.

Fadi El Dabaghi† Marie-Claude Druguet‡

Abstract

In a high performance computing frame, this work consists of the numerical simulation of high enthalpy flows. This model is governed by a conservation laws system, including the compressible Euler equations and transport equations convecting vibrational energies and chemical componants of the air in a non-equilibrium state. The method of solutions is based on a finite volume approach with a 1^{st} order time explicit scheme combined to the Van-Leer flux. Despite the axisymetric formulation considered in this study, the numerical resolution still requires large *CPU* times. Thus, and in order to reduce the running time of such a solver, the parallelization on the KSR1 has been undertaken partially on the most important *CPU* time part of the code: the resolution of the chemical nonequilibrium step; three test cases of increasing size are considered in this experience with the use of different parallel tile strategies. The comparative analysis of the corresponding results, illustrated by measured performance curves, exhibits on the one hand the excellent speed-up and indicates on the other hand how to improve or at least how to preserve these performances in large scale industrial applications.

Key-Words *Axisymmetric, Parallel Computing, Thermo-Chemical Nonequilibrium, Euler Equations, Hypersonic, Van-Leer Flux, KSR, Finite Volumes, Speed-Up.*

† INRIA - Projet MENUSIN, B.P. 105 Rocquencourt, 78153 Le Chesnay Cedex, France.

‡ IUSTI-URA CNRS 1168, Marseille, France - European Space Agency, France - Dept. of Maths, Univ. of Houston, Texas 77204-3476

Introduction

Malgré les acquis et les progrès considérables de la dernière décennie, la modélisation physique et numérique des écoulements hypersoniques reste encore un challenge important. En fait, à l'état actuel, nous disposons dans ce créneau de la mécanique des fluides, de modèles physiques assez réalistes et de solveurs numériques relativement robustes, ces deux aspects évoluant très vite et se complémentant sans cesse. En effet, les mesures de plus en plus fiables et nombreuses, relevées lors des récentes missions spatiales ont contribué à mieux cerner l'aspect physique des écoulements hypersoniques. Dans ce qui suit, nous limitons le cadre d'étude à une partie de ce grand défi, à savoir la modélisation physique et numérique des écoulements de la phase de rentrée dans l'atmosphère de la navette spatiale du projet HERMES.

Afin de prédire les caractéristiques dynamiques et physico-chimiques de l'écoulement d'air autour d'un tel engin, dans des conditions de rentrée atmosphérique dites de "haute enthalpie", nous utilisons deux types d'approches complémentaires: la première fondée sur la réalisation d'expériences au sein de la soufflerie hypersonique *TCM2* construite à l'Université de Provence à Marseille et la seconde basée sur la modélisation numérique dont le solveur est parallélisé sur le supercalculateur KSR1 de l'INRIA à Rocquencourt. La soufflerie à choc réfléchi à piston libre de l'Université de Provence est composée principalement, comme le montre la Figure 0, d'un réservoir d'air haute pression, d'un tube de compression contenant un piston libre, d'un tube à choc et d'une tuyère de détente associée à une chambre d'expérience [1].

Une telle installation expérimentale permet d'obtenir un nombre de Mach de l'ordre de 10 dans la chambre d'expérience, juste à la section de sortie de la tuyère. Grâce à cette soufflerie, la génération expérimentale d'un écoulement d'air à haute enthalpie, i.e. dans des conditions similaires à celles de la rentrée atmosphérique devient possible, mais comme nous le constatons, elle requiert l'utilisation d'un dispositif particulièrement complexe et coûteux. Pour ces raisons, avant de réaliser toute expérience dans la soufflerie, il est important de simuler numériquement l'écoulement au sein de la tuyère, ceci afin de prédire, en sortie de tuyère, les conditions thermochimiques de l'écoulement dans lequel sera placée la maquette. Dans ce travail, on s'intéresse donc plus particulièrement à la modélisation numérique de l'écoulement d'air stationnaire à l'intérieur d'une tuyère de détente axisymétrique [2]. Cet écoulement, en raison des conditions de haute enthalpie auxquelles il est soumis, se trouve dans un état de déséquilibre chimique et vibrationnel. L'air n'est alors plus un mélange de 79 % de N_2 et 21% de O_2 (approximativement) mais un mélange complexe à plusieurs composants [3]; le modèle retenu dans cette étude comprend 5 composants : O_2, N_2, O, N, NO. On suppose aussi que l'écoulement est non visqueux et laminaire. En ce qui concerne le déséquilibre chimique, on ne considère que les réactions de dissociation et d'échange [3]. En effet, en raison des plages de température moyennement élevées dans lesquelles évolue l'écoulement

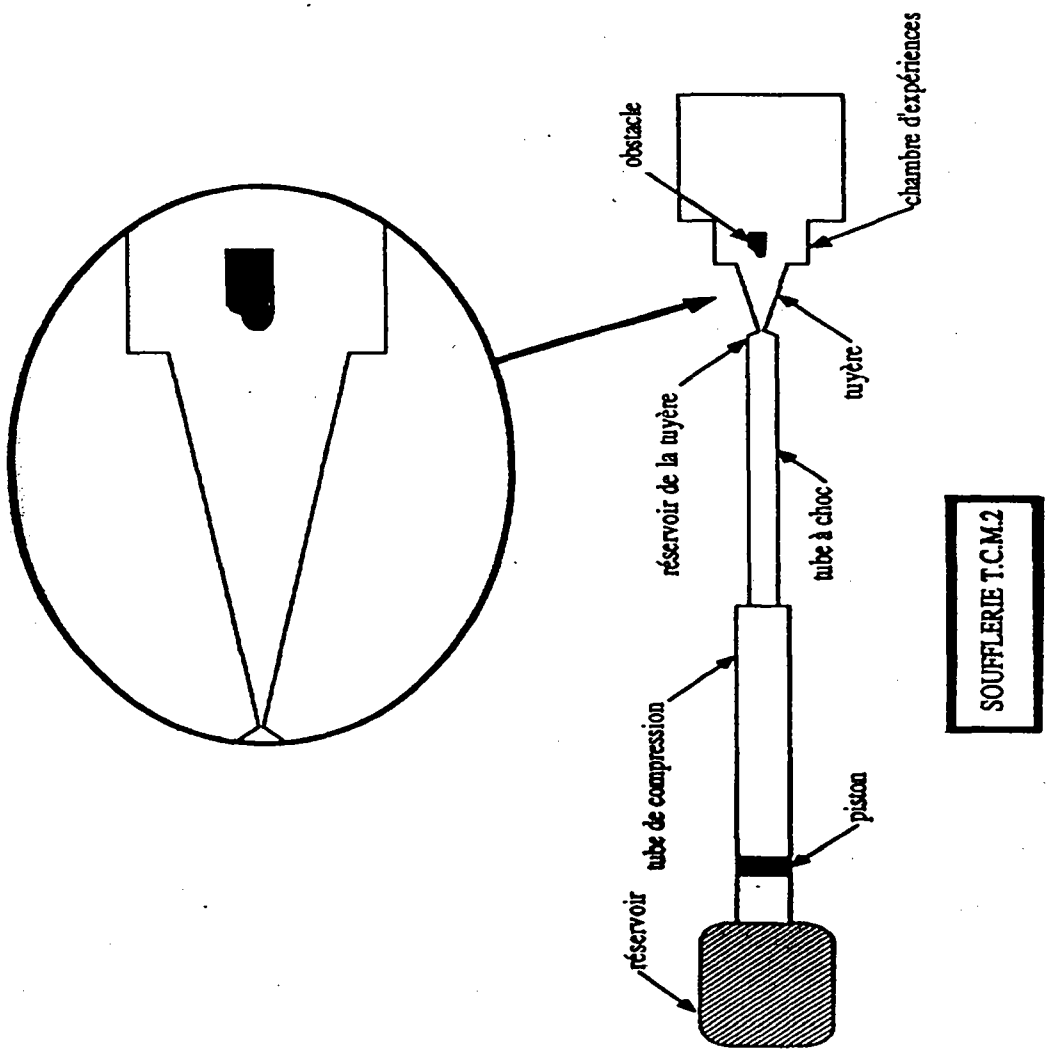

FIG. 0. *Schéma de principe de la soufflerie hypersonique TCM2 construite à l'IUSTI-MHEQ Marseille [1]*

(< 8000 K), on peut négliger le processus d'ionisation. Quant au déséquilibre thermodynamique, les modes translationnel et rotationnel sont à l'équilibre (leur temps de relaxation étant très court devant le temps caractéristique de l'écoulement); ainsi seul le mode vibrationnel est supposé en déséquilibre [4] : l'oxygène et l'azote diatomiques sont en déséquilibre vibrationnel, tandis que l'oxyde d'azote est considéré à l'équilibre vibrationnel (son temps de relaxation vibrationnel est très court

devant les autres temps [4]). Par ailleurs, vu que les temps de relaxation des processus chimiques et vibrationnels sont d'un ordre de grandeur proche, on s'attachera également à modéliser le couplage qui existe entre les 2 processus [5][6].

L'écoulement ainsi modélisé, est gouverné par un système de lois de conservation comprenant les équations d'Euler compressible, des équations d'évolution des composants chimiques du mélange, ainsi que des équations d'évolution des énergies de vibration des espèces hors d'équilibre vibrationnel. Cependant, malgré la simplification du problème tridimensionnel par une approche axisymétrique, et ce même sur des cas académiques (maillage à 3000 nœuds), la résolution numérique de ce type de modèle nécessite des temps *CPU* relativement importants (de l'ordre de quelques heures sur des supercalculateurs [7][8]). Pour cette raison, dans ce papier nous concentrons l'effort sur la parallélisation du code numérique sur la KSR1 afin de réduire les temps d'exploitation de telles simulations. La KSR1 (Kendall Square Research) se distingue en étant le 1er ordinateur MIMD à mémoire partagée virtuelle, et à architecture massivement parallèle et extensible, parmi le parc des supercalculateurs parallèles. En fait sur le plan hardware strict, cette machine se présente comme un ordinateur à mémoires distribuées; or ce type d'architecture pose un problème de taille quant à la portabilité des codes développés dessus et l'innovation majeure de KSR réside dans la technologie AllCache qui fait apparaître ces mémoires comme partagées. Cet aspect très important rend d'une part le développement de la programmation parallèle très convivial et attractif sur une architecture massivement parallèle et facilite d'autre part la parallélisation des codes existants ainsi que le portage de programmes initialement parallélisés sur d'autres calculateurs à mémoire partagée.

Le plan de ce papier est composé de trois parties. Ainsi, au paragraphe 1, on rappelle de façon détaillée les équations gouvernant le modèle retenu. Ensuite, on donne dans le paragraphe 2, la formulation variationnelle associée à ce problème, suivi de sa discrétisation en espace, dans un cadre volumes finis, avec l'utilisation du flux de Van-Leer. Cette formulation discrétisée est alors associée à un schéma en temps du premier ordre semi-implicite. Enfin, au paragraphe 3, on donne l'algorithme de résolution générale, ainsi qu'une analyse détaillée du sous programme représentant la plus grande part du temps de calcul; dans ce travail préliminaire on limite la parallélisation à cette étape. Après une description succinte de la KSR1, on présente les différentes stratégies utilisées lors de l'implémentation du code de calcul en vue de sa parallélisation. Trois cas tests, de tailles croissantes en calcul, ont fait l'objet de cette démarche. L'analyse comparative des résultats correspondants, illustrée par des courbes de performance, montre d'une part les excellents *Speed-Up* obtenus et d'autre part indique comment améliorer, sinon au moins garder, ces performances d'une part lors de la parallélisation globale du solveur et d'autre part lors des applications industrielles de très grande taille. On présente enfin quelques résultats donnés par le solveur, à l'aide de figures décrivant l'évolution des grandeurs physiques calculées de l'écoulement.

1 Position du Problème et Modélisation

Avant de poser le problème à résoudre, rappelons brièvement les conditions du modèle traité: l'étude concerne l'écoulement hypersonique, laminaire et non visqueux d'un mélange gazeux à 5 composants (O_2, N_2, O, N, NO) dont les espèces diatomiques O_2 et N_2 sont en déséquilibre vibrationnel. Ce type d'écoulement est gouverné par les équations d'Euler pour la partie aérodynamique, par 3 équations d'évolution des espèces chimiques (O, N, NO) exprimant leur conservation, ainsi que par 2 équations d'évolution des énergies vibrationnelles relatives à O_2 et N_2. Les évolutions chimiques de O_2 et N_2 sont déduites des 3 équations d'évolution des espèces O, N et NO, et de l'équation de conservation du nombre total d'atomes dans le mélange considéré. Soit Ω le domaine de l'écoulement, toutes ces équations s'écrivent sous la forme bien connue du système conservatif suivant:

$$\frac{\partial W}{\partial t} + \nabla . \mathcal{F}(W) = S \quad \text{dans} \quad \Omega \tag{1}$$

avec

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \\ \rho_{i_1} \\ \rho_{i_2} e_{V_{i_2}} \end{pmatrix} \quad , \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \omega_{C_{i_1}} \\ \omega_{V_{i_2}} \end{pmatrix} \tag{2}$$

$$i_1 \in I_1 = \{O, N, NO\} \quad \text{et} \quad i_2 \in I_2 = \{O_2, N_2\}$$

$\mathcal{F}(W)$ étant le flux des quantités de conservation; ρ, ρ_{i_1} et ρ_{i_2} désignent respectivement les densités du mélange, de l'espèce chimique $i_1 \in I_1$, et de l'espèce vibrante $i_2 \in I_2$; u, v et w les composantes de la vitesse de l'écoulement, e l'énergie totale spécifique du mélange et $e_{V_{i_2}}$ l'énergie spécifique de vibration de l'espèce i_2; enfin $(\omega_{C_{i_1}})$ et $(\omega_{V_{i_2}})$ représentent respectivement les termes sources chimiques et vibrationnels.

Pour fermer le système d'équations ainsi obtenu, on utilise l'équation d'état pour le mélange:

$$P = \rho R T \tag{3}$$

où P est la pression de l'écoulement, T sa température et R une moyenne des R_l, où $R_l = \frac{\mathcal{R}}{M_l}$ avec $\mathcal{R}$ la constante des gaz parfaits et M_l la masse molaire de l'espèce l. De plus, pour calculer la température de l'écoulement T on utilise l'expression de l'énergie totale ρe donnée par la loi de comportement suivante:

$$\rho e = \sum_{l \in I_1 \cup I_2} \rho_l C_{v_l} T + \sum_{l \in I_1 \cup I_2} \rho_l h_l^o + \sum_{l \in L} \rho_l e_{V_l} + \frac{1}{2}\rho(u^2 + v^2 + w^2) \tag{4}$$

avec $L = \{O_2, N_2, NO\}$, et où C_{v_l} et h_l^o dénotent respectivement la chaleur spécifique à volume constant et l'enthalpie spécifique de formation de chacun des composants du mélange.

Dans la suite de ce paragraphe, on s'attachera à expliciter brièvement les termes sources chimiques ($\omega_{C_{i_1}}$) et vibrationnels ($\omega_{V_{i_2}}$). Tout d'abord, en ce qui concerne le modèle de cinétique chimique, et ce en raison des hypothèses effectuées (absence d'ionisation), on décrit le déséquilibre chimique à l'aide de 17 réactions : 15 réactions de dissociation et 2 réactions d'échange [9].

$$\left.\begin{array}{lllllllll} O_2 & + & l & \rightleftharpoons & 2O & + & l & & \\ N_2 & + & l & \rightleftharpoons & 2N & + & l & & \\ NO & + & l & \rightleftharpoons & N & + & O & + & l \\ NO & + & O & \rightleftharpoons & O_2 & + & N & & \\ N_2 & + & O & \rightleftharpoons & NO & + & N & & \end{array}\right\} \tag{5}$$

où l vaut chacun des éléments de $I_1 \cup I_2$. On peut réécrire ces équations sous la forme générale suivante

$$\sum_{l \in I_1 \cup I_2} \nu'_{l,r}\, l \underset{K_{b,r}}{\overset{K_{f,r}}{\rightleftharpoons}} \sum_{l \in I_1 \cup I_2} \nu''_{l,r}\, l \quad , \quad r = 1, 17 \tag{6}$$

$\nu'_{l,r}$ et $\nu''_{l,r}$ sont les coefficients stœchiométriques correspondant à la réaction r et à l'espèce l. Les constantes directes ($K_{f,r}$) et inverses ($K_{b,r}$) de cinétique chimique, intervenant dans chaque réaction r, dépendent de la température T et sont données par le modèle de Evans et al[10] de la façon suivante :

$$K_{f,r}(T) = C_f T^{S_f} \exp(-\frac{\theta_D}{T}) \tag{7}$$

$$K_{b,r}(T) = C_b T^{S_b} \exp(-\frac{\theta_D}{T}) \tag{8}$$

où θ_D est la température caractéristique de dissociation des espèces diatomiques et C_f, C_b, S_f, S_b des constantes tabulées par Evans.

En considérant le bilan élémentaire de chaque réaction r, on peut exprimer le terme source chimique $\omega_{C_{i_1}}$ de l'équation (1), appelé aussi terme de production de l'espèce i_1, de la manière suivante:

$$\omega_{C_{i_1}} = M_{i_1} \sum_{r=1}^{17} \left(\nu''_{i_1,r} - \nu'_{i_1,r}\right) \left[K_{f,r} \prod_{l \in I_1 \cup I_2} \left(\frac{\rho_l}{M_l}\right)^{\nu'_{i_1,r}} - K_{b,r} \prod_{l \in I_1 \cup I_2} \left(\frac{\rho_l}{M_l}\right)^{\nu''_{i_1,r}} \right] \tag{9}$$

où $i_1 \in I_1$ et M_l est la masse de l'espèce l, $\forall l \in I_1 \cup I_2$. Or, l'interaction entre le déséquilibre chimique et le déséquilibre vibrationnel est forte car les temps de relaxation de ces processus sont du même ordre de grandeur ($10^{-4}s$ pour la relaxation vibrationnelle et $10^{-3}s$ pour la relaxation chimique). Il est donc nécessaire de tenir compte du couplage entre ces 2 processus. En effet, l'influence du déséquilibre vibrationnel sur les réactions chimiques (modèle CVD dû à Treanor et Marrone

[6]) se traduit de la façon suivante: une molécule diatomique $i_2 \in I_2$ excitée vibrationnellement va se dissocier plus facilement qu'une molécule non excitée; cette influence s'exprime alors par une modification de la constante directe de cinétique chimique qui dépendra alors non seulement de T mais également de la température vibrationnelle $T_{V_{i_2}}$ et d'un paramètre U_{i_2} permettant de privilégier la dissociation des molécules situées sur les niveaux supérieurs de vibration:

$$K_{f,r}(T, T_{V_{i_2}}, U_{i_2}) = K_{f,r}(T).V(T, T_{V_{i_2}}, U_{i_2}) \tag{10}$$

où $V(T, T_{V_{i_2}}, U_{i_2})$ est appelé facteur de couplage.

En ce qui concerne le modèle de relaxation vibrationnelle i.e. le calcul du terme source vibrationnel $\omega_{V_{i_2}}$, on considère dans le cadre de cette étude 2 types d'échanges d'énergie interne: les échanges vibration-translation (modèle VT) et les échanges vibration-vibration (modèle VV). Pour les échanges V-T, on a étendu au cas d'un mélange gazeux, le modèle de Landau-Teller [11] traitant les échanges V-T dans un gaz pur; le terme de production d'énergie vibrationnelle correspondant à ces échanges s'exprime alors de la façon suivante [2]:

$$(\omega_{V_{i_2}})_{VT} = \frac{E^o_{V_{i_2}} - E_{V_{i_2}}}{\tau^{i_2}_{VT}} + \frac{E_{V_{i_2}}}{\rho_{i_2}}\,\omega_{C_{i_2}} \tag{11}$$

où $i_2 \in I_2$, $\omega_{C_{i_2}}$ est déduit des $\omega_{C_{i_1}}$ par l'intermédiaire de la loi de conservation du nombre d'atomes, $E_{V_{i_2}} = \rho_{i_2} e_{V_{i_2}}$, $E^o_{V_{i_2}} = \rho_{i_2} e^o_{V_{i_2}}$ avec $e^o_{V_{i_2}}$ l'énergie spécifique de vibration à l'équilibre, i.e. à la température T de l'écoulement et enfin $\tau^{i_2}_{VT}$ est le temps de relaxation vibrationnelle de l'espèce i_2 pour les échanges V-T. Pour un mélange gazeux, ce temps $\tau^{i_2}_{VT}$ peut s'exprimer sous la forme d'une moyenne de plusieurs temps de relaxation de gaz purs:

$$\frac{1}{\tau^{i_2}_{VT}} = \sum_{l \in I_1 \cup I_2} \frac{\xi_l}{\tau^{i_2,l}_{VT}} \tag{12}$$

où ξ_l est la fraction molaire de l'espèce l dans le mélange à 5 composants.

Quant au terme de production d'énergie vibrationnelle correspondant aux échanges V-V on a retenu l'expression proposée par Stupochenko et al [12]:

$$\begin{aligned}(\omega_{V_{i_2}})_{VV} = &\frac{E_{V_{i_3}}\left(E_{V_{i_2}} + \rho_{i_2} R_{i_2} \theta_{V_{i_2}}\right) \exp\left(\dfrac{\theta_{V_{i_3}} - \theta_{V_{i_2}}}{T}\right)}{\tau^{i_2,i_3}_{VV} \cdot \rho R \theta_{V_{i_3}}} \\ &- \frac{E_{V_{i_2}}\left(E_{V_{i_3}} + \rho_{i_3} R_{i_3} \theta_{V i_3}\right)}{\tau^{i_2,i_3}_{VV} \cdot \rho R \theta_{V_{i_3}}}\end{aligned} \tag{13}$$

$$\forall\ i_2, i_3 \in I_2$$

avec R, respectivement R_l, la constante des gaz parfaits rapportée au mélange, respectivement à l'espèce $l \in I_2$ et θ_{V_l} la température caractéristique de vibration de l'espèce $l \in I_2$.

Afin de compléter le modèle CVD, i.e. pour prendre en compte l'influence des réactions chimiques sur le déséquilibre vibrationnel, on introduit le modèle de couplage CVDV dû à Treanor et Marrone [5] qui exprime le bilan d'énergie vibrationnelle lors de la dissociation et de la recombinaison des molécules diatomiques; ce bilan d'énergie peut s'écrire de la façon suivante :

$$(\omega_{V_{i_2}})_{CV} = \frac{\overline{G_{V_{i_2}}} - E_{V_{i_2}}}{\rho_{i_2}} \left(\frac{d\rho_{i_2}}{dt}\right)_b - \frac{\overline{E_{V_{i_2}}} - E_{V_{i_2}}}{\rho_{i_2}} \left(\frac{d\rho_{i_2}}{dt}\right)_f , \ \forall i_2 \in I_2 \tag{14}$$

où l'indice f désigne le phénomène lié à la dissociation et l'indice b le phénomène lié à la recombinaison; $\overline{E_{V_{i_2}}}$ est l'énergie de vibration moyenne perdue par dissociation

$$\overline{E_{V_{i_2}}} = \frac{\rho_{i_2} R_{i_2} \theta_{V_{i_2}}}{\exp\left(\frac{\theta_{V_{i_2}}}{T_{F_{i_2}}}\right) - 1} - \frac{N_{i_2} \rho_{i_2} R_{i_2} \theta_{V_{i_2}}}{\exp\left(\frac{N_{i_2} \theta_{V_{i_2}}}{T_{F_{i_2}}}\right) - 1} \tag{15}$$

avec $\frac{1}{T_{F_{i_2}}} = \frac{1}{T_{V_{i_2}}} - \frac{1}{T} - \frac{1}{U_{i_2}}$, où U_{i_2} peut être considéré comme la température caractéristique de recombinaison des molécules et N_{i_2} le nombre de niveaux de vibration de l'espèce i_2; $\overline{G_{V_{i_2}}}$ est l'énergie de vibration moyenne gagnée à chaque recombinaison

$$\overline{G_{V_{i_2}}} = \lim_{T_{V_{i_2}} \to T} \overline{E_{V_{i_2}}} \tag{16}$$

Ainsi, le terme source des équations d'évolution des énergies vibrationnelles peut s'écrire comme la somme des bilans d'énergie vibrationnelle dûs aux différents processus d'échanges:

$$\omega_{V_{i_2}} = (\omega_{V_{i_2}})_{VT} + (\omega_{V_{i_2}})_{VV} + (\omega_{V_{i_2}})_{CV} \tag{17}$$

2 Méthode de Solution

L'écoulement que l'on cherche à modéliser et à simuler au sein de la tuyère n'étant pas partout supersonique (zone subsonique dans le convergent), le système d'équations stationnaires n'est alors pas partout hyperbolique. Ceci nous a donc amené à adopter une approche instationnaire donnée par le système (1)-(2), pour lequel on recherche les solutions faibles à travers l'approche variationnelle.

2.1 Formulation Variationnelle et Schéma Numérique

On rappelle brièvement la méthode variationnelle de type Galerkin, appliquée à l'équation (1) comme suit:

$$\int_\Omega \left(\frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W)\right) \phi \, d\Omega = \int_\Omega S\phi \, d\Omega \ , \ \forall \phi \tag{18}$$

où Ω est le domaine de l'écoulement. Soit $\tilde{\Omega}$ le domaine discret associé à Ω et soit q_i un nœud du domaine $\tilde{\Omega}$, issu de la discrétisation spatiale (i allant de 1 à Ns, où Ns est le nombre de nœuds). L'approche variationnelle utilisée dans cette étude étant une approche du type volumes finis, on associe à chaque nœud q_i, un volume de contrôle $\mathcal{D}_i$ entourant ce nœud. Ces volumes sont construits de façon telle que la réunion des $\mathcal{D}_i$ soit égale à $\tilde{\Omega}$, avec la condition que l'intersection des $\mathcal{D}_i$ appartienne à la réunion des $\partial\mathcal{D}_i$. Si l'on prend pour ϕ la fonction caractéristique du volume de contrôle $\mathcal{D}_i$.

$$\phi(x,y,z) = \begin{cases} 1 & si\ (x,y,z) \in \mathcal{D}_i \\ 0 & sinon \end{cases} \tag{19}$$

on approche le problème continu (18) défini sur Ω par le problème discret suivant, défini sur $\tilde{\Omega}$

$$\int_{\mathcal{D}_i} \frac{\partial W}{\partial t} d\Omega + \int_{\partial\mathcal{D}_i} \mathcal{F}(W)\cdot n\ d\sigma = \int_{\mathcal{D}_i} S\ d\Omega \tag{20}$$

pour tout volume de contrôle $\mathcal{D}_i$; n est la normale extérieure définie sur chaque face de $\mathcal{D}_i$. Or, comme les configurations dans lesquelles évolue l'écoulement sont axisymétriques (tuyère axisymétrique), on introduit la notion de cellule $\mathcal{C}_i$ associée à chaque volume de contrôle $\mathcal{D}_i$ et qui est définie comme l'intersection de $\mathcal{D}_i$ et d'un plan passant par l'axe de symétrie du domaine. Ainsi, $\mathcal{D}_i$ peut être vu comme le volume issu de $\mathcal{C}_i$ par une rotation d'angle ε du plan contenant $\mathcal{C}_i$, autour de l'axe de symétrie. D'autre part, on choisit une discrétisation structurée, où chaque nœud q_i du domaine discret peut être repéré par 2 indices (j,k) de manière bijective: $j = 1, N_j$ (dans le sens axial du domaine) et $k = 1, N_k$ (dans le sens radial) tel que $Ns = N_j \times N_k$.

Compte tenu de ce qui précède, l'intégration du système (20) sur tout domaine $\mathcal{D}_i$, par l'utilisation d'une technique de perturbation de domaine [13], permet d'obtenir dans le cas axisymétrique, le système suivant:

$$mes(\mathcal{C}_{j,k})\frac{\partial W_{j,k}}{\partial t} + \sum_{p=1,\ (Arêtes\ de\ \mathcal{C}_{j,k})}^{4} \mathcal{F}^p(W_{j,k})\cdot n_p\ - aire(\mathcal{C}_{j,k})H_{j,k} = mes(\mathcal{C}_{j,k})S_{j,k}. \tag{21}$$

Ci-dessus, $aire(\mathcal{C}_{j,k})$ est l'aire de la cellule, $\mathcal{F}^p$ et n_p sont respectivement le flux et la normale sur la $p^{ième}$ arête de la cellule $\mathcal{C}_{j,k}$; $mes(\mathcal{C}_{j,k})$ désigne la mesure de la cellule $\mathcal{C}_{j,k}$ au sens suivant:

$$\begin{aligned} mes(C_{j,k}) &= \lim_{\epsilon\to 0}\left(\frac{1}{\varepsilon}\int_{\mathcal{D}_i} d\Omega\right) \\ &= -\frac{1}{3}\sum_{l=1}^{4}(y_l^2 + y_{l+1}^2 + y_l\ y_{l+1})(x_{l+1} - x_l) \end{aligned} \tag{22}$$

où (x_l, y_l) sont les coordonnées du nœud $q_l \in A(q_i)$ voisin de q_i et tel que

$$(x_5, y_5) = (x_1, y_1)$$

$A(q_i)$ étant l'ensemble des quatre nœuds voisins de $q_i \Longleftrightarrow (j,k)$:

$$A(q_i) = \{(j+1,k)\ ;\ (j,k+1)\ ;\ (j-1,k)\ ;\ (j,k-1)\} \tag{23}$$

Notons que la prise en compte de l'axisymétrie du problème d'une part simplifie le système d'équations, puisque l'équation sur la troisième composante (w) de la vitesse a disparu, au prix d'un terme source supplémentaire H, et d'autre part modifie l'expression de la fonction flux $\mathcal{F}$ [13]. Ainsi, nous obtenons

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \\ \rho_{i_1} \\ \rho_{i_2} e_{V_{i_2}} \end{pmatrix} \quad , \quad H = \begin{pmatrix} 0 \\ 0 \\ 2P \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad , \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_{C_{i_1}} \\ \omega_{V_{i_2}} \end{pmatrix} \tag{24}$$

avec P la pression de l'écoulement. Par soucis de simplicité d'écriture et suite à l'hypothèse d'axisymétrie, u représente maintenant la composante axiale de la vitesse ($u = v_z$ classiquement) et v la composante radiale ($v = v_r$).

Les flux entrant et sortant de la cellule $\mathcal{C}_{j,k}$, i.e. $\sum \mathcal{F}(W_{j,k}) \cdot n$, sont traités selon une technique de décomposition de flux proposée par Van-Leer [14] dans le cas des équations d'Euler pour un gaz parfait. L'extension de cette technique aux cas d'écoulements de gaz réels nécessite l'utilisation de la notion de "gamma équivalent", $\tilde{\gamma}$, utilisée par Gnoffo [15]. Ceci permet de prendre en compte l'influence du déséquilibre thermochimique sur l'écoulement sans changer la formulation de la décomposition de flux de Van-Leer. Le "gamma équivalent" est calculé comme le rapport entre l'enthalpie spécifique du mélange et son énergie spécifique, et peut être exprimé en fonction de ρ et de e:

$$\tilde{\gamma} = \frac{h}{e} = \tilde{\gamma}(\rho, e). \tag{25}$$

Ainsi, la décomposition de $\mathcal{F}(W)$, issue de la décomposition de flux de Van-Leer, est du premier ordre en espace et a pour expression:

$$\mathcal{F}(W) = \mathcal{F}^+(W) + \mathcal{F}^-(W) \tag{26}$$

où W est le vecteur conservatif défini en (24).

En ce qui concerne la discrétisation en temps, on utilise un schéma d'Euler retardé du premier ordre, défini par:

$$\frac{\partial W_{j,k}}{\partial t} = \frac{W_{j,k}^{m+1} - W_{j,k}^{m}}{\Delta t_{j,k}} \quad , \quad m = 1, Mt \tag{27}$$

où le pas de temps local $\Delta t_{j,k}$ est limité par la loi de CFL et Mt le nombre maximal d'itérations en temps autorisé. D'autre part, le schéma en temps utilisé est semi-implicite, car les termes de flux du système (21) sont traités explicitement, tandis

que, pour des raisons de stabilité, le terme source S doit être traité implicitement [2]:

$$S_{j,k}^{m+1} = S_{j,k}^{m} + \left(\frac{\partial S}{\partial W}\right)_{j,k}^{m} \delta W_{j,k}^{m+1} + O((\delta W_{j,k}^{m})^2) \tag{28}$$

avec $\delta W_{j,k}^{m+1} = W_{j,k}^{m+1} - W_{j,k}^{m}$. La difficulté engendrée par l'implicitation des termes sources $\omega_{C_{i_1}}$ et $\omega_{V_{i_2}}$ est relativement petite puisque la résolution du système est effectuée de façon découplée: chaque groupe d'équations est résolu séparément. (Les 4 premières équations de (21) et (24) constituent le groupe "Euler", et les variables correspondantes sont repérées par l'indice E, tandis que les 5^e, 6^e et 7^e équations constituent le groupe "Chimie" dont les variables sont indicées c, et enfin les 8^e et 9^e équations constituent le groupe "Vibration" et leur variable est identifiée par l'indice v). Grâce à ce découplage des équations, on est amené à calculer uniquement les matrices jacobiennes suivantes définies cellule par cellule:

$$(S_C')_{j,k}^{m} = \left(\frac{\partial \omega_{C_{i_1}}}{\partial \rho_{i_1}}\right)_{j,k}^{m} \quad , \quad i_1 \in I_1 \tag{29}$$

et

$$(S_V')_{j,k}^{m} = \left(\frac{\partial \omega_{V_{i_2}}}{\partial (\rho_{i_2} e_{V_{i_2}})}\right)_{j,k}^{m} \quad , \quad i_2 \in I_2 \tag{30}$$

Compte tenu des notations précédentes, la résolution du système complet (21) et (24) se ramène à la résolution des 3 systèmes simplifiés suivants, pour chaque cellule et à chaque pas de temps:

$$(\delta W_E)_{j,k}^{m+1} = -\frac{\Delta t_{j,k}}{mes(C_{j,k})} \sum_{\substack{p=1 \\ (Arêtes\ de\ C_{j,k})}}^{4} \mathcal{F}^p\left((W_E)_{j,k}^{m}\right) \cdot n_p + \Delta t_{j,k} \frac{aire(C_{j,k})}{mes(C_{j,k})} (H_E)_{j,k}^{m} \tag{31a}$$

$$\left(Id_{3\times 3} - \Delta t_{j,k}(S_C')_{j,k}^{m}\right)(\delta W_C)_{j,k}^{m+1} = -\frac{\Delta t_{j,k}}{mes(C_{j,k})} \sum_{\substack{p=1 \\ (Arêtes\ de\ C_{j,k})}}^{4} \mathcal{F}^p\left((W_C)_{j,k}^{m}\right) \cdot n_p + \Delta t_{j,k}(S_C)_{j,k}^{m} \tag{31b}$$

$$\left(Id_{2\times 2} - \Delta t_{j,k}(S_V')_{j,k}^{m}\right)(\delta W_V)_{j,k}^{m+1} = -\frac{\Delta t_{j,k}}{mes(C_{j,k})} \sum_{\substack{p=1 \\ (Arêtes\ de\ C_{j,k})}}^{4} \mathcal{F}^p\left((W_V)_{j,k}^{m}\right) \cdot n_p + \Delta t_{j,k}(S_V)_{j,k}^{m} \tag{31c}$$

où $Id_{N\times N}$ désigne la matrice identité de taille $(N \times N)$.

Quant aux conditions limites utilisées, elles se résument simplement à des conditions de glissement sur la paroi de la tuyère ($U.n = 0$, où U est la vitesse de

l'écoulement) d'une part, et à des conditions d'axisymétrie sur son axe ($v = 0$) d'autre part.

La résolution de ces trois systèmes découplés permet alors d'obtenir les variables conservatives de l'écoulement. Quant à la température T, elle est obtenue à partir de l'expression (4) de l'énergie totale, par une méthode itérative, puisque ρe dépend de façon non linéaire de T. Enfin, la pression P est déduite de l'équation d'état (3).

En ce qui concerne la phase de la mise en œuvre algorithmique, elle se déroule comme présentée dans le sous-paragraphe suivant.

2.2 Aspects Algorithmiques

La résolution numérique de notre modèle s'effectue en plusieurs étapes comme le montre l'algorithme global ci-dessous. Auparavant, et afin d'alléger les notations, on considère que chaque variable rencontrée dans l'algorithme ci-après et indicée j, k , indique que le calcul de celle-ci est effectué dans le cadre de la double boucle suivante:

Pour $k = 1, N_k$ Faire

Pour $j = 1, N_j$ Faire

.....
.....
.....
$(.)_{j,k}$
.....
.....
.....

Fin Faire

Fin Faire

Algorithme 1: Algorithme Global du Solveur

0. Lecture des données
1. Adimensionnement et Initialisations
2. Construction géométrique
   - 2.1. Génération du maillage ..... $q_i \longleftrightarrow (j,k)$
   - 2.2. Calcul des aires et des volumes ..... $mes(C_{j,k})$, $aire(C_{j,k})$
   - 2.3. Calcul des normales pour chaque $C_{j,k}$ ..... $(n^p)_{j,k}$, $p = 1,4$
3. Initialisation par une solution isentropique ..... $W^1_{j,k}$
4. Pour $m = 1, Mt$ Faire :
   - 4.1. Calcul de la température T à partir de (4) ..... $T^m_{j,k}$

Calcul de la pression P à partir de (3) ..... $P_{j,k}^m$

Calcul de $\tilde{\gamma}$ à partir de (25) ..... $\tilde{\gamma}_{j,k}^m$

4.2. Calcul du pas de temps local ..... $\Delta t_{j,k}$

4.3. Etape Euler (Variables indicées par E)

Calcul des flux de Van-Leer pour W_E à partir de (26) ..... $\mathcal{F}((W_E)_{j,k}^m)$

Traitement des conditions aux limites et résolution de (31a) .... $(W_E)_{j,k}^{m+1}$

..... i.e. $(\rho)_{j,k}^{m+1}$, $(\rho u)_{j,k}^{m+1}$, $(\rho v)_{j,k}^{m+1}$, $(\rho e)_{j,k}^{m+1}$

4.4. Etape Chimie (Variables indicées par c)

Calcul des termes sources chimiques ..... $(\omega_{C_{i_1}})_{j,k}^m$

Calcul du Jacobien à partir de (29) ..... $(S_C')_{j,k}^m$

Calcul des flux de Van Leer pour W_C à partir de (26) ..... $\mathcal{F}((W_C)_{j,k}^m)$

Résolution de (31b) ..... $(W_C)_{j,k}^{m+1}$

..... i.e. $(\rho_{i_1})_{j,k}^{m+1} (i_1 \in I_1)$

4.5. Etape Vibration (Variables indicées par v)

Calcul des termes sources vibrationnels ..... $(\omega_{V_{i_2}})_{j,k}^m$

Calcul du Jacobien à partir de (30) ..... $(S_V')_{j,k}^m$

Calcul des flux de Van Leer pour W_V à partir de (26) ..... $\mathcal{F}((W_V)_{j,k}^m)$

Résolution de (31c) ..... $(W_V)_{j,k}^{m+1}$

..... i.e. $(\rho_{i_2} e_{V_{i_2}})_{j,k}^{m+1} (i_2 \in I_2)$

4.6. Calcul du résidu; si convergence aller à Fin Faire

4.7. Mise à jour de la solution ..... $(W_{j,k}^m \longleftarrow W_{j,k}^{m+1})$

Fin Faire

A ce stade, il est à noter que le poids de calcul des différentes étapes est assez inégal. Dans ce travail précurseur à une parallélisation de l'ensemble du solveur, on s'est limité à résoudre en parallèle uniquement l'étape 4.4 du déséquilibre chimique qui consomme 65% du temps *CPU* global du solveur. Cette étape, réorganisée pour les besoins d'un calcul parallèle, se présente sous une forme très adéquate à la parallélisation, comme l'illustre bien l'algorithme détaillé suivant. Auparavant, il est à noter que l'algorithme présenté ci-après correspond à la version optimisée de cette étape.

Algorithme 2: Algorithme détaillé de l'étape 4.4

Pour $k = 1, N_k$ Faire

Pour $j = 1, N_j$ Faire

1 Calcul des termes sources chimiques

Pour $r = 1, 17$ Faire

Calcul de la cste cinétique directe $K_{f,r}(T)$ à partir de (7) ..... $K_{f,r}(T^m_{j,k})$
Calcul de la cste cinétique inverse $K_{b,r}(T)$ à partir de (8)..... $K_{b,r}(T^m_{j,k})$
Calcul de la cste cinétique directe couplée $K_{f,r}(T, T_{V_{i_2}}, U_{i_2})$ à partir de (10)
..... $K_{f,r}(T, T_{V_{i_2}}, U_{i_2})^m_{j,k}$

Fin Faire

Pour $i_1 \in I_1$ Faire

Calcul du terme source chimique $(\omega_{C_{i_1}})$ à partir de (9) ..... $(\omega_{C_{i_1}})^m_{j,k}$

Fin Faire

2 Calcul des jacobiens

Pour $r = 1, 17$ Faire
Pour $i_1 \in I_1$ Faire
Calcul de $\frac{\partial K_{f,r}}{\partial \rho_{i_1}}$, $\frac{\partial K_{b,r}}{\partial \rho_{i_1}}$ ..... $\left(\frac{\partial K_{f,r}}{\partial \rho_{i_1}}\right)^m_{j,k}$, $\left(\frac{\partial K_{b,r}}{\partial \rho_{i_1}}\right)^m_{j,k}$
Fin Faire
Fin Faire

Pour $i_1 \in I_1$ Faire
Pour $i_1 \in I_1$ Faire
Calcul du Jacobien S'_C à partir de (29)..... $\left(S'_C\right)^m_{j,k}$
Fin Faire
Fin Faire

3 Calcul des flux de Van Leer pour W_C à partir de (26) ..... $\mathcal{F}((W_C)^m_{j,k})$

4 Résolution de (31b) ..... $(W_C)^{m+1}_{j,k}$
..... i.e. $(\rho_{i_1})^{m+1}_{j,k} (i_1 \in I_1)$

Fin Faire

Fin Faire

3 Implémentation Parallèle et Résultats Numériques

Malgré la simplification relative apportée par l'approche axisymétrique décrite au paragraphe 2, le coût en temps *CPU* de la résolution numérique reste élevé. En effet, la complexité croissante du calcul inhérent au modèle traité, provient essentiellement de la prise en compte du déséquilibre thermochimique, comme le prouvent les étapes 4.4 et 4.5 de l'algorithme 1. Par ailleurs, et ce pour des maillages académiques de l'ordre de 3000 nœuds, la convergence vers une solution stationnaire nécessite environ 3 heures sur une machine performante mono-processeur de type RISC. Or, pour des configurations de maillages plus réalistes (de l'ordre de

10^5 – 10^6 nœuds), il est clair que l'exploitation informatique de ce solveur consommerait des jours voire des semaines de calcul; afin d'y remédier, la parallélisation de tels codes semble être incontournable.

Nous avons constaté, à travers les multiples expériences numériques, lors de la mise en œuvre informatique du solveur, que l'étape 4.4 (Chimie) consomme environ 65% du temps *CPU* sur n'importe quel ordinateur mono-processeur. Dans le but d'une parallélisation globale de ce solveur, on fait porter l'effort dans un premier temps, sur le traitement parallèle de cette étape, tout en notant que la parallélisation de l'étape 4.5 ne demanderait pas d'effort supplémentaire sur le plan algorithmique, vu sa structure similaire. Ainsi, après une phase cruciale d'optimisation scalaire, par une réorganisation des différents tableaux et variables, l'étape 4.4 s'est avérée prête à être parallélisée de façon naturelle, comme le montre bien l'algorithme 2. L'architecture à mémoire partagée virtuelle de la KSR1 nous a facilité la réalisation de notre objectif, comme le montrent les excellentes performances présentées ci-après. Auparavant, on décrit brièvement les caractéristiques techniques de la KSR1 et pour de plus amples détails ou analyses on se réfère aux documents [16][17][18][19].

3.1 Présentation sommaire de la KSR1

La KSR1 se présente comme le premier supercalculateur hautement parallèle extensible ("scalable") de 8 à 1088 cellules, muni d'une mémoire partagée virtuelle. Le système d'exploitation de la KSR1 est un UNIX multi-threads basé sur OSF1. Globalement, la topologie d'interconnection des cellules est hiérarchique à deux niveaux. Au niveau 0, appelé Ring:0, les cellules sont organisées par groupes ou anneaux qui contiennent chacun 32 cellules au maximum. Au niveau 1, 34 groupes du type Ring:0 au plus sont connectés via un "Ring-of-Rings", noté Ring:1. En fait, chaque Ring:0 contient également 2 "slots" supplémentaires pour assurer d'une part l'interconnection des différentes cellules du Ring:0, et d'autre part la communication entre les différents Ring:0 constituant le Ring:1.

Concernant la structure des cellules, chacune comprend un processeur du type RISC à 64 bits de fréquence 20 MHz, particulièrement adapté à la migration des données ainsi qu'au calcul flottant, doté d'une puissance de crête de 40 Mflops; de plus cette cellule dispose d'une part d'une mémoire locale de 32 Mo appelée LocalCache et d'autre part d'une mémoire virtuelle notée AllCache Engine:0. Par ailleurs, chaque processeur est muni d'un cache interne à accès rapide, appelé Sub-Cache, constitué de 256 Ko pour les données et de 256 Ko pour les instructions. En outre, il est à noter que la structure interne de chaque processeur, notamment à travers ses unités fonctionnelles, permet d'exécuter 2 instructions en un cycle ce qui explique sa performance en calcul flottant.

Pour ce qui est de l'innovation AllCache, cette technologie permet à l'utilisateur de programmer dans un environnement de mémoire partagée virtuelle, alors que la KSR1 est une machine à mémoire distribuée sur le plan hardware. Ceci rend facile

le portage d'applications initialement développées sur des machines MIMD traditionnelles d'une part, et un développement convivial de nouveaux programmes parallèles d'autre part. La mémoire de chaque cellule fait partie d'un espace hiérarchique d'adresses virtuelles. Le AllCache gère automatiquement la migration des données ainsi que leur adresse; les temps moyens d'attente et les capacités des différents niveaux d'intervention de AllCache sont donnés dans le tableau indicatif suivant :

Mémoires	Temps en Cycles	Capacités
SubCache	2	256 KB
LocalCache	18	32 MB
Ring:0	126	1 GB
Ring:1	600	34 GB

La prise en compte de ces caractéristiques hardware peut s'avérer très utile en vue d'une optimisation poussée de la parallélisation[20][21].

Sur le plan software, la plupart des langages de programmation évolués sont supportés par le compilateur de la KSR1; notamment, en ce qui concerne l'environnement du Fortran 77, on dispose d'un optimiseur-paralléliseur automatique (*KAP*) qui est un préprocesseur Fortran-Fortran. Cet optimiseur a été utilisé dans ce travail pour l'obtention d'une analyse de dépendance des différentes parties du programme; le résultat de cette analyse nous a permis d'optimiser l'aspect séquentiel du programme: malgré la complexité de notre programme, *KAP* nous a permis d'alléger la structure des données et donc d'obtenir une meilleure localité de celles-ci.

Par ailleurs, la parallélisation a été effectuée à l'aide de directives internes insérées dans le code et reconnues par le compilateur fortran, combinées dans certains cas à l'utilisation de certaines variables d'environnement au niveau de l'OS ("Operating System"). Les trois principales constructions de parallélismes sont les directives suivantes :

TILES: elles parallélisent les boucles en les découpant en groupes d'itérations,
SECTIONS: elles exécutent plusieurs fragments de code en parallèle,
REGIONS: elles exécutent en parallèle plusieurs copies d'un même fragment de code.

Au vu de l'algorithme 2, la directive *TILE* s'est avérée suffisante pour l'optimisation parallèle recherchée. En fait, plusieurs stratégies de tiling sont disponibles: *SLICE*, *MOD*, *WAVE* et *GRAB*. Les trois premières sont statiques, i.e. les arguments (nombre de threads, nombre d'itérations par groupe) de la directive *TILE* correspondante sont fixés dès le début de l'exécution du programme; ces stratégies privilégient la localité des données et le minimum d'overhead à la répartition des charges. Quant à la dernière stratégie, à savoir *GRAB*, elle permet une affectation dynamique des tiles au cours de l'exécution elle-même; ainsi

cette stratégie complètement dynamique privilégie l'équilibrage des charges sur les processeurs avec un coût minimum de mise en œuvre des threads.

3.2 Performances parallèles

Dans le cadre des trois cas tests réalisés dans ce travail et présentés ci-dessous, nous avons opté pour une parallélisation autour de la boucle sur k dans l'algorithme 2, à l'aide des deux stratégies du tiling *SLICE* et *MOD*; on rappelle que $k = 1, N_k$ avec N_k le nombre de nœuds dans le sens radial du domaine. Brièvement, la stratégie *SLICE* consiste à découper l'espace de la boucle de manière telle que les itérations soient également réparties sur les threads (tâches), chaque thread exécutant une seule tile; autrement dit, cette stratégie nécessite une bijection entre le nombre de threads et le nombre de tiles. Quant à la stratégie *MOD*, elle permet un découpage plus fin en comparaison avec celui donné par la stratégie *SLICE*: une même thread exécute plusieurs tiles. De plus, elle fait en sorte que l'affinité des données soit maintenue pour une pthread.

Sur le plan de la discrétisation spatiale du domaine de calcul, nous avons adopté des maillages du type structuré ($N_j \times N_k$); les deux premiers cas tests traités sont notés (*a*) et (*b*), et correspondent respectivement à un maillage (61 × 16) et (171 × 16). Pour ces deux cas, nous avons considérés 10 pas de temps de calcul, ce qui ne présente comme intérêt que l'évaluation de l'influence de la granularité sur les performances parallèles du solveur. En ce qui concerne le dernier cas (*bb*), c'est le même maillage que dans le cas (*b*) sauf que le nombre d'itérations en temps est de 3000, ce qui assure la convergence vers une solution stationnaire. Ce cas (*bb*) servira à consolider et à vérifier la stabilité des performances obtenues au cas (*b*), lors d'un calcul "relativement important".

Vu le faible nombre de nœuds, nous nous sommes limités dans cette étude à une exploitation sur 16 processeurs au maximum. Avant de procéder à la présentation des performances mesurées, on introduit les notations suivantes utilisées dans la légende des courbes:

$NPROC$:	ensemble de nombre de processeurs=$\{1,2,4,8,16\}$
$TIME/SEC$:	Temps en secondes
tU:	Temps de calcul *User* (appelé couramment Temps CPU)
tE:	Temps de calcul *Elapsed* (englobe tU et Temps système)
Paral. $tU(nproc)$:	tU obtenu en parallèle sur $nproc$ processeurs, $nproc \in NPROC$
Paral. $tE(nproc)$:	tE obtenu en parallèle sur $nproc$ processeurs, $nproc \in NPROC$
Seq. $tU(1)$:	tU obtenu en séquentiel
Seq. $tE(1)$:	tE obtenu en séquentiel
Speed–Up:	Rapport de $tU(1)$ respectivement $tE(1)$, en parallèle ou en séquentiel, par rapport à $tU(nproc)$ respectivement à $tE(nproc)$, en parallèle avec $nproc \in NPROC$
Efficiency:	Rapport de $tU(nproc)$ sur $tE(nproc)$, $nproc \in NPROC$

Vu le grand nombre de cas test effectués, et dans le but de faciliter la lecture

d'une figure, il s'est avéré utile de les codifier de façon aussi complète que possible. Pour les performances absolues en temps (Figures 1, 2, 7 et 14), nous avons utilisé la nomenclature suivante :

Radical1.Radical2.Suffixe

Radical1 comporte successivement et dans l'ordre les informations suivantes :
- le nom du cas traité ($(a), (b), (bb)$)
- la stratégie utilisée (M pour *MOD*, S pour *SLICE*)
- la taille de la tile ou "*Tilesize*" (T suivi de deux caractères xx ou 00, xx indiquant que cette taille est imposée en tant qu'argument de la directive *TILE*, 00 indiquant qu'elle est laissée libre, i.e. ajustable de façon automatique par *Presto* lors de l'exécution)
- le nombre de threads demandé ou "*Numthreads*" (N suivi de deux caractères xx ou 00, xx indiquant que ce nombre est imposé comme argument de *TILE*, 00 indiquant qu'il est laissé libre)

Radical2 correspond au nombre de threads demandées au niveau de l'OS (00 indique que rien n'a été demandé au niveau de l'OS)

Suffixe vaut U pour le temps *User* et E pour le temps *Elapsed*.

Il est à noter que si le nombre de threads est demandé explicitement en tant que paramètre de la directive *TILE* (Nxx), il est laissé libre au niveau de l'OS (.00), et réciproquement. Quant aux performances relatives ou *Speed-Up*, i.e. définies comme le rapport de deux temps, elles sont repérées par la nomenclature définie ci-dessus, à laquelle on a rajouté sur la gauche du Radical1, un préfixe de 3 caractères:

- r indique qu'il s'agit d'un rapport
- pp indique que ce rapport est effectué entre un temps parallèle sur *nproc* processeurs et un temps parallèle sur un processeur
- ps indique que ce rapport est effectué entre un temps parallèle sur *nproc* processeurs et un temps séquentiel

Tous les cas de calcul annoncés ont été exécutés après la commande *Allocate_Cells nproc*, $nproc \leq 16$ au niveau de l'OS, ce qui laisse la machine accessible à d'autres utilisateurs éventuels sur les cellules non dédiées. Par ailleurs, on rappelle que toutes les performances présentées ci-après concernent uniquement l'étape chimie.

Tout d'abord, on présente les performances du cas test (b). Sur les Figures 1 et 2, on a tracé les performances absolues, i.e. les temps *User* et *Elapsed* exprimés en secondes, en fonctions du nombre de processeurs utilisés.
Plus précisément, on a présenté sur une même figure, afin de les comparer, les temps obtenus avec les deux stratégies *SLICE* et *MOD*, et avec trois variantes différentes en ce qui concerne la stratégie *MOD*. Au vu des courbes, on peut remarquer que les écarts de temps entre les différentes options sont peu importants.

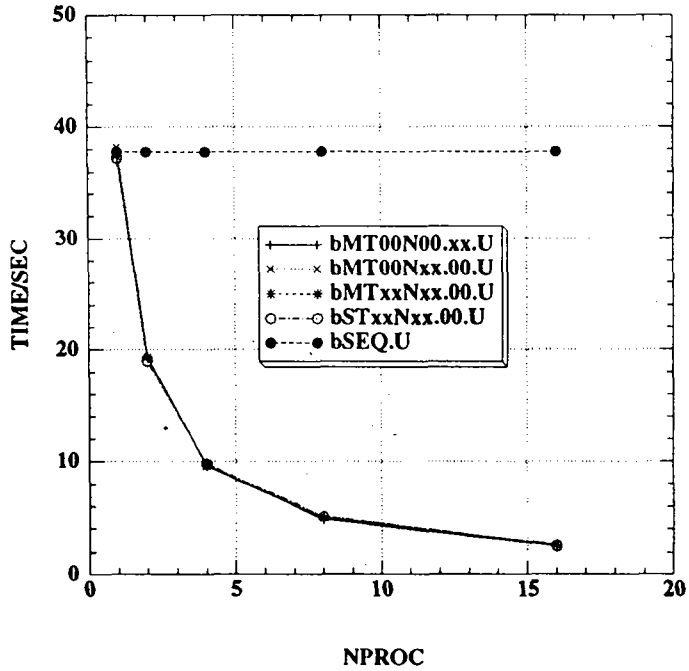

FIG. 1. *Temps User (tU) Parallèle et Sequentiel: cas b Slice & Mod Strategy*

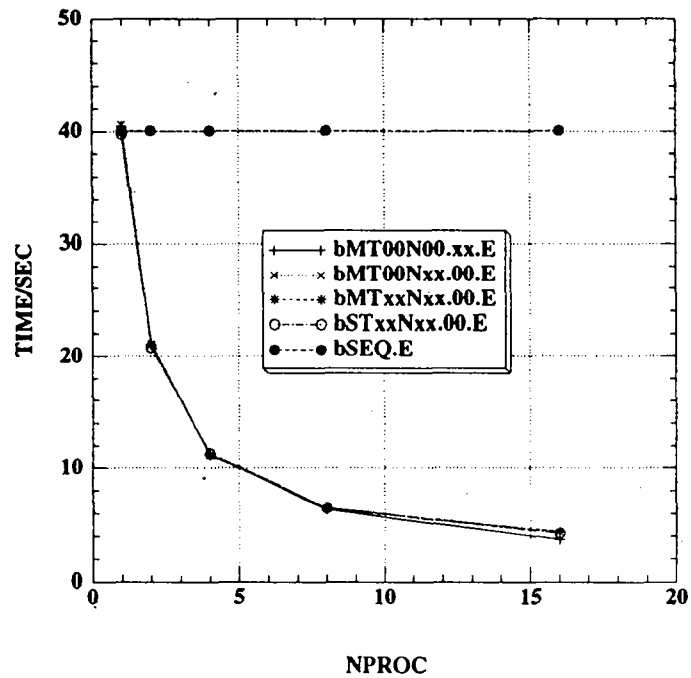

FIG. 2. *Temps Elapsed (tE) Parallèle et Sequentiel: cas b Slice & Mod Strategy*

Pour affiner les comparaisons de performances entre les diverses versions de *TILE* selon les arguments utilisés, on présente sur les Figures 3 à 6 les performances relatives, i.e. les *Speed–Up* ps et pp en temps *User* ou en temps *Elapsed*, en fonction

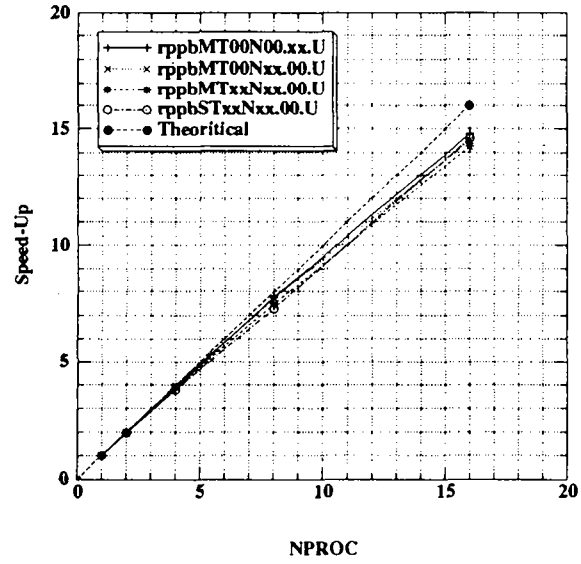

FIG. 3. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$

cas b : Slice & Mod Strategy

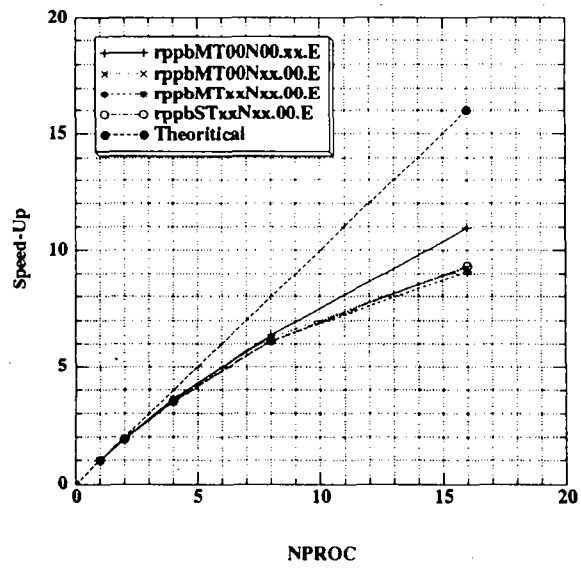

FIG. 4. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$

cas b : Slice & Mod Strategy

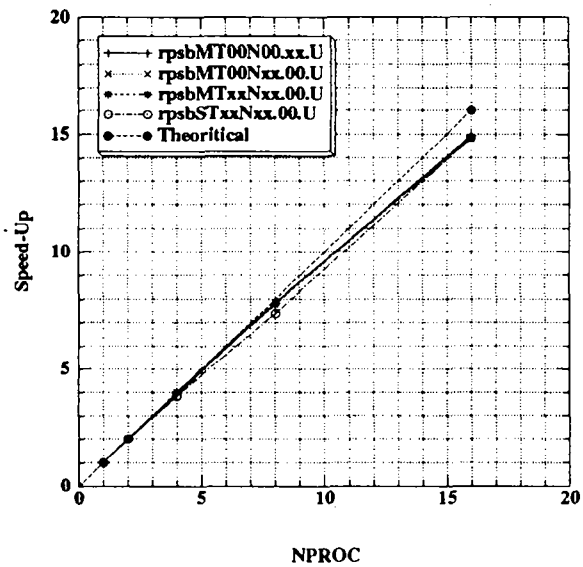

FIG. 5. $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$

cas b : Slice & Mod Strategy

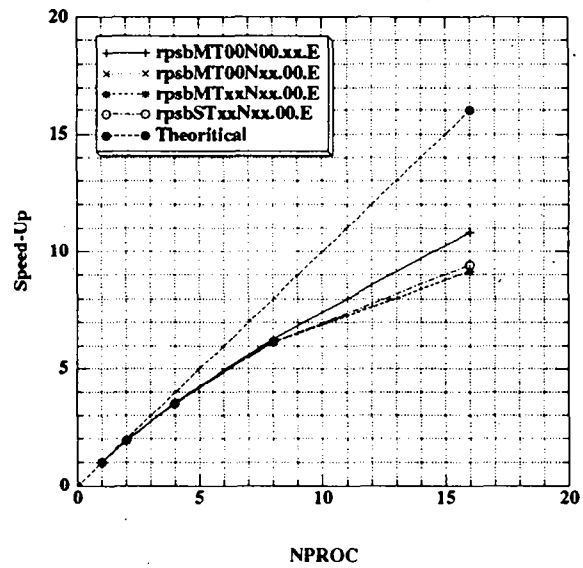

FIG. 6. $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$

cas b : Slice & Mod Strategy

du nombre de processeurs utilisés. Afin d'avoir toujours à la vue l'élément de référence, à savoir la performance idéale, nous avons tracé en plus des courbes *Speed–Up* calculées, celles du *Speed–Up* théorique.

Dans le cas des *speed–Up* pour les temps *User* (Figure 3 et 5), on peut remarquer globalement les très bonnes performances obtenues, que ce soit le *Speed–Up* calculé

par rapport à la version parallèle sur un processeur (pp), Figure 3, ou celui calculé par rapport à la version purement séquentielle du code (ps), Figure 5. On observe à travers ces performances sur le cas (*b*) que la première variante de la stratégie *MOD* (T00N00.xx), à savoir celle où l'on indique uniquement le nombre de threads au niveau de l'OS, donne les meilleurs résultats; on note cette variante *MOD* automatique. Ceci dit, la performance avec la stratégie *SLICE* est sensiblement identique à celle avec la stratégie *MOD* automatique, avec un léger avantage à cette dernière. En ce qui concerne les performances pour les temps *Elapsed* (Figures 4 et 6), on peut remarquer qu'elles sont beaucoup plus sensibles aux stratégies et arguments de *TILE* utilisés que pour les temps *User*. Mais il ne faut pas oublier que le temps *Elapsed* dépend étroitement de la charge de la machine et des overheads créés par l'intervention de Presto ou dûs à une localité des données non optimale.

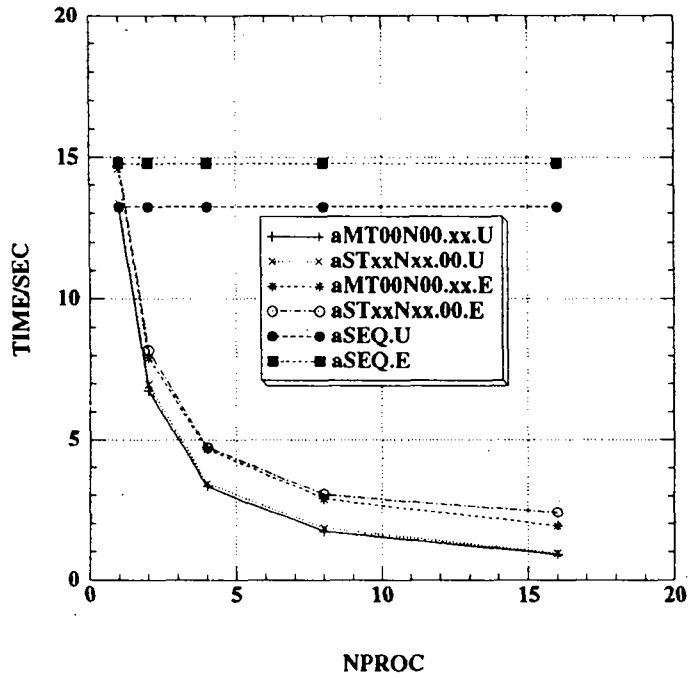

FIG. 7. *tU , tE Parallèle et Sequentiel: cas a Slice & Mod Strategy*

Maintenant, afin de cerner l'impact de la granularité sur ces deux stratégies, on les applique au cas (*a*) (61 × 16), bien plus petit que le cas (*b*). Les performances en temps absolu (*User* et *Elapsed*) sont présentées sur la Figure 7 et en *Speed–Up* (pp et ps) sur les Figures 8 et 9. D'une façon générale, on remarque que qualitativement l'allure des courbes de performances est similaire au cas (*b*), mais quantitativement les *Speed–Up* ici obtenus sont moins bons que ceux du cas (*b*). En effet, pour les *Speed–Up* relatifs au temps *User*, on note que les performances obtenues en pp et ps sur la Figure 8 sont du même ordre (14-15 sur 16 processeurs) que pour le cas (*b*) (Figures 3 et 5), avec toujours un avantage de la stratégie *MOD* sur la stratégie *SLICE*. Par contre, pour les *Speed–Up* en temps *Elapsed*, on passe d'un *Speed–Up*

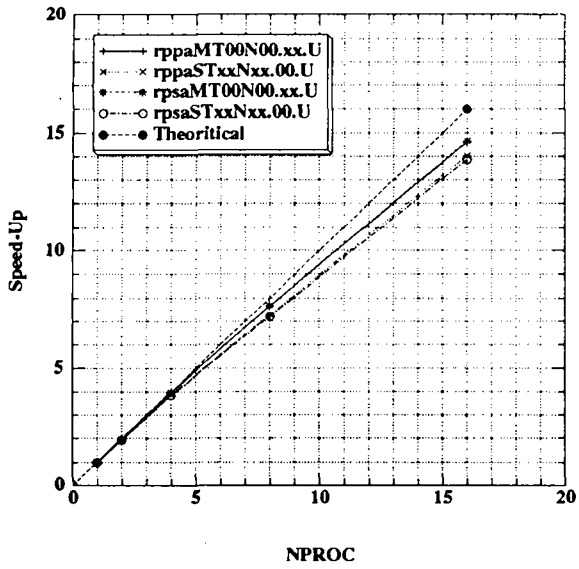

FIG. 8. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$ & $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$
cas a : Slice & Mod Strategy

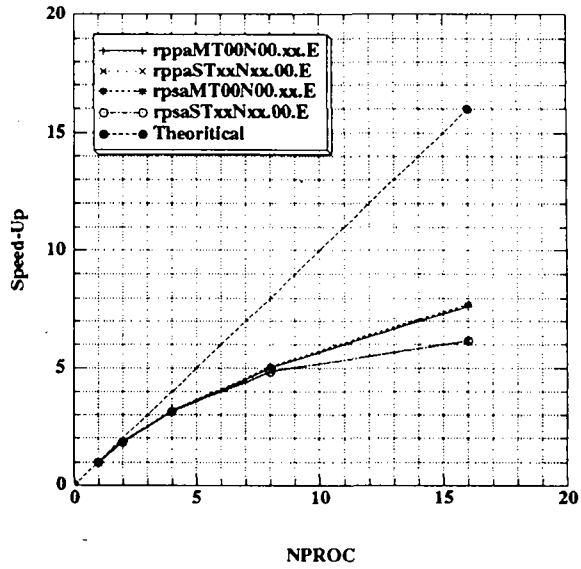

FIG. 9. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$ & $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$
cas a : Slice & Mod Strategy

compris entre 9 et 11 sur 16 processeurs dans le cas (*b*) à un *Speed–Up* compris entre 6 et 8 dans le cas (*a*): la faible granularité (et par conséquent des tiles de poids faible), le temps système relativement constant pour la gestion des threads et des tiles, ainsi que la quantité de calcul relativement petite (10 itérations en temps) jouent concurrentiellement pour diminuer la performance, comme l'explique la remarque suivante.

Remarque 3.1. Prenons par exemple le cas de parallélisation de la boucle k sur 16 processeurs i.e. le traitement de 16 tâches parallèles. Soit *nop* le nombre d'opérations dans la boucle k, i.e. pour chaque thread, et ce, à chaque itération en temps, et soit *tsys* le temps système nécessaire pour gérer ces *nop* opérations. Si l'on effectue 10 itérations en temps, le temps de calcul de ces opérations est $tnop = 10\ nop$ pour chaque thread, auquel il faut ajouter le temps système (quasi incompressible) pour gérer ces 10 *nop*. On obtient donc un temps global t, pour les 10 itérations et pour chaque processeur: $t = tnop + tsys$. Si *nop* est petit (cas a), *tnop* l'est aussi et donc *tsys* n'est pas négligeable devant *tnop*. Par contre, pour 3000 itérations, le temps de calcul des *nop* opérations est $tnop = 3000\ nop$ pour chaque thread, auquel on ajoute encore un temps système qui est quasi le même que *tsys* (car, en raison de la localité des données affectées à chaque thread -et conservée au cours des itérations-, le système a moins besoin de gérer ces opérations sur les différents processeurs). Ainsi, lorsque les itérations en temps augmentent, si l'on utilise les mêmes jeux de données (contenu et

adresse), le *tsys* cumulé augmente très peu d'une itération à une autre, et devient donc négligeable devant le temps de calcul effectif qui, lui, croît linéairement avec le nombre d'itérations. Cette remarque prendra toute sa signification dans le calcul du cas (*bb*).

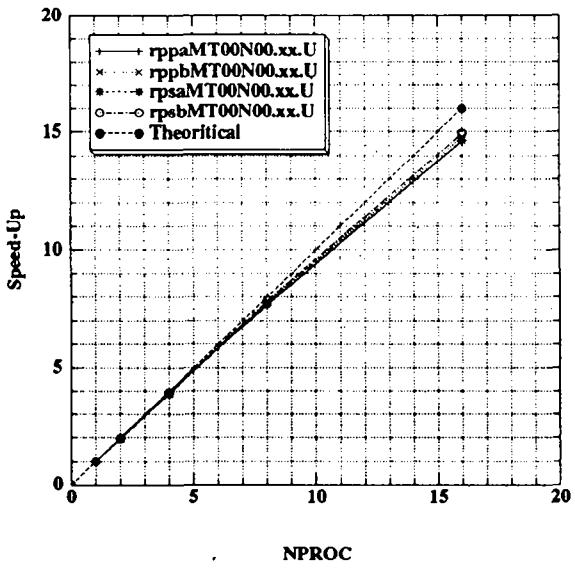

FIG. 10. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$ & $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$

cas a & b : Mod Strategy

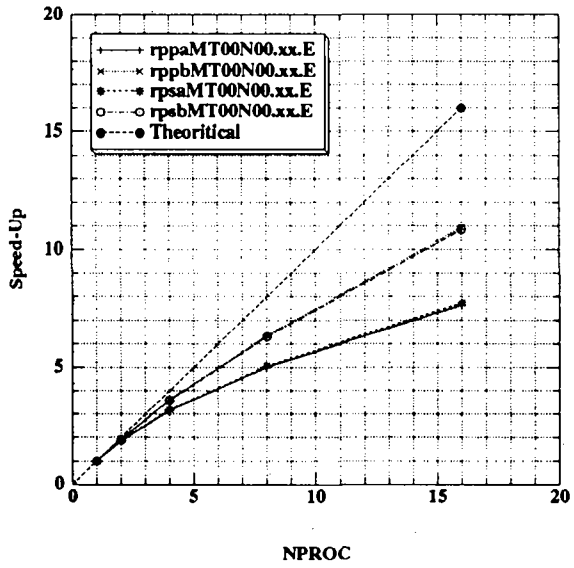

FIG. 11. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$ & $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$

cas a & b : Mod Strategy

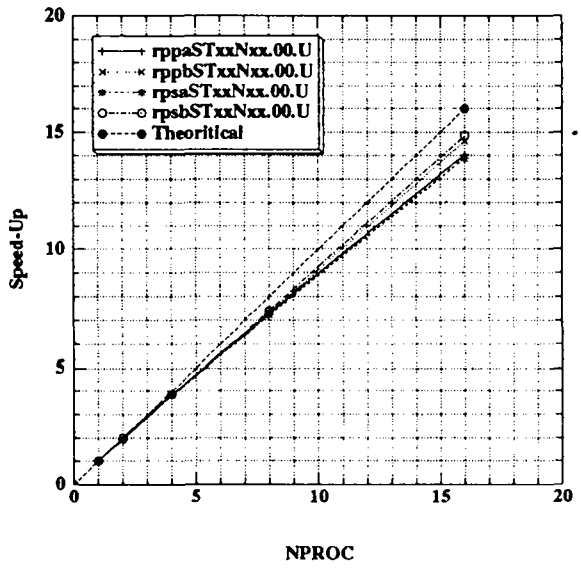

FIG. 12. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$ & $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$

cas a & b : Slice Strategy

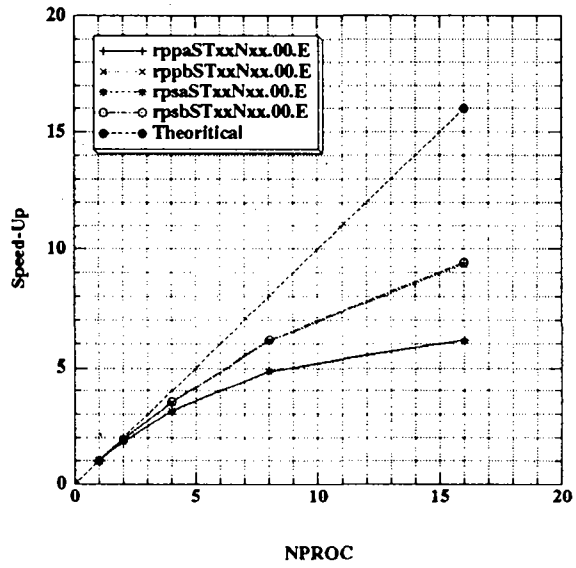

FIG. 13. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$ & $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$

cas a & b : Slice Strategy

Ainsi ce faible *Speed–Up* en temps *Elapsed* dans le cas (*a*), à savoir 61, s'explique facilement par le fait que la taille du problème est plus petite que dans le cas (*b*) (171). En conséquence, la granularité étant plus faible, le découpage de la boucle *k* avec la stratégie *SLICE* peut s'avérer plus "inégal" qu'avec la stratégie *MOD*. Cette observation est confirmée par l'analyse des courbes de comparaison entre les cas (*a*) et (*b*), Figures 10 à 13. Dans le cas des *Speed–Up* pour les temps *User*, on note que la différence entre les performances issues des cas (*a*) et (*b*) est très légère lors de l'implémentation de la stratégie *MOD*, Figure 10, mais est plus marquée lors de l'emploi de la stratégie *SLICE*, Figure 12, quel que soit le type de rapport calculé (pp ou ps); ceci est naturel vu que cette dernière privilégie la forte granularité. De plus, pour chaque comparaison, la meilleure performance provient du cas (*b*), ceci s'expliquant par la plus forte granularité du cas (*b*) par rapport au cas (*a*). Pour ce qui est des *Speed–Up* relatifs au temps *Elapsed*, Figures 11 et 13, en pp et ps, on constate la même chose que pour le temps *User*, mis à part que les performances dans l'absolu sont plus faibles d'une part et que la différence des *Speed–Up* entre les deux stratégies est nettement plus importante.

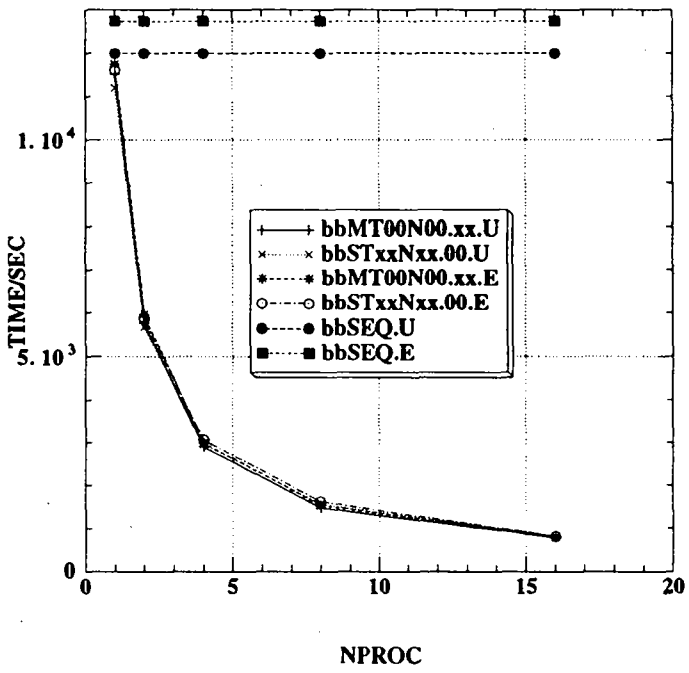

FIG. 14. *tU , tE Parallèle et Sequentiel: cas bb Slice & Mod Strategy*

Ayant présenté les résultats correspondant aux cas tests de 10 itérations, et ayant relevé l'influence de la granularité et de la stratégie utilisée, sur les performances obtenues, on s'intéresse maintenant à un cas de calcul plus réaliste (i.e. comprenant un nombre d'itérations suffisant pour obtenir la convergence vers la solution recherchée), afin de vérifier la stabilité des performances. Les résultats correspondant au cas de calcul réaliste (*bb*) sont présentés sur les Figures 14 pour

les temps absolus et 15, 16 pour les *Speed–Up*.

Au vu de la Figure 14, on note que la remarque 3.1 est bien vérifiée pour les deux stratégies de *TILE* considérées, que ce soit en temps *User* ou en temps *Elapsed*.

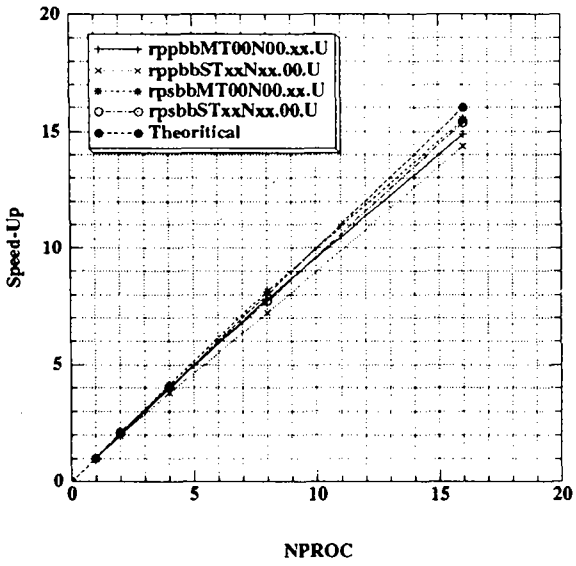

FIG. 15. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$ & $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$

cas bb : Slice & Mod Strategy

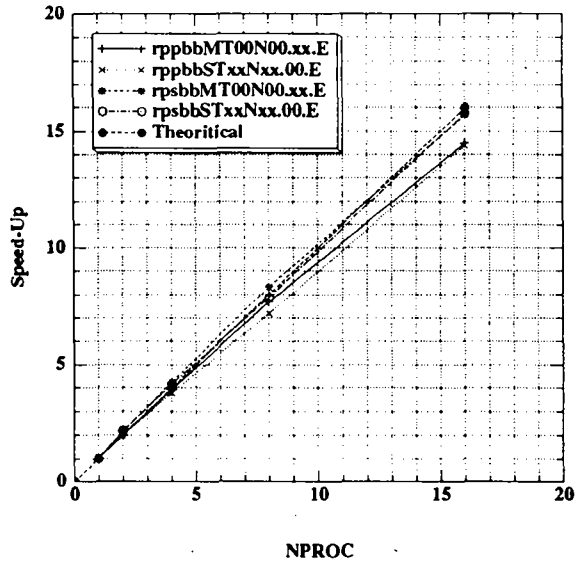

FIG. 16. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$ & $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$

cas bb : Slice & Mod Strategy

Ceci s'accompagne bien évidemment du meilleur *Speed-Up* correspondant, qui, sur 16 processeurs, est de l'ordre de 15.5 en ps et en temps *User* (Figure 15), et de 15.8 en ps et en temps *Elapsed* (Figure 16). On remarque par ailleurs un *Speed-Up* superlinéaire sur 2, 4 et 8 processeurs en ps (Figures 15 et 16), ceci est dû essentiellement à un overhead qui décroit lorsque le nombre de processeurs diminue.

Pour compléter l'analyse de chacun des cas (*a*), (*b*) et (*bb*), on présente maintenant des comparaisons entre les performances issues de ces cas tests (Figure 17 à 24). Tout d'abord, on s'intéresse aux *Speed–Up* de temps *User*, pp et ps, dans le cas d'utilisation des stratégies *MOD* (Figures 17 et 19) et *SLICE* (Figures 18 et 20). En ce qui concerne la stratégie *MOD*, on note que les performances sont semblables entre (*a*), (*b*) et (*bb*) dans le cas du *Speed–Up* pp, tandis que (*bb*) est meilleur que (*b*), lequel est meilleur que (*a*), dans le cas du *Speed-Up* ps. Pour ce qui est de l'utilisation de la stratégie *SLICE*, on observe globalement les mêmes performances que celles obtenues avec la stratégie *MOD*, mais on note cependant que le cas (*bb*) est moins performant que (*b*) pour le *Speed–Up* pp.

Ensuite, on s'intéresse aux *Speed–Up* de temps *Elapsed* (Figures 21 à 24); quelle que soit la façon de calculer le *Speed-Up*, i.e. pp ou ps, et quelle que soit la stratégie utilisée, le cas (*bb*) est meilleur que le cas (*b*) qui est meilleur que le cas (*a*), avec dans le cas ps, une performance quasi linéaire (15.8 sur 16 processeurs) pour le cas

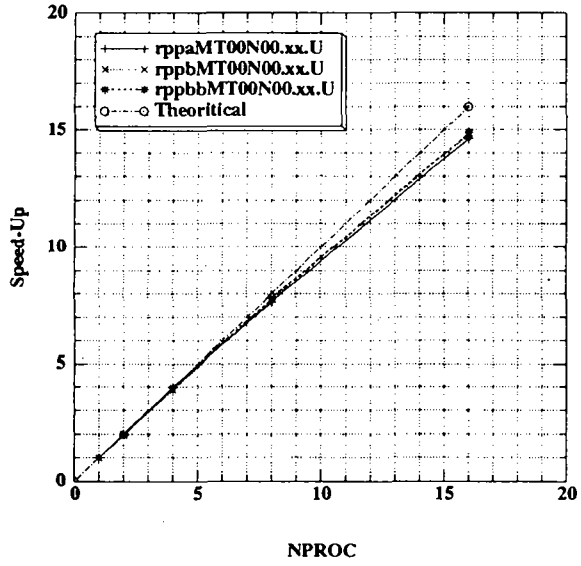

FIG. 17. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$

cas a, b & bb : Mod Strategy

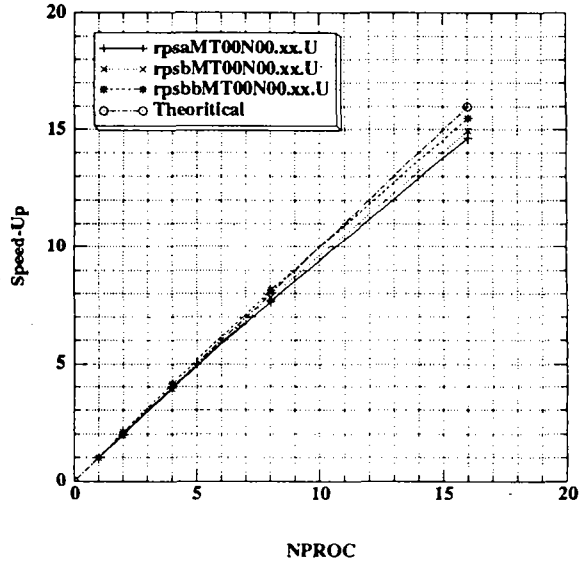

FIG. 18. $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$

cas a, b & bb : Mod Strategy

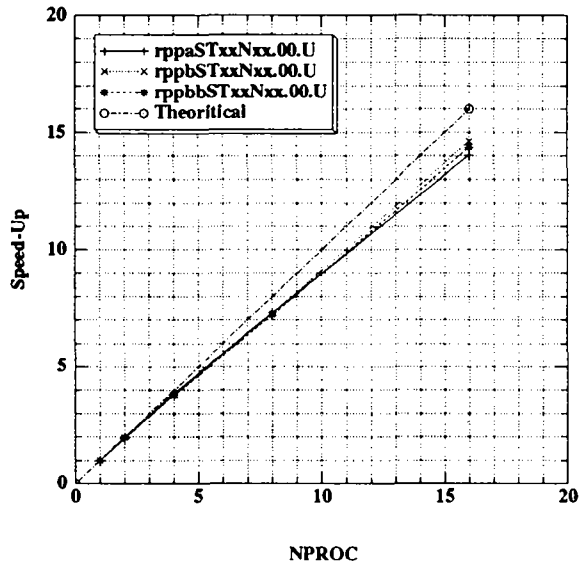

FIG. 19. $\frac{Paral.\ tU(1)}{Paral.\ tU(nproc)}$

cas a, b & bb : Slice Strategy

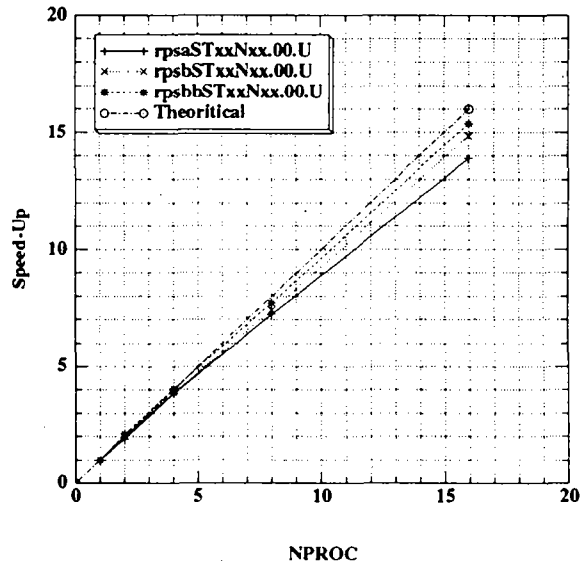

FIG. 20. $\frac{Seq.\ tU(1)}{Paral.\ tU(nproc)}$

cas a, b & bb : Slice Strategy

(*bb*), ce qui est loin d'être le cas pour (*a*) comme montré sur les Figures 22 et 24.

Enfin, on termine ces comparaisons (*a*) – (*b*) – (*bb*) par une présentation de l'Efficacité en fonction du nombre de processeurs. Dans les trois cas (*a*), (*b*) et (*bb*), et pour les deux stratégies *MOD* automatique (Figure 25) et *SLICE* (Figure

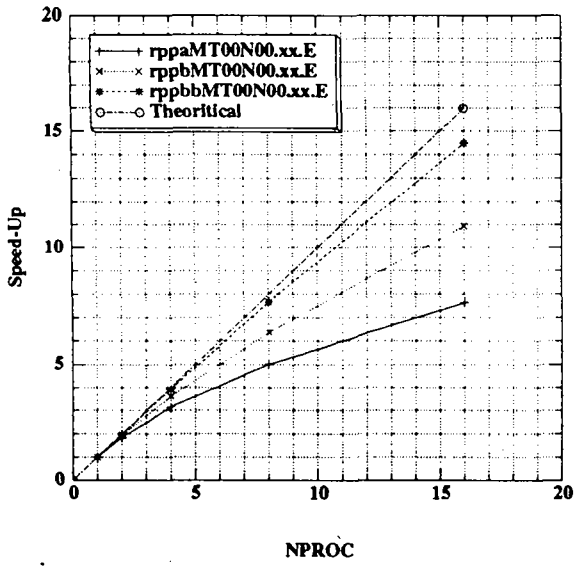

FIG. 21. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$

cas a, b & bb : Mod Strategy

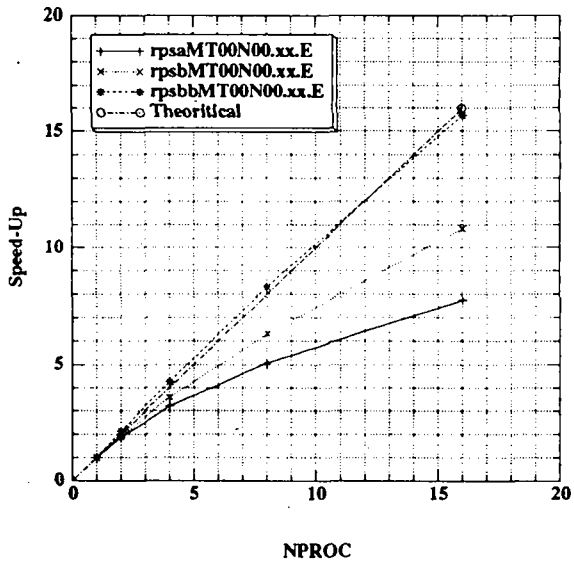

FIG. 22. $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$

cas a, b & bb : Mod Strategy

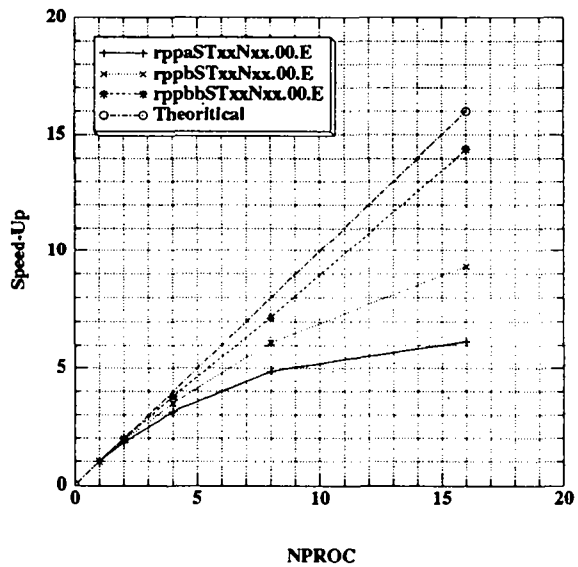

FIG. 23. $\frac{Paral.\ tE(1)}{Paral.\ tE(nproc)}$

cas a, b & bb : Slice Strategy

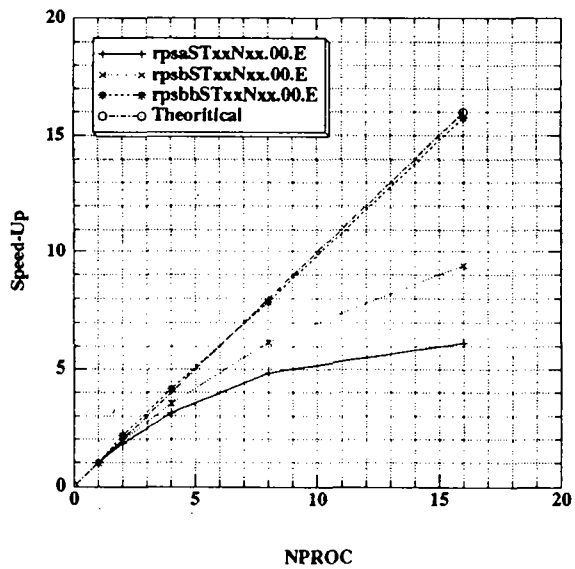

FIG. 24. $\frac{Seq.\ tE(1)}{Paral.\ tE(nproc)}$

cas a, b & bb : Slice Strategy

26), nous constatons que l'Efficacité du temps *User* par rapport au temps *Elapsed* est presque parfaite (95 à 98 %) pour le cas (*bb*), comme l'annonçait déjà la remarque 3.1. Sans anticiper sur la conclusion, cette analyse laisse prévoir d'ores et déjà que l'on pourrait garder de telles performances avec un nombre d'inconnues très grand aussi longtemps que l'on ne crée pas de défauts de Cache.

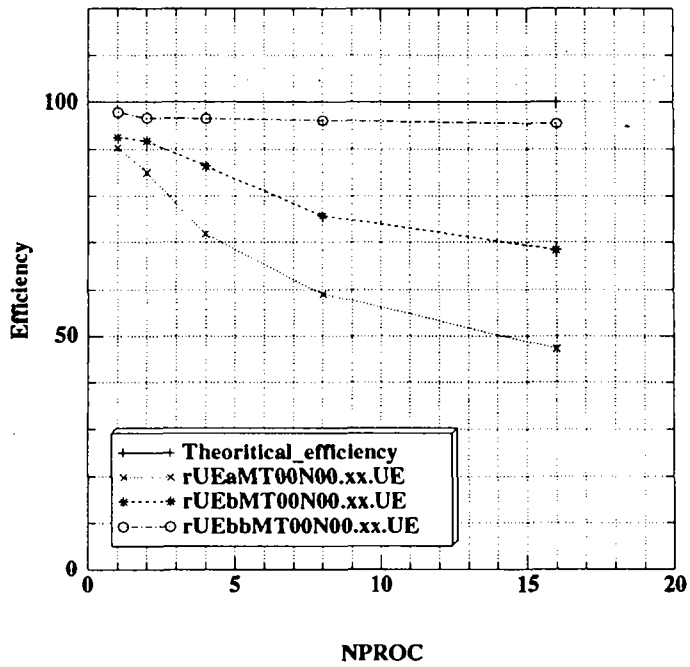

FIG. 25. $\frac{Paral.\ tU(nproc)}{Paral.\ tE(nproc)}$ *eff.*

cas a, b & bb : Mod Strategy

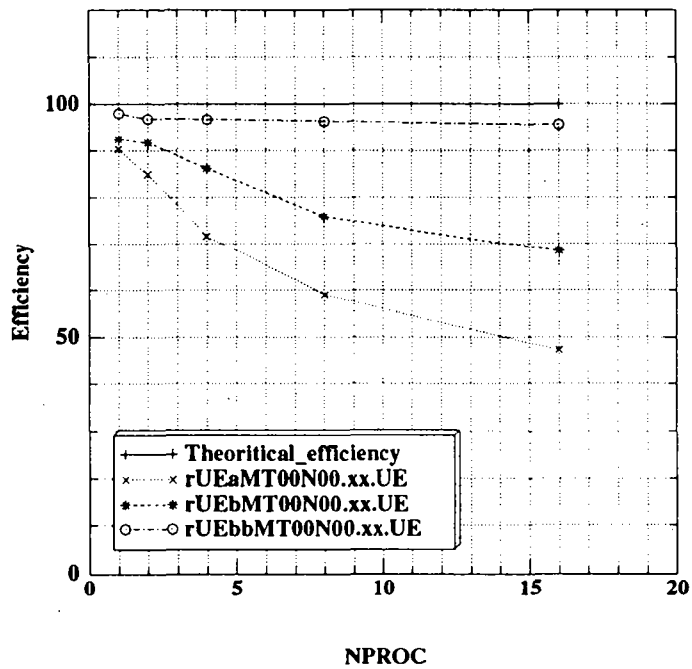

FIG. 26. $\frac{Paral.\ tU(nproc)}{Paral.\ tE(nproc)}$ *eff.*

cas a, b & bb : Slice Strategy

3.3 Résultats Numériques

Afin d'illustrer notre propos, nous présentons ci-après quelques résultats numériques calculés par le code et correspondant au cas test 8.2 du workshop "Hypersonic Flows for Reentry Problems"[22][23]. Ces résultats montrent les évolutions des grandeurs physiques telles que la pression, le nombre de Mach et les différentes températures dans la tuyère $TCM2$, obtenues avec une pression réservoir de 1530 bar et une température réservoir de 6500 K. Nous présentons d'abord sur la Figure 27 un agrandissement du maillage quadrangulaire courbe et structuré autour du col de la tuyère dont l'axe de symétrie mesure 1.12 m.

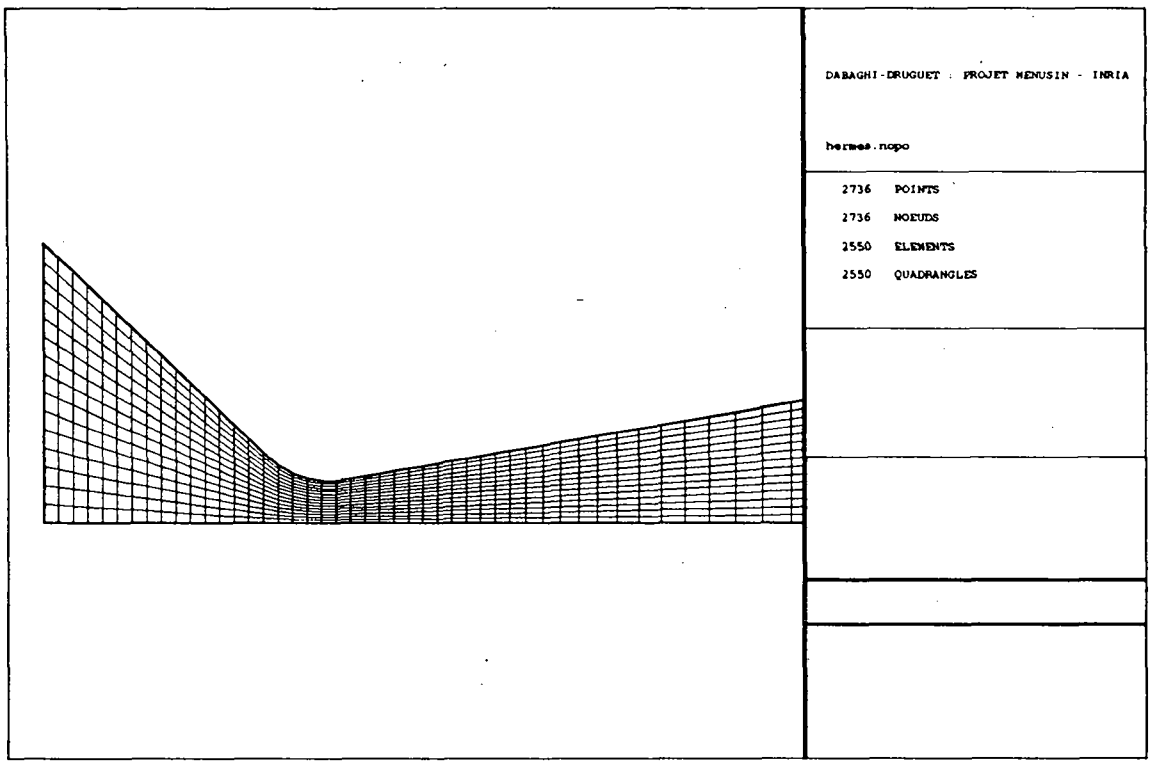

FIG. 27. *Agrandissement du maillage autour du col de la tuyère*

Sur les Figures 28A et 28B, nous avons tracé l'évolution du nombre de Mach, d'une part sous forme d'iso-contour et d'autre part sous forme d'une courbe de valeurs le long de l'axe de symétrie de la tuyère, où il atteint la valeur maximale de 11.2 dans la section de sortie. En ce qui concerne la pression, nous montrons sur les Figures 29A et 29B, son évolution qui passe de 1530 Bar dans la section d'entrée du convergent à quelques centaines de $Pascal$ en sortie de tuyère. Enfin sur les Figures 30A, 30B et 30C sont tracés les iso-contours des différentes températures et sur la Figure 30D leur valeur le long de l'axe de symétrie où l'on note le figeage des températures de vibration: partant de l'équilibre dans le convergent (6500 K), les températures vibrationnelles figent environ à 1600 K pour l'oxygène et à 2600 K pour l'azote.

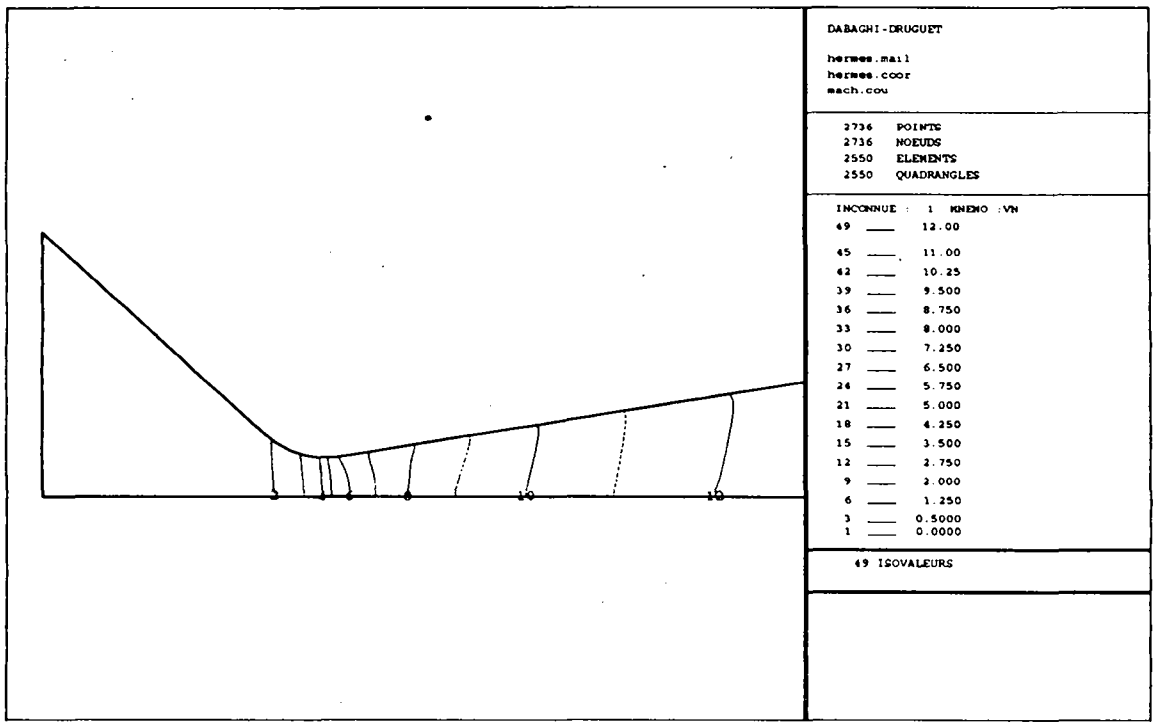

FIG. 28A. *Iso-Contours du nombre de Mach autour du col*

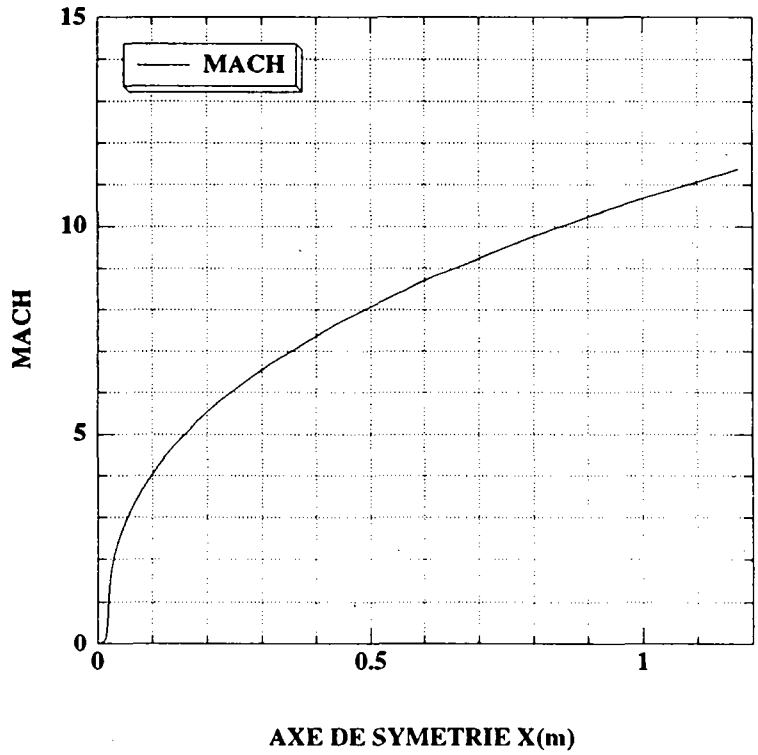

FIG. 28B. *Valeurs du nombre de Mach sur l'axe de symétrie*

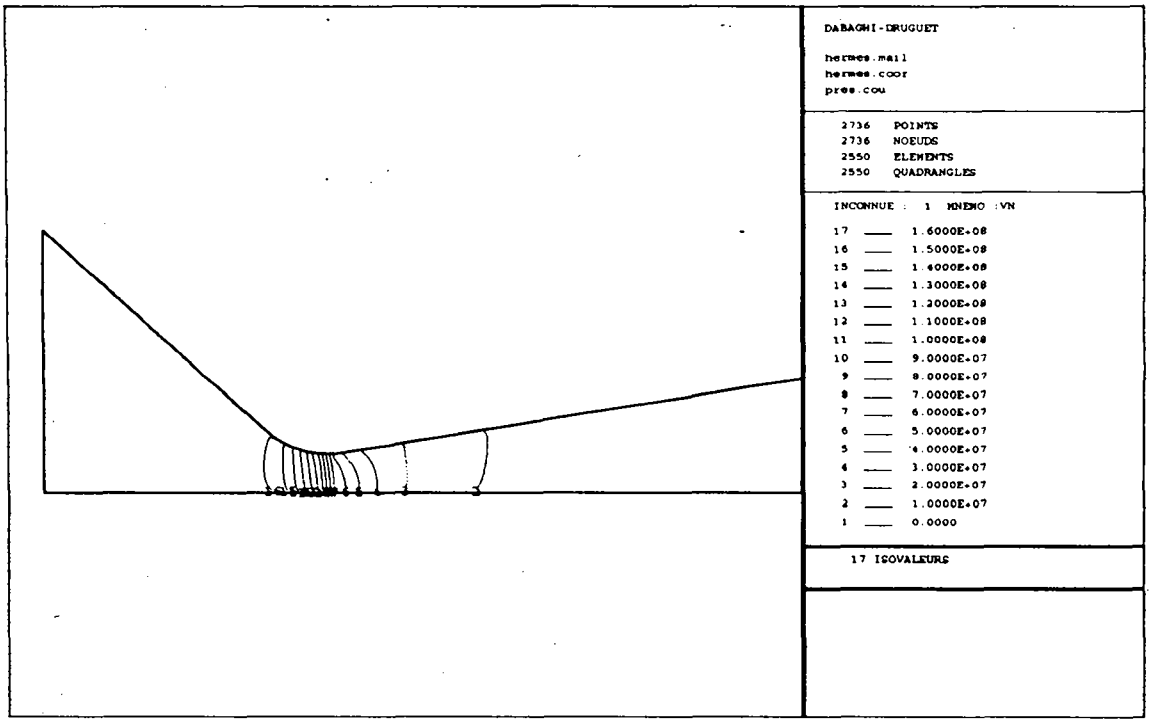

FIG. 29A. *Iso-Contours de la Pression autour du col*

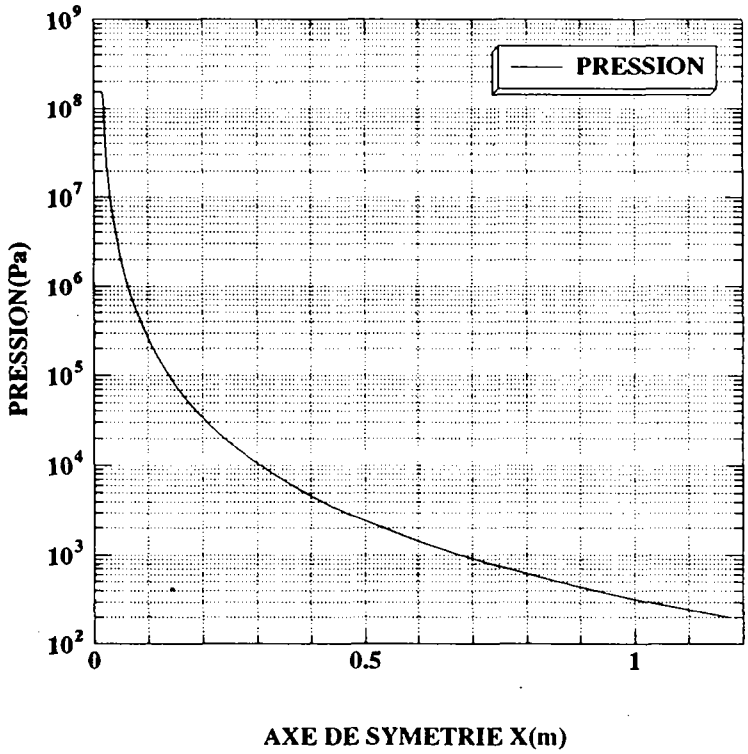

FIG. 29B. *Valeurs de la Pression sur l'axe de symétrie (éch. log.)*

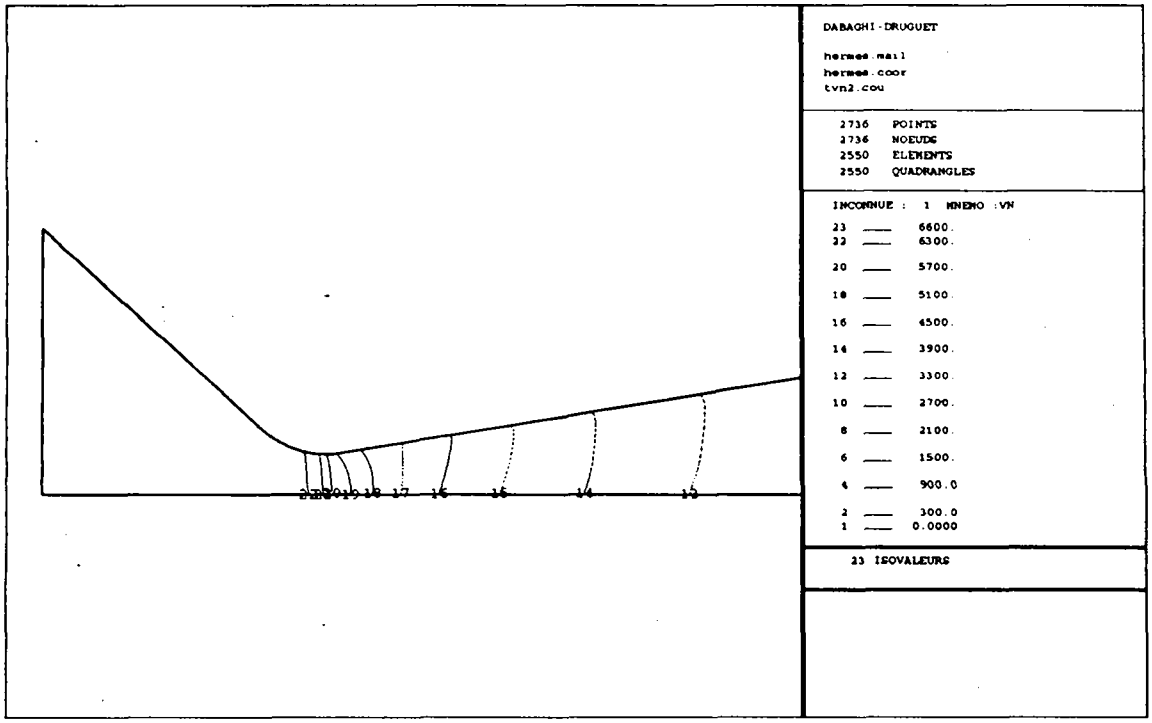

FIG. 30A. *Iso-Contours de la Température de Vibration de l'Azote N_2 autour du col*

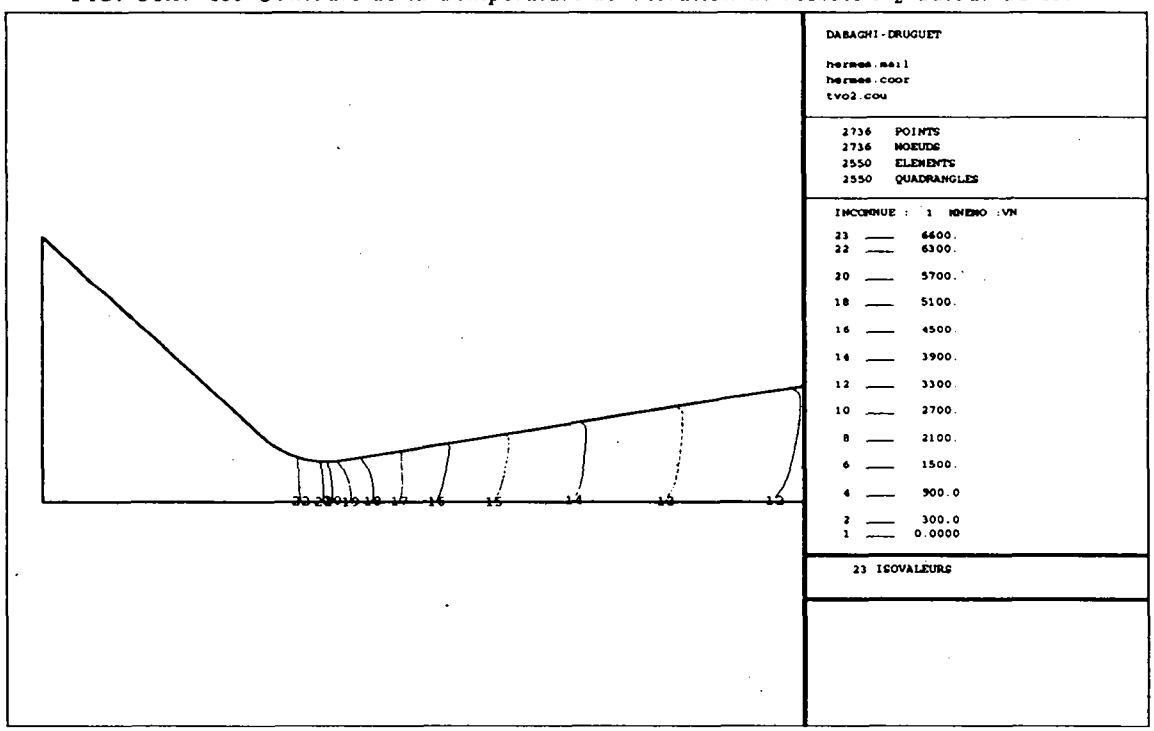

FIG. 30B. *Iso-Contours de la Température de Vibration de l'Oxygène O_2 autour du col*

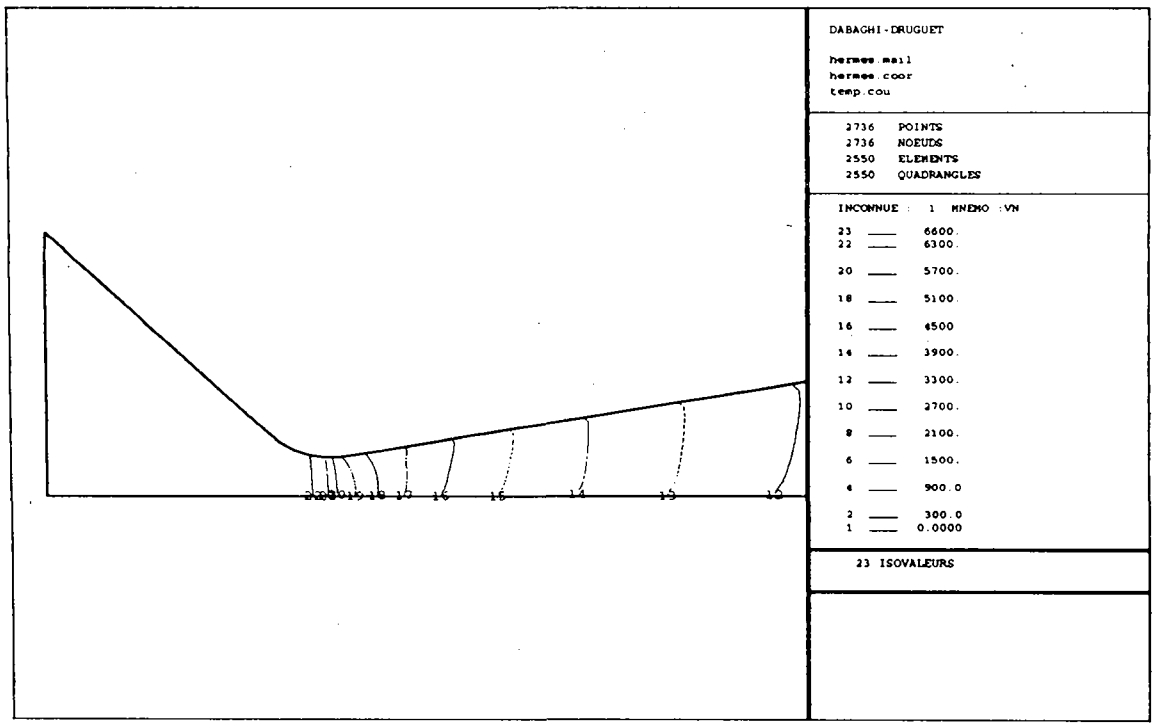

FIG. 30C. *Iso-Contours de la Température de l'écoulement autour du col*

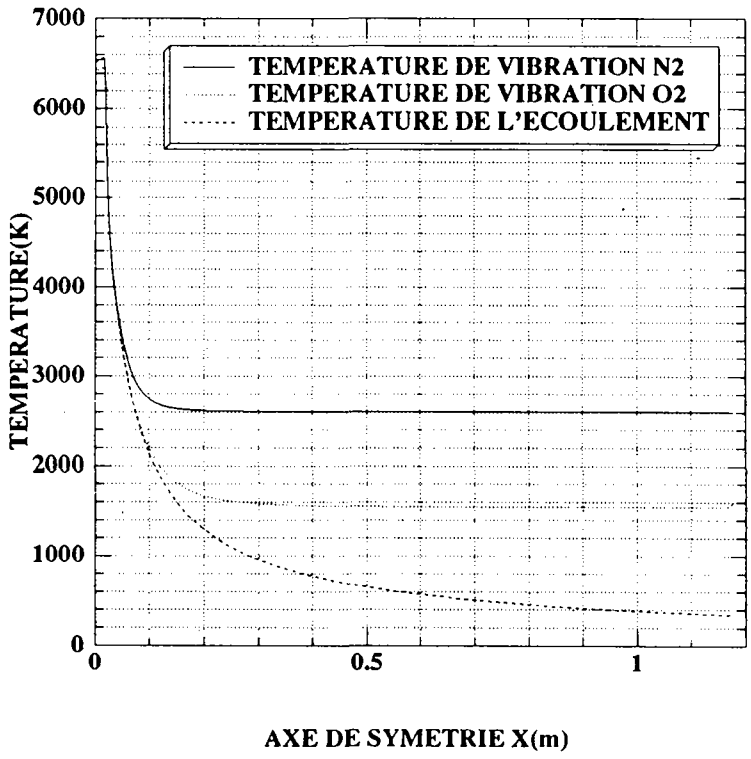

FIG. 30D. *Valeurs des Températures sur l'axe de symétrie*

4 Conclusions et Perspectives

Au vu des excellentes performances obtenues lors du traitement parallèle du code et présentées ici, il apparaît très intéressant de continuer nos investigations, notamment par le lancement en parallèle de cas tests de taille de plus en plus grande. Ainsi en gardant comme point de départ, le maillage à 171×16 nœuds, il serait bon de procéder à un raffinement systématique du maillage en divisant chaque cellule par 4 ce qui revient à multiplier approximativement le nombre de points par 4 : 341×31. Ceci peut être répété un certain nombre de fois jusqu'à obtenir un nombre de nœuds de l'ordre de 800.000:

$$3000 \longrightarrow 12.000 \longrightarrow 48.000 \longrightarrow 192.000 \longrightarrow 768.000$$

ce qui correspond avec les besoins du code en mémoire à 30 Mo environ par tile sur 1 processeur lors d'une exécution du programme sur 16 processeurs; cette mémoire requise étant de l'ordre de la taille du LocalCache. De telles tailles de problèmes permettraient de tester d'une part la KSR1 avec des applications de forte granularité et d'autre part sa limite de défaut de Cache afin de connaître les performances dans ce cas.

La parallélisation, naturelle sur N_k, est aussi possible sur N_j et éventuellement sur $N_j \times N_k$. Ce choix dépend de la taille du problème à résoudre et du nombre de processeurs disponibles. Notre objectif est donc de trouver à travers un paramétrage astucieux, un moyen qui permette le choix automatique du type de parallélisation (N_j , N_k , $N_j \times N_k$) et du nombre de processeurs tout en tenant compte de la localité des données et des conflits que peuvent entraîner ces différents types de parallélisation avec un recours éventuel aux instructions *SECTIONS* et *REGIONS*. Par ailleurs la parallélisation globale nous permettra d'envisager d'une part des géométries plus importantes comme mentionné plus haut et d'autre part des modèles plus complexes où l'on peut considérer un plus grand nombre de composants chimiques.

Remerciements :

Les auteurs voudraient remercier MM. les Professeurs R. Brun et D. Zeitoun, de l'Université de Provence-Marseille, Dr J. Périaux et Dr B. Stoufflet de Dassault Aviation, M. le Professeur P. Le Tallec de l'INRIA, et M. A. CLO de KSR-France pour avoir permis la réalisation de ce travail.

Références

[1] Y. Burtschell, *Performances, Dimensionnement et Simulation Numérique d'une Soufflerie Hypersonique à Choc Réfléchi à Piston Libre* ; Thèse de Doctorat, Université de Provence, Spécialité Energétique, Octobre 1990.

[2] M. C. Druguet, *Contribution à l'Etude des Ecoulements Eulériens Hypersoniques en Déséquilibre Thermo-Chimique* ; Thèse de Doctorat, Université de Provence, Spécialité Mécanique-Energétique, Novembre 1992.

[3] W. G. Vincenti and G. H. Kruger, *Introduction to Physical Gas Dynamics* ; John Wiley and Sons, Inc., New-York, 1967.
[4] R. Brun, *Transport et Relaxation dans les Ecoulements Gazeux*; Masson, 1986.
[5] C. E. Treanor and P. V. Marrone, *Effects of Dissociation on the Rate of Vibrational Relaxation*; The Physics of Fluids, 5 (9), September 1962.
[6] P. V. Marrone and C. E. Treanor, *Chemical Relaxation with Preferential Dissociation from Excited Vibrational Levels* ; The Physics of Fluids, 6 (9), September 1963.
[7] M. C. Druguet, D. Zeitoun and R. Brun, *High Performance Computing II* ; SHPC Oct. 1991, Ed. M. Durand and F. El Dabaghi, North-Holland, 1991.
[8] M. C. Druguet, D. Zeitoun and R. Brun, *Computing Methods in Applied Sciences and Engineering* ; Ed. R. Glowinski, Nova Science Publishers, New-York 1992.
[9] C. Park, *Problems of Rate Chemistry in the Flight Regimes of Aeroassisted Orbital Transfert Vehicles* ; AIAA Paper, 84-1730, 1984.
[10] J. S. Evans, C. J. Jr Schexnayder, and P. W. Huber, *Boundary-Layer Electron Profiles for Entry of a Blunt Body at High Altitude* ; Technical Report D-7332, NASA TN, July 1973.
[11] Landau and Teller, Phys. Z. Sowjet, 10 (34), 1936.
[12] Y. V. Stupochenko, S. A. Losev and A. I. Osipov, *Relaxation in Shock Waves* ; Springer-Verlag, New-York, 1967.
[13] A. Goudjo et J. A. Désidéri, *Un Schéma de Volumes-Finis Décentré pour la Résolution des Equations d'Euler en Axisymétrique* ; Rapport de Recherche INRIA 1005, 1989.
[14] B. Van Leer, *Flux Vector Splitting for the Euler Equations* ; Lecture Notes in Physics, 170:507-512, 1982.
[15] P. A. Gnoffo, *Hypersonic Flows over Biconics Using a Variable-Effective-Gamma, Parabolized-Navier-Stokes Code* ; AIAA Paper, 83-1666, 1983.
[16] T. H. Dunigan, *Kendall Square Multiprocessor: Early Experiences and Performances* ; Technical Report, Oak Ridge National Laboratory, Oak Ridge, TN, USA, 1992.
[17] Kendall Square Research, *KSR Parallel Programming* ; Waltham, MA, USA, Feb. 1992.
[18] ——, *KSR Customer Education: Porting Aplications to the KSR1* ; Waltham, MA, USA, Apr. 1992.
[19] ——, *KSR Technical Summary* ; Waltham, MA, USA, 1992.
[20] C.F. Baillie and A.E. Macdonald, *Porting the Well-Posed Topographical Meteorological Model to the KSR Parallel Supercomputer* ;
[21] S.R. Breit, C. Pangali and D.M. Zirl, *Technical Applications on the KSR1: High Performance and Ease of Use* ; Proceedings of Compcom'93, San Francisco, CA, USA, Feb. 1993, pp. 303-310.
[22] D. Zeitoun, P. Colas, P. Imbert and R. Brun, *Non Equilibrium Flow in a Hypersonic Wind Tunnel Nozzle*, Hypersonic Flows for Reentry Problems, Vol I et II, Springer-Verlag 1991.
[23] M. C. Druguet, D. Zeitoun and R. Brun, *Chemical and Vibrational Non Equilibrium Inviscid Hypersonic Nozzle Flow*, Hypersonic Flows for Reentry Problems, Vol III, Springer-Verlag 1992.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399