Orléans

CNRS CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

CALCUL PARALLÈLE D'ÉCOULEMENTS RÉACTIFS COMPRESSIBLES

Dmitry DAVIDENKO

Cpee

Plan de l'exposé

I. Introduction

II. Plates-formes parallèles et principes de parallélisation

III. Quelques tests comparatifs

IV. Simulations parallèles d'écoulements réactifs à l'ICARE-CNRS

V. Conclusions et perspectives

Plan de l'exposé

I. **Introduction**

II. Plates-formes parallèles et principes de parallélisation

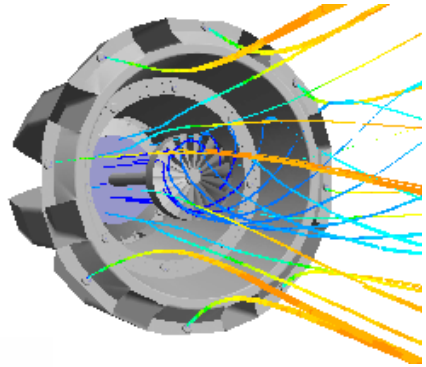
III. Quelques tests comparatifs

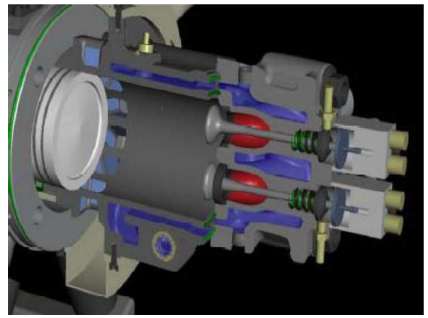
IV. Simulations parallèles d'écoulements réactifs à l'ICARE-CNRS

V. Conclusions et perspectives

Applications pratiques d'écoulements réactifs compressibles

- Moteurs aéronautiques
- Propulseurs des lanceurs spatiaux
- Moteurs d'automobile
- Turbines terrestres
- Lasers à gaz
- Appareils de rentrée atmosphérique





Approches utilisées dans les simulations

- **Milieu continu (Kn = l / L << 1)**
  - **Équations d'Euler pour le gaz non visqueux**
  - **Équations de Navier-Stokes pour le gaz visqueux**
  - **Méthodes mixtes Navier-Stokes - Monte Carlo pour les écoulements turbulents**

- **Milieu raréfie ou micro-échèles (Kn = l / L ≈ 1)**
  - **Modèle Bhatnagar-Gross-Krook (BGK) pour l'équation cinétique**
  - **Méthode « Direct Simulation Monte Carlo » (DSMC)**

Approche milieu continu : équations de Navier-Stokes

Continuité

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0$$

Quantité de mouvement

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}$$

Bilan énergétique

$$\frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u_j}{\partial x_j} = \frac{\partial u_i \tau_{ij}}{\partial x_j} - \frac{\partial q_j}{\partial x_j}$$

$$E = e + \tfrac{1}{2} u_i u_i$$

Bilan massique des espèces chimiques

$$\frac{\partial (\rho Y_s)}{\partial t} + \frac{\partial (\rho u_j Y_s)}{\partial x_j} = -\frac{\partial J_{sj}}{\partial x_j} + W_s$$

$$s \in [1, N_s - 1] \qquad \sum_{s=1}^{Ns} \tilde{Y}_s = 1$$

Termes visqueux

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

$$q_i = -\lambda \frac{\partial T}{\partial x_i} + \sum_{s=1}^{Ns} h_s J_s$$

Modèle de cinétique chimique

Réaction d'échange

$$\sum_s \alpha_s \mathrm{A}_s \Leftrightarrow \sum_s \beta_s \mathrm{A}_s$$

$$\omega_\mathrm{f} = K_\mathrm{f} \prod_s C_s^{\alpha'_s} \qquad \omega_\mathrm{b} = K_\mathrm{b} \prod_s C_s^{\beta'_s}$$

Réaction de dissociation-recombinaison

$$\sum_s \alpha_s \mathrm{A}_s + \mathrm{M} \Leftrightarrow \sum_s \beta_s \mathrm{A}_s + \mathrm{M}$$

$$\omega_\mathrm{f} = K_\mathrm{f} C_\mathrm{M} \prod_s C_s^{\alpha'_s} \qquad \omega_\mathrm{b} = K_\mathrm{b} C_\mathrm{M} \prod_s C_s^{\beta'_s}$$

Constante de vitesse d'une réaction chimique (K_f, K_b)

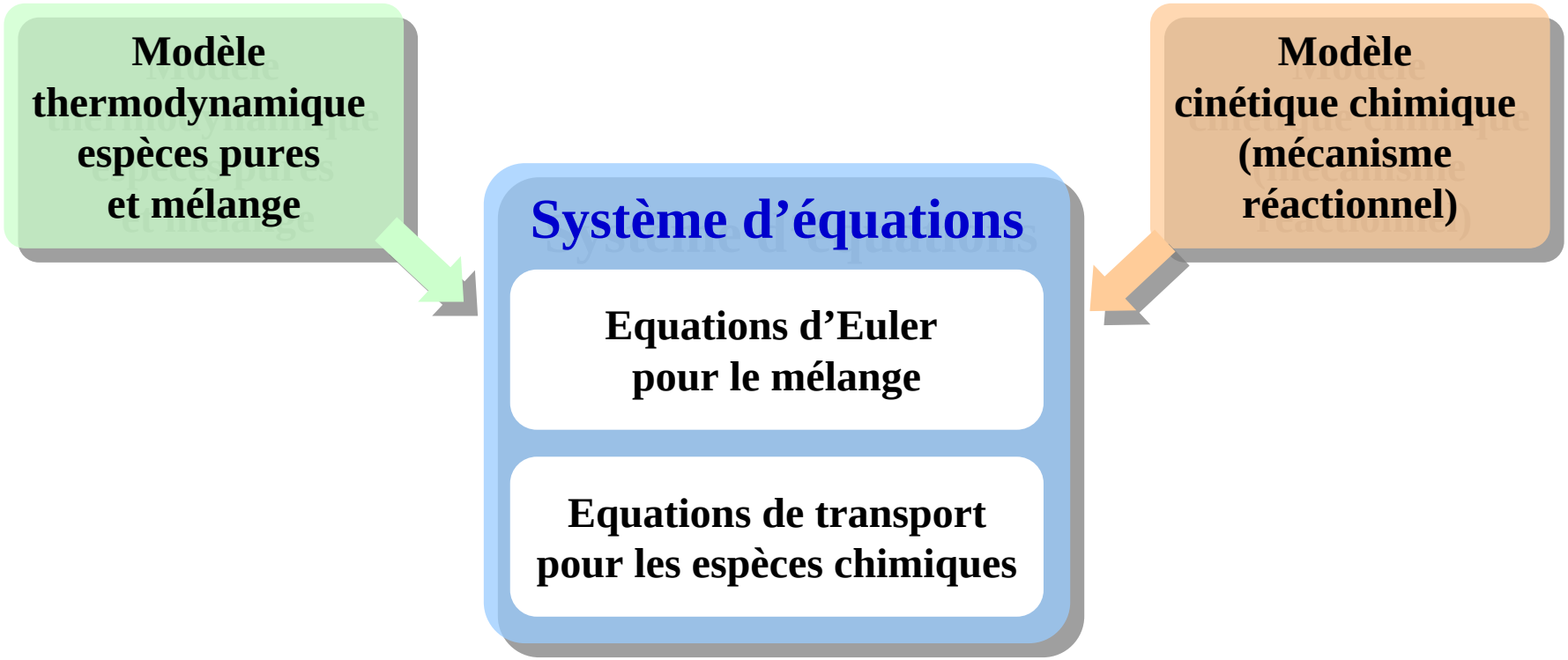
Loi d'Arrhenius

$$K = A T^b \exp\left(-\frac{E}{R_\mathrm{u} T} \right)$$

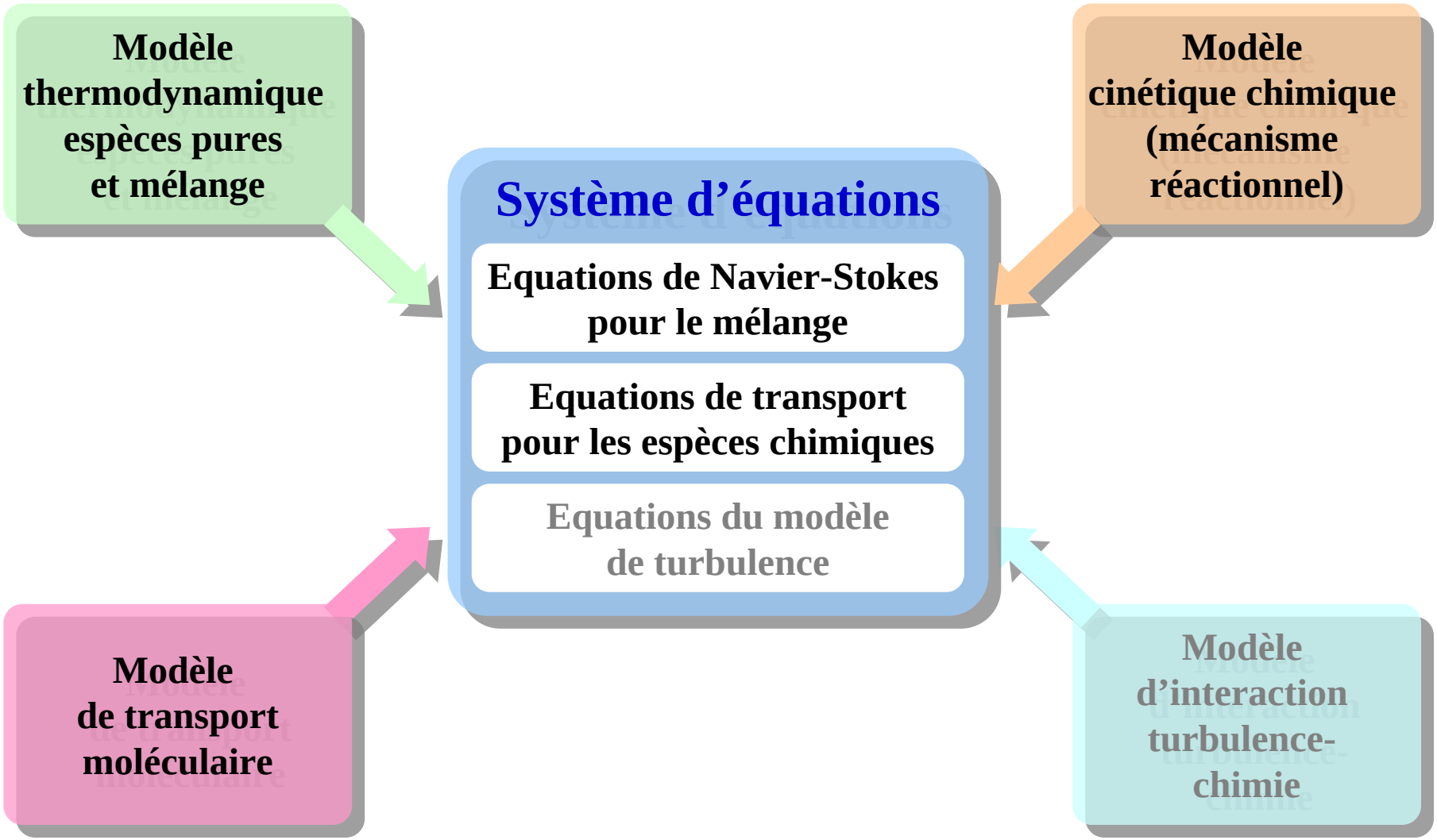
Taux de production d'une espèce chimique

$$W_s = M_s \sum_{r=1}^{N_r} \left(\beta_{sr} - \alpha_{sr} \right) \left(\omega_{\mathrm{f}\,r} - \omega_{\mathrm{b}\,r} \right)$$

Approche milieu continu non visqueux réactif



Approche milieu continu visqueux réactif



Facteurs augmentant la coût de calcul

- Configurations complexes
  - Géométrie 3D
  - Problème non stationnaire
- Conditions sévères
  - Haute pression
  - Mélanges réactifs concentrés
  - Systèmes réactionnels complexes
- Haute précision
  - Résolution spatiale et temporelle
  - Basse erreur numérique

Plan de l'exposé

I. Introduction

II. Plates-formes parallèles et principes de parallélisation

III. Quelques tests comparatifs

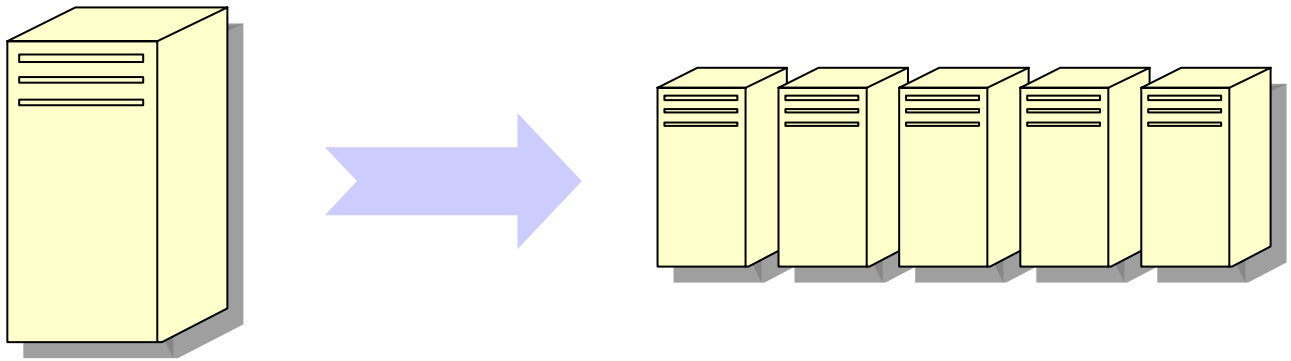
IV. Simulations parallèles d'écoulements réactifs à l'ICARE-CNRS

V. Conclusions et perspectives

Pourquoi calcul parallèle ?

- Le besoin de performance d'ordinateurs croît plus vite que leur puissance individuelle.
- Production développée des composants informatiques
  - Disponibilité
  - Prix modéré

Résolution du problème par la voie extensive

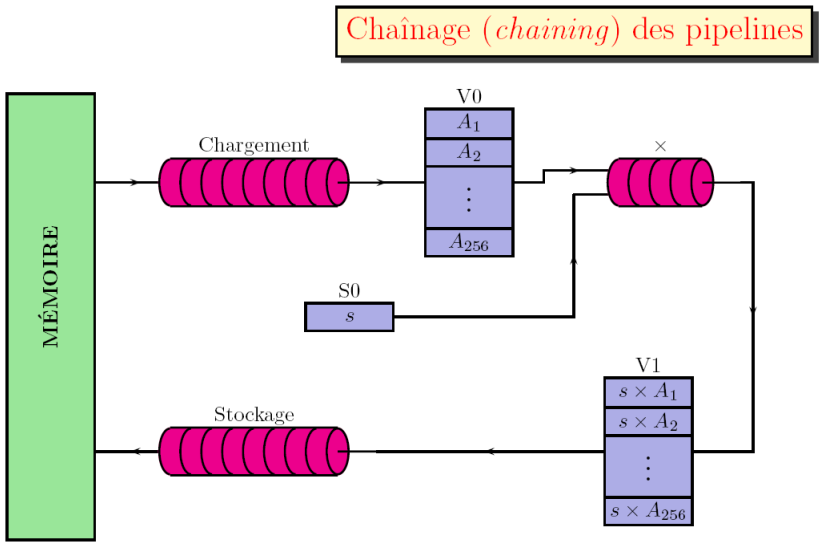


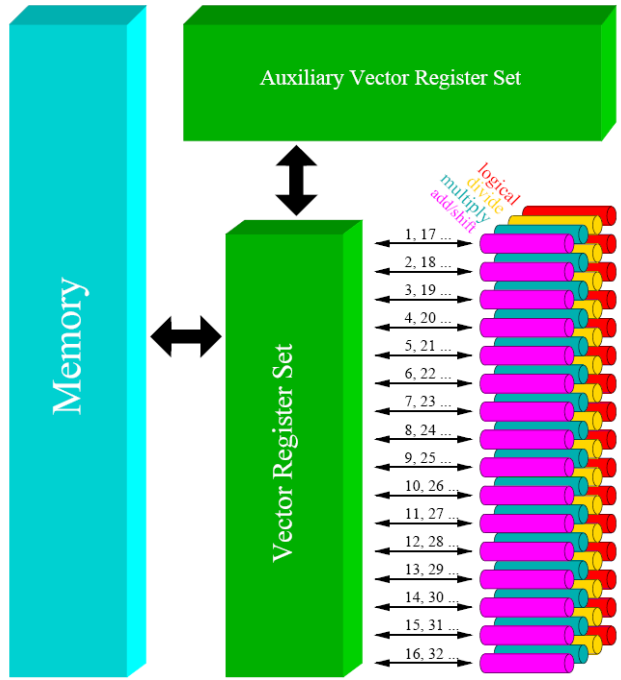
Architectures parallèles

- Vectorielle
- Parallèle à mémoire partagée
- Parallèle à mémoire distribuée

Architecture vectorielle

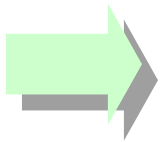
- **Principes**
  - Parallélisation au niveau de l'unité centrale
  - SIMD (Single Instruction – Multiple Data)
- **Matériel**
  - Pipelines dans l'unité centrale
  - Voies multiples d'accès à la mémoire



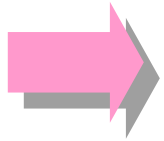


Vectorisation de code

- **Principes de parallélisme**
  - Exécution parallèle des opérateurs dans les boucles
  - Génération d'un code parallèle par le compilateur
- **Impact sur le codage**
  - La plus longue boucle à l'intérieur
  - Accès contigu aux données
  - Pas d'appel aux sous-routines dans les boucles
  - Pas d'operateur de sortie
  - Conflits de dépendance



```
do j = 1, 10
   do i = 1, 1000
      A(i,j) = B(i,j)
   end do
end do
```



```
do i = 1, Ni
   A(i–1) = B(i)
   C(i)   = A(i)
end do
```

≠

```
do i = 1, Ni
   A(i–1) = B(i)
end do
do i = 1, Ni
   C(i)   = A(i)
end do
```

Vectorisation : « + » et « – »

Gain de performance important sur les calculateurs vectoriels (famille NEC SX)

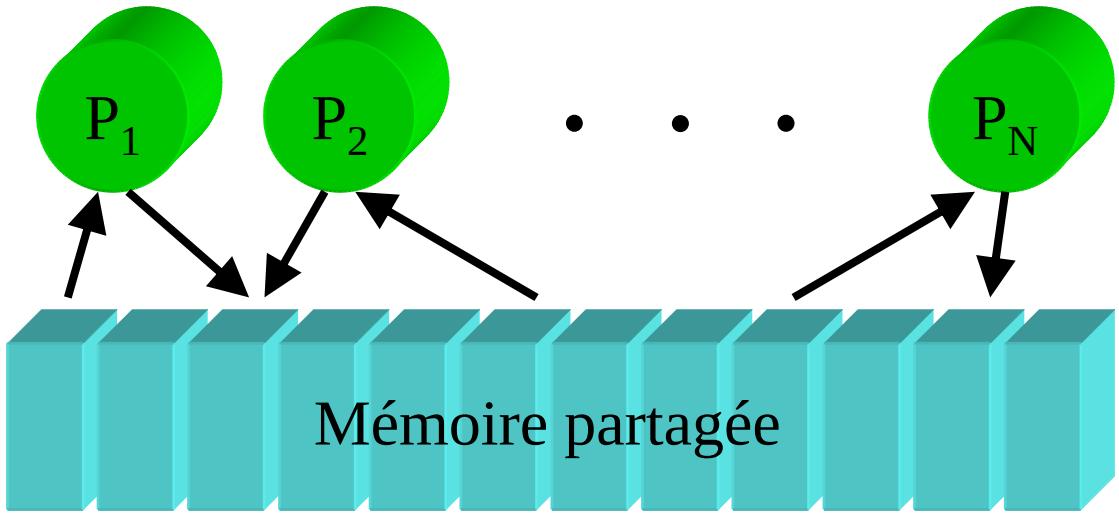
Disponible comme option d'optimisation sur les PC avec les compilateurs Intel ou PGI

Problèmes de codage et d'optimisation

Attention à la longueur des boucles et aux conflits d'accès à la mémoire

Architecture parallèle à mémoire partagée

- **Principes**
  - Partage d'un calcul entre plusieurs processeurs ayant accès indépendant à la mémoire commune
  - SIMD (Single Instruction – Multiple Data) et MIMD (Multiple Instructions – Multiple Data)
- **Matériel**
  - Multiprocesseurs
  - Voies multiples d'accès à la mémoire

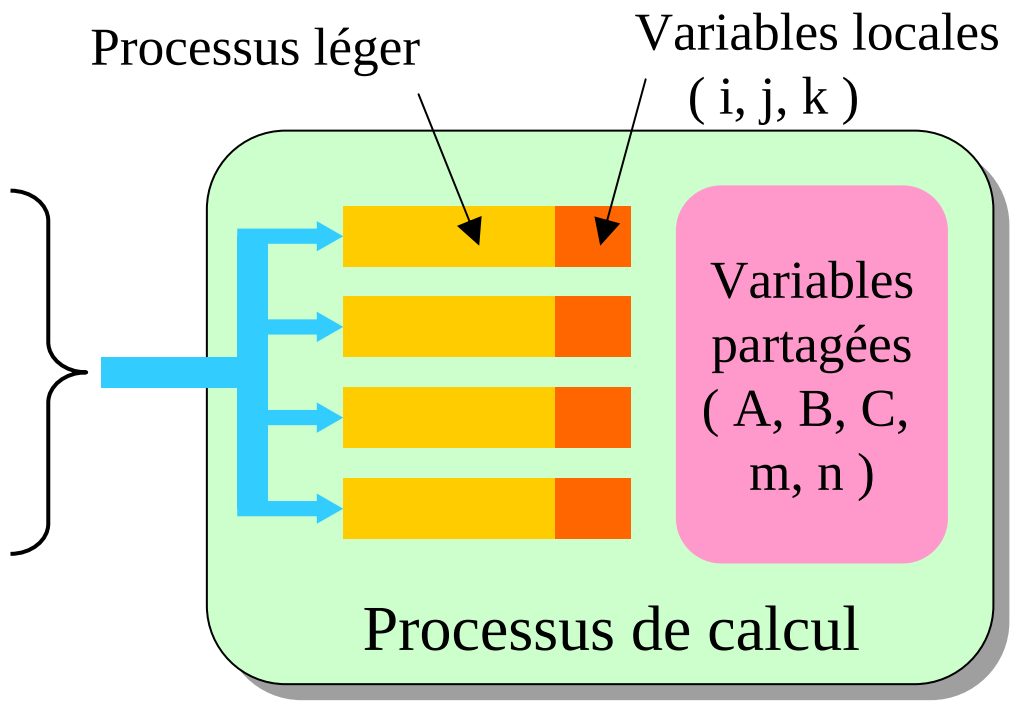


Code parallélisé OpenMP
(OpenMP = *Open Multi Processing*)

➲ Étapes de parallélisation

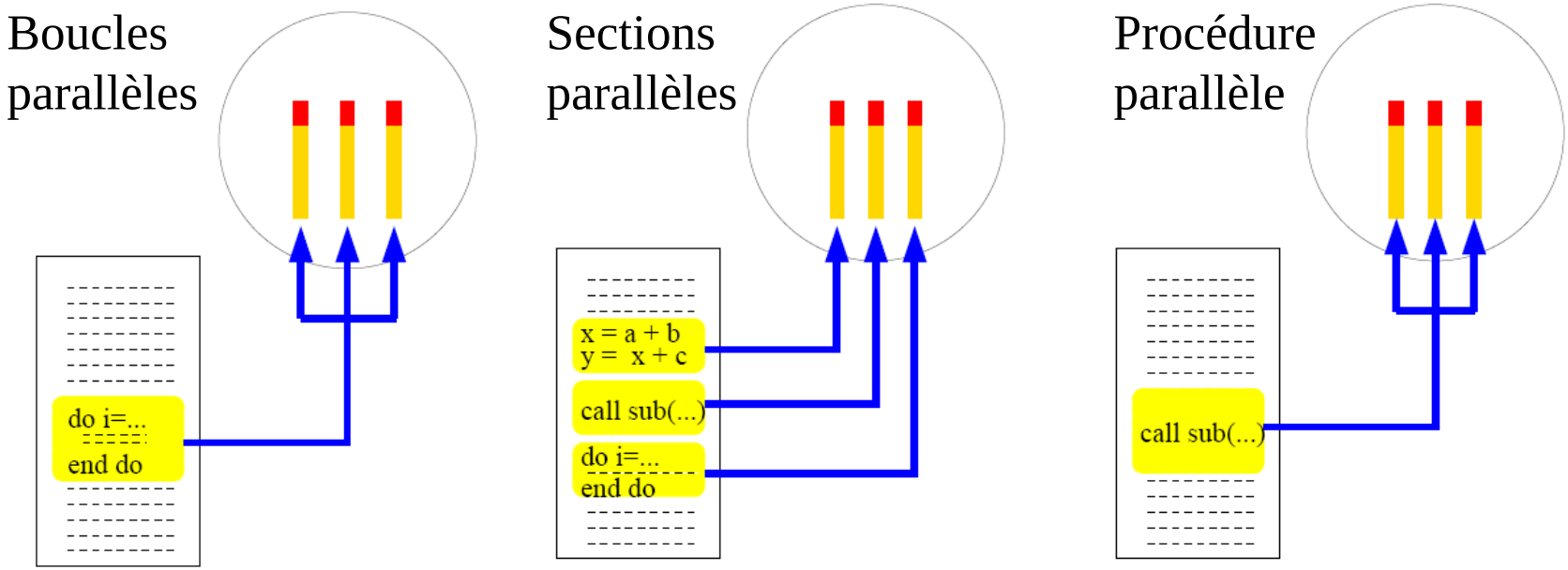
- **Indication des régions d'exécution parallèle du code par l'introduction d'instructions OpenMP**
- **Génération d'un code parallèle par le compilateur**
- **Exécution des régions parallèles par des processus légers (*threads*)**

```
! Produit de matrices
!$OMP PARALLEL
!$OMP DO SCHEDULE(RUNTIME)
do j = 1, m
   do k = 1, n
      do i = 1, m
         C(i,j) = C(i,j) + A(i,k) * B(k,j)
      end do
   end do
end do
!$OMP END DO
!$OMP END PARALLEL
```



Code parallélisé OpenMP (2)

- **Principes de parallélisme**
  - Répartir les itérations d'une boucle entre les processus légers
  - Exécuter différentes sections du code séparément
  - Exécuter plusieurs occurrences d'une procédure
- **Difficultés possibles**
  - Synchronisation des variables partagées lors de l'écriture
  - Utilisation efficace du cash pour éviter les conflits d'accès à la mémoire

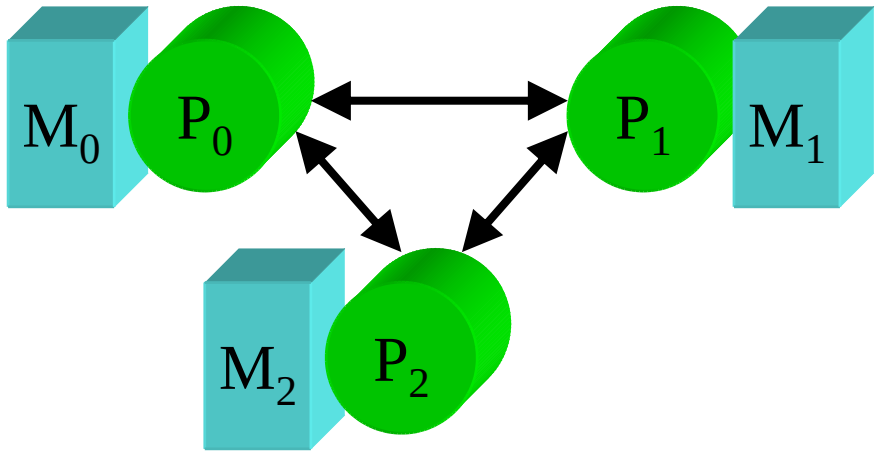


Parallélisation OpenMP : « + » et « – »

- **Gain de performance important sur les calculateurs multiprocesseurs**
- **Code portable parallèle-séquentiel**
- **Disponible sur les PC avec les compilateurs Intel ou PGI**
- **Problèmes d'optimisation et de débogage**
- **Attention aux conflits d'accès à la mémoire**

Architecture parallèle à mémoire distribuée

- **Principes**
  - Distribution des tâches entre plusieurs processus ayant des espaces mémoire réservés
  - Chaque processeur s'occupe de son processus
- **Matériel**
  - Multiprocesseurs
  - Clusters d'ordinateurs liés par un réseau



Code parallélisé MPI
(MPI = *Message Passing Interface*)

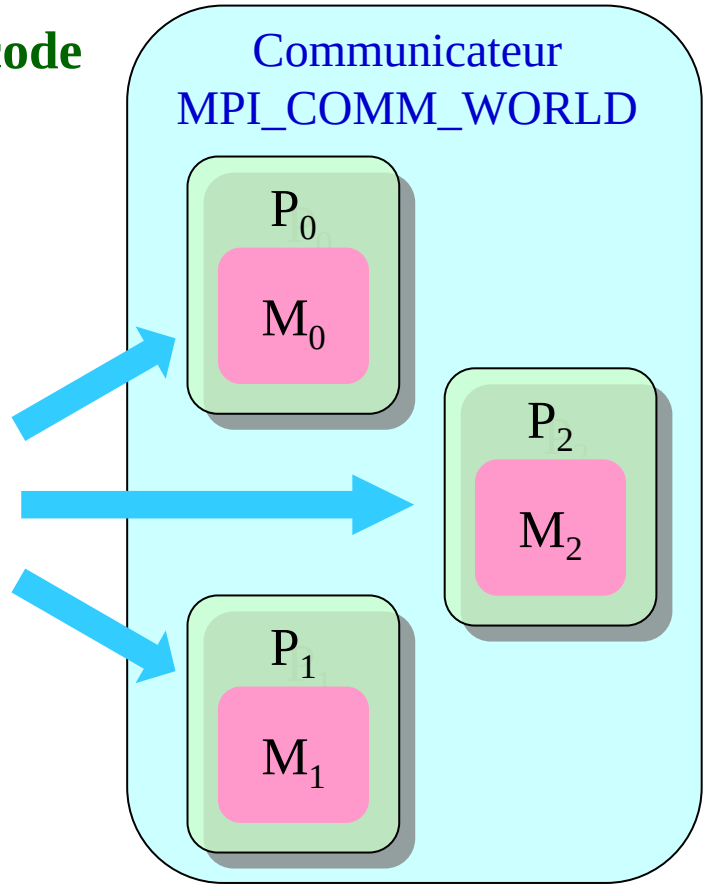
- **Étapes de parallélisation**
  - **Insertion des appels aux procédures MPI dans le code**
  - **Génération d'un code parallèle par le compilateur en utilisant des bibliothèques MPI**
  - **Exécution de plusieurs instances du code dans un environnement parallèle MPICH ou LAM**

```
include 'mpif.h'
integer :: nb_procs, rang, code

call MPI_INIT(code)
call MPI_COMM_SIZE(MPI_COMM_WORLD, nb_procs, code)
call MPI_COMM_RANK(MPI_COMM_WORLD, rang, code)

print, nb_procs, rang

call MPI_FINALIZE(code)
```



Code parallélisé MPI (2)

- **Principes de partage du travail entre les processus**
  - **Exécutions multiples d'une même tâche avec des entrées variées puis obtention de statistiques.**
  - **Décomposition du domaine de calcul en plusieurs zones et partage des zones entre les processus.**
    Les processus font des échanges de données relatives aux frontières communes et à l'avancement du calcul.
  - **Décomposition d'un problème complexe en sous-problèmes décrits par des modèles spécifiques et partage des sous-problèmes entre les processus.**
    Les processus font des échanges de données relatives au couplage des sous-problèmes.

Parallélisation MPI : « + » et « – »

- **Gain de performance important sur les calculateurs multiprocesseurs et sur les clusters**
- **Disponible sur toutes les plates-formes y compris des systèmes hétérogènes**
- **Code spécialisé parallèle**
- **Problèmes d'optimisation et de débogage**
- **Attention à la latence réseau**

IDRIS – Institut du Développement et des Ressources en Informatique Scientifique

- **Centre de calcul**
- **Formation sur la programmation, l'optimisation et la parallélisation de codes**
- **Aide au développement d'outils informatiques haute performance**

10 nœuds NEC SX8

Clusters IBM Power4 1024 proc

www.idris.fr

Plan de l'exposé

I. Introduction

II. Plates-formes parallèles et principes de parallélisation

III. Quelques tests comparatifs

IV. Simulations parallèles d'écoulements réactifs à l'ICARE-CNRS

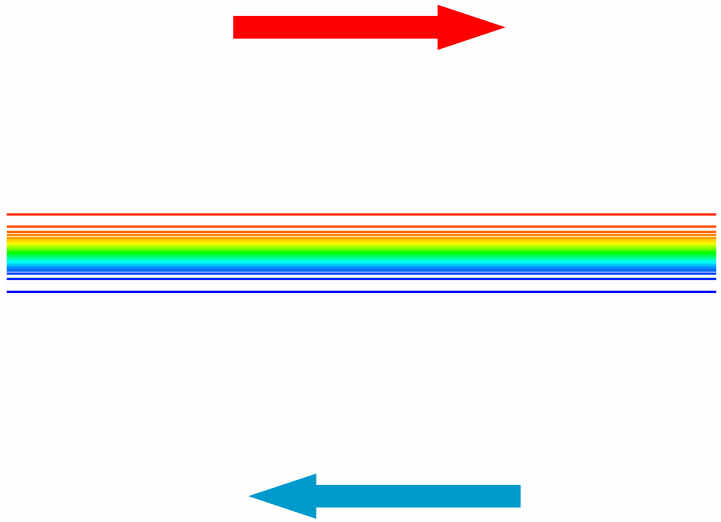
V. Conclusions et perspectives

Machines de calcul testées

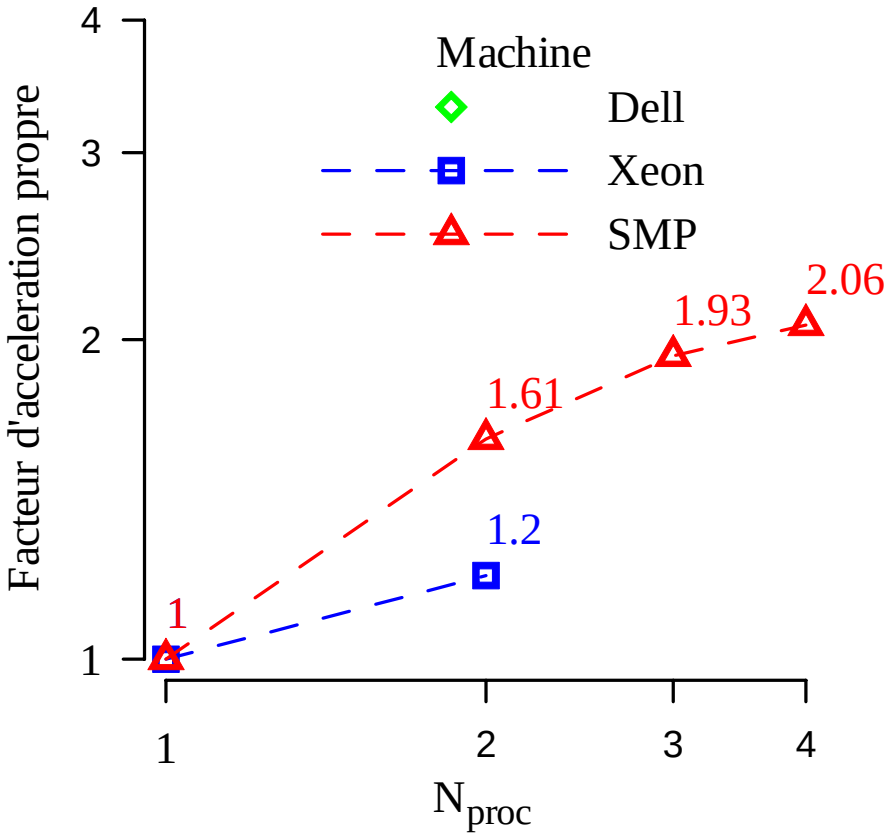
- PC de bureau Dell sous Windows (« Dell »)
  - mono-processeur P4 à 3 GHz
  - 1 G de mémoire à 800 MHz
- Cluster de PC sous Linux Red Hat 32 bits (« Xeon »)
  - 2 processeurs P4 Xeon à 2,8 GHz par nœud
  - 1 G de mémoire à 533 MHz par nœud
- Machine SMP sous Linux Red Hat 64 bits (« SMP »)
  - 2 processeurs double cœur AMD Opteron 280 à 2.4 GHz
  - 16 G de mémoire à 1 GHz

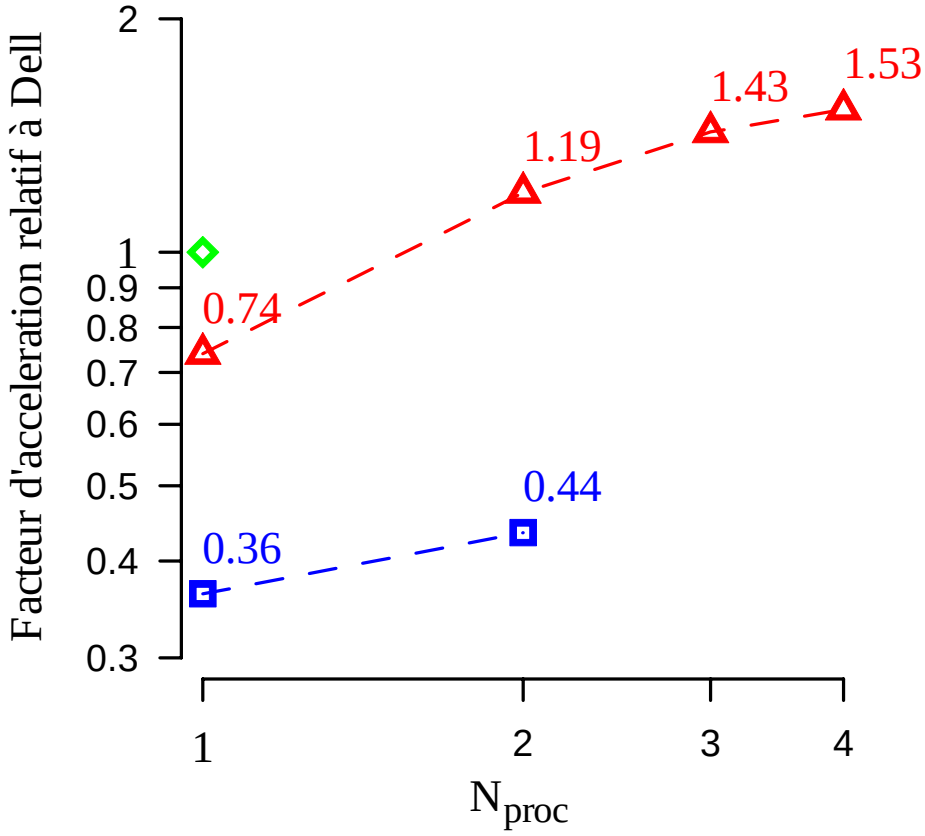
Test 1 : Calculs scalaires

- **Méthode**
  - **Pseudo-parallélisation :**
    exécution simultanée de plusieurs codes séquentiels
  - **Lancer autant de calculs que processeurs**
- **Code**
  - **Résolution des équations Navier-Stokes non stationnaires en 2D**
  - **Adaptation à la plate-forme vectorielle (longues boucles)**
    ⇒ accès intenses à la mémoire
- **Problème simulé**
  - **Couche de mélange 2D entre**
    2 écoulements à contresens
  - **2 espèces non réactives**
  - **Maillage 256 x 256 cellules**



Test 1 : Résultats



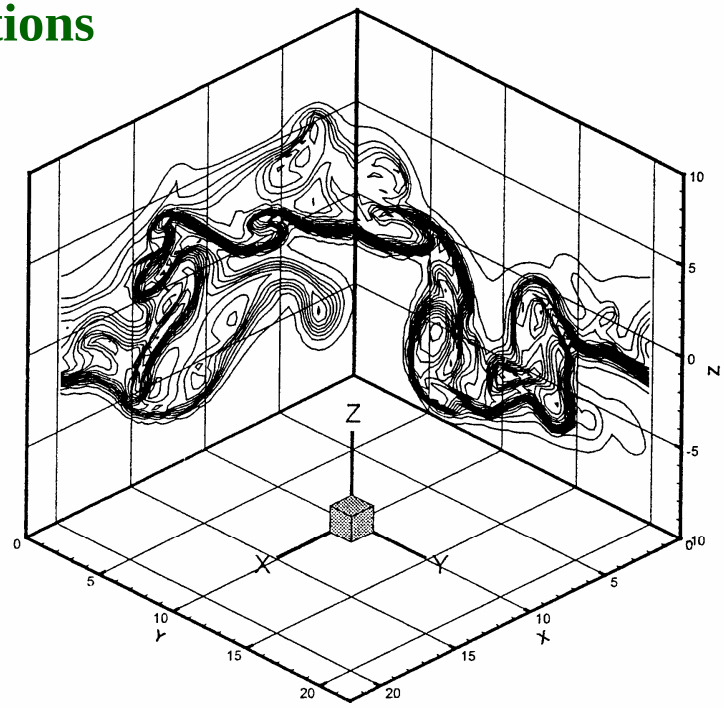


Conclusions

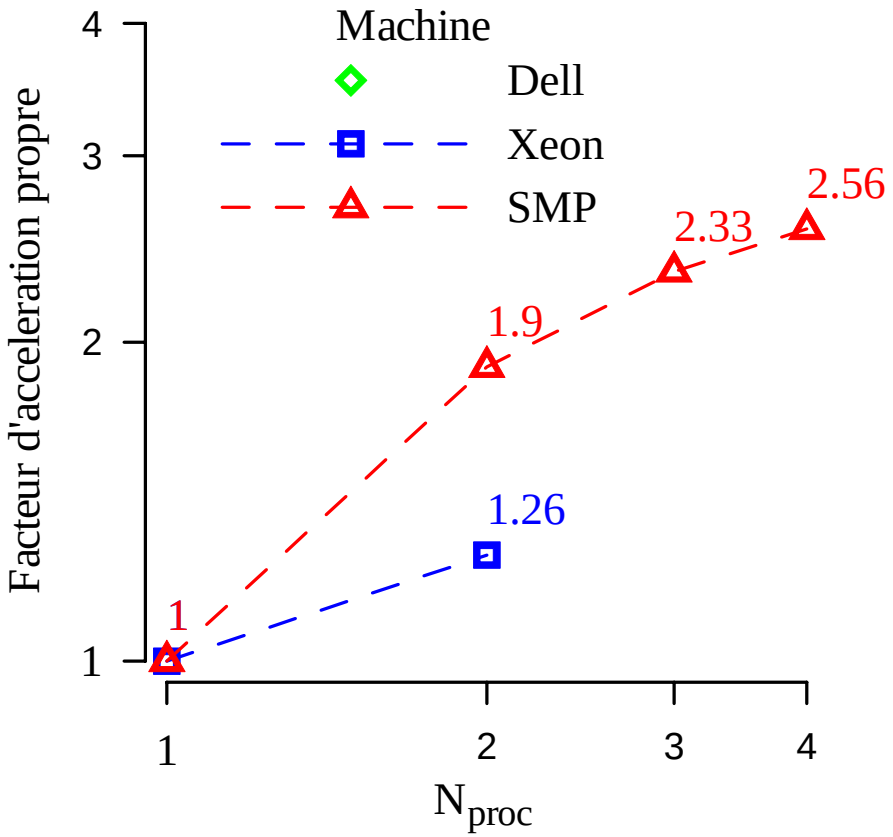
- Conflits fréquents d'accès à la mémoire ne permettent pas d'atteindre une accélération proportionnelle au nombre de processeurs
- Accélération est plus faible si la mémoire est lente

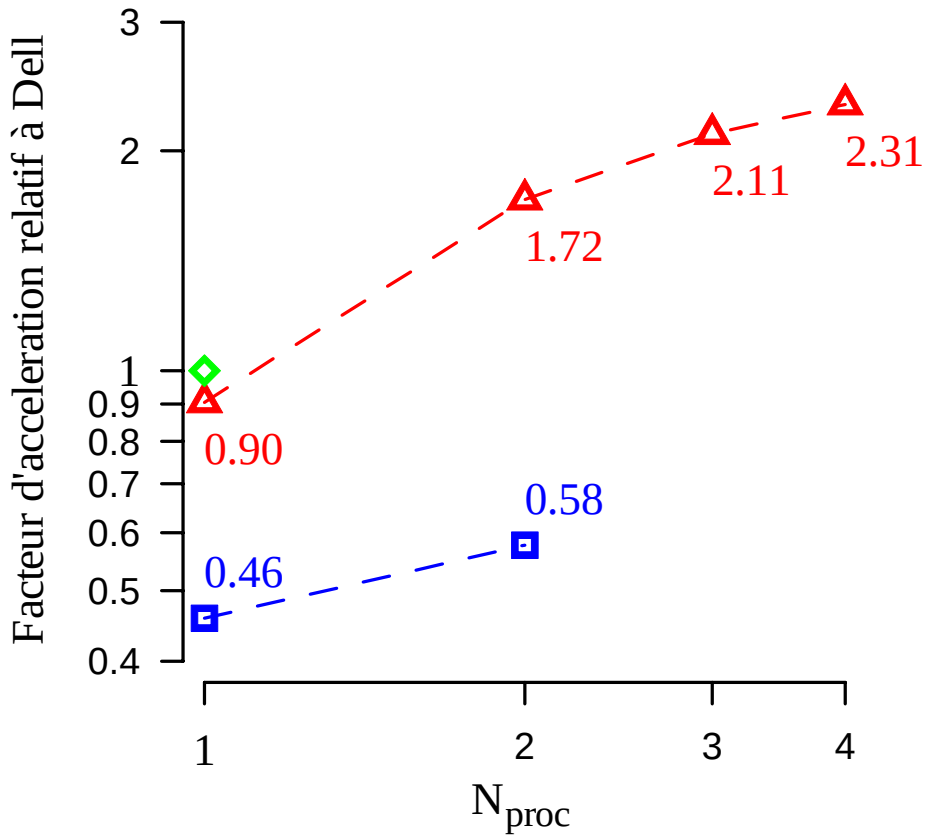
Test 2 : Parallélisation OpenMP

- **Méthode**
  - 1 calcul par machine
  - Nombre variable de processus légers
- **Code**
  - Résolution des équations Navier-Stokes non stationnaires en 3D
  - Adaptation à la plate-forme vectorielle et multiprocesseurs
  - Parallélisation de boucles et de sections
- **Problème simulé**
  - Couche de mélange 3D entre 2 écoulements à contresens
  - 2 espèces non réactives
  - Maillage 36 x 36 x 60 cellules



Test 2 : Résultats





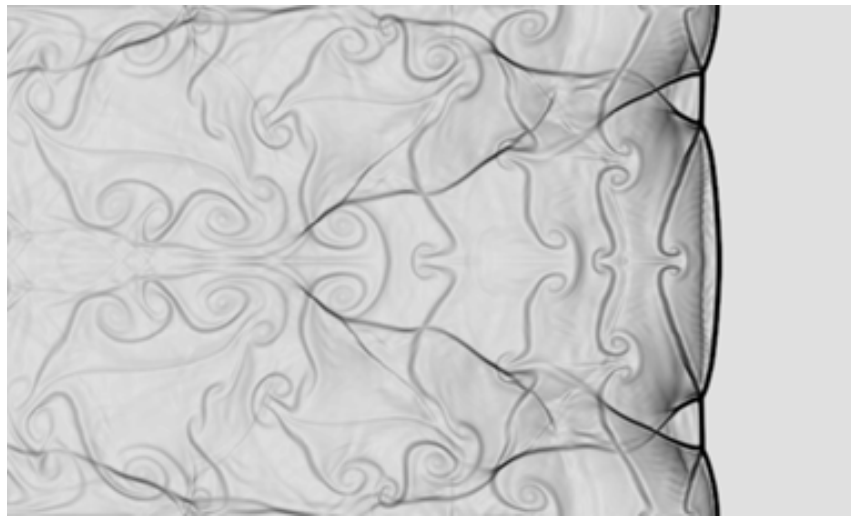
Conclusions

- **Bonne accélération sur 2 processeurs pour SMP mais peu de gain sur 3 et 4 processeurs (conflits fréquents d'accès à la mémoire)**
- **Faible accélération si la mémoire est lente**

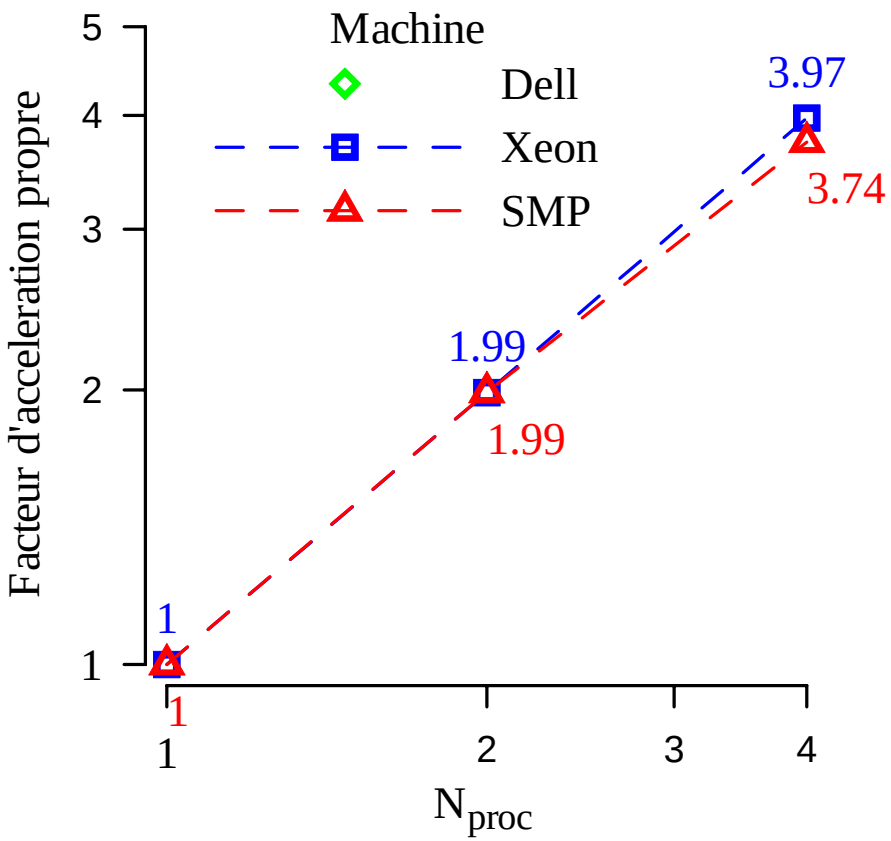
Test 3 : Parallélisation MPI

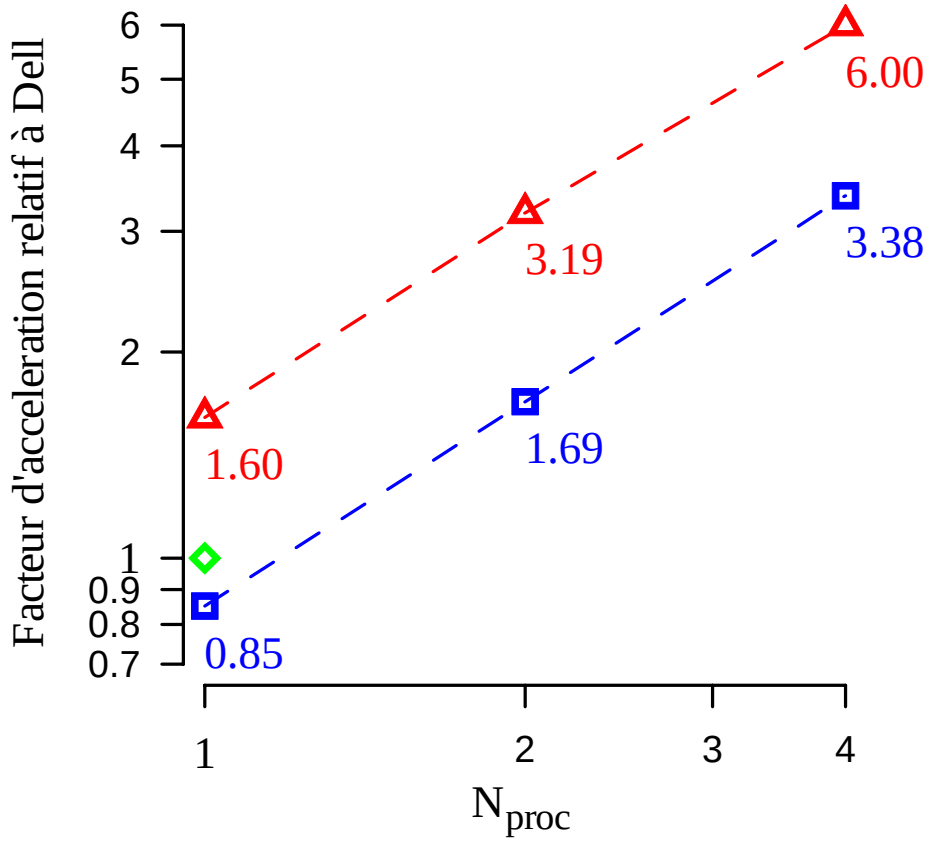
- **Méthode**
  - 1 calcul sur une ou plusieurs machines
  - Nombre variable de processus parallèles
- **Code**
  - Résolution des équations Euler non stationnaires en 2D
  - Parallélisation par décomposition du domaine de calcul
- **Problème simulé**
  - Détonation cellulaire 2D dans un mélange H_2 / O_2 / Ar
  - 7 réactions entre 7 espèces
  - Maillage 900 x 400 cellules

sens de propagation



Test 3 : Résultats





Conclusion

- **Accélération proportionnelle au nombre de processus parallèles**
- **Bonnes performances pour la machine SMP et le cluster Xeon**

Plan de l'exposé

I. Introduction

II. Plates-formes parallèles et principes de parallélisation

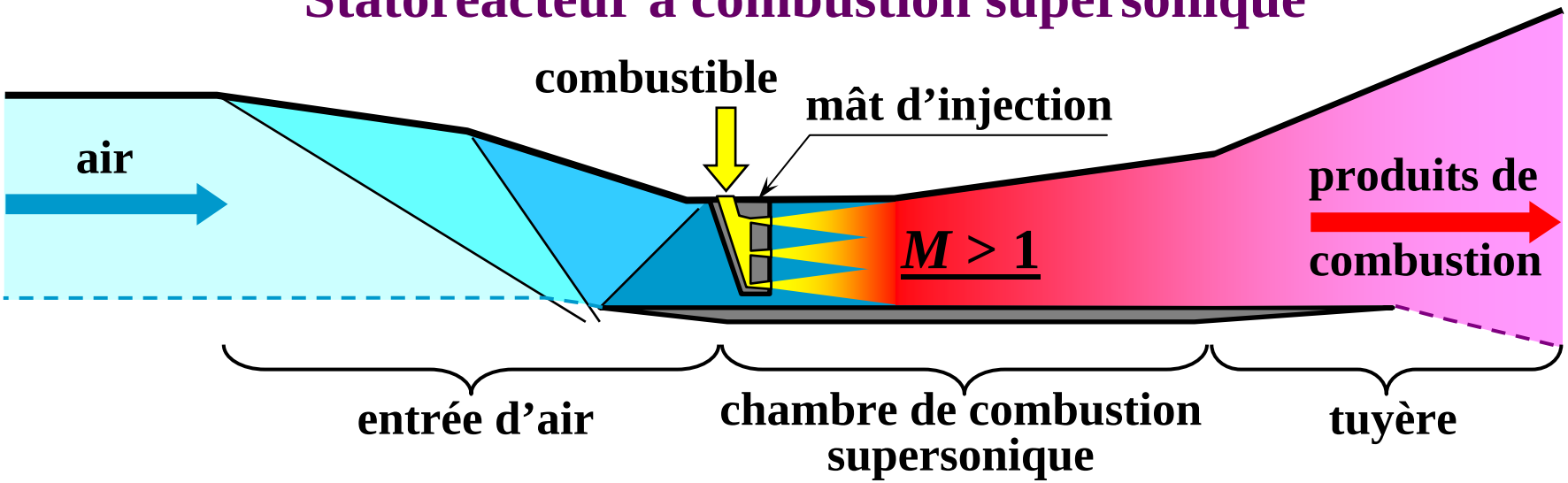
III. Quelques tests comparatifs

IV. Simulations parallèles d'écoulements réactifs à l'ICARE-CNRS

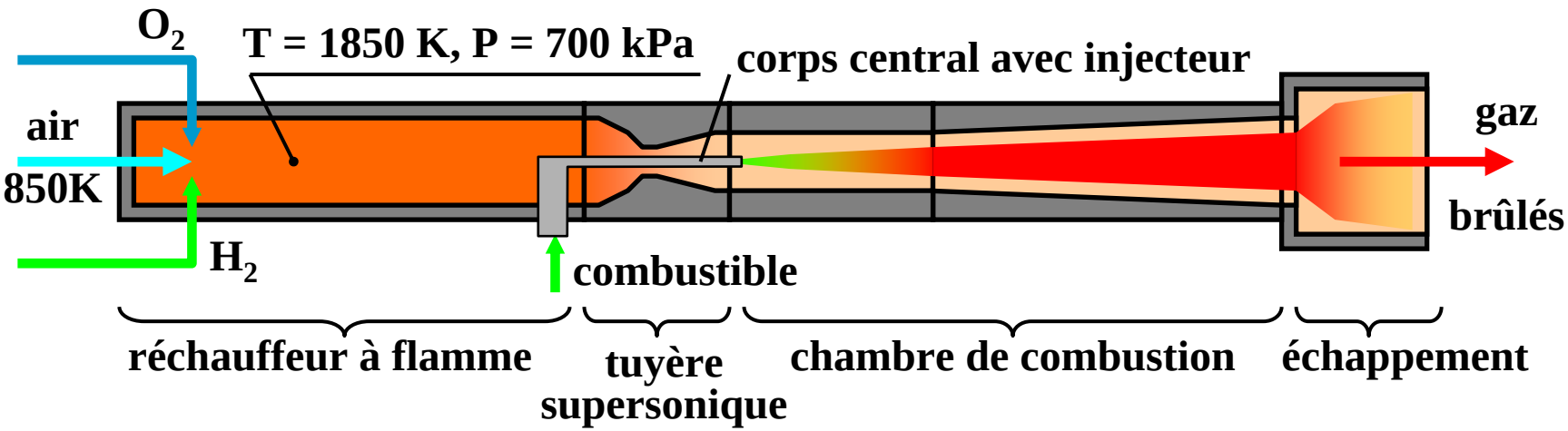
V. Conclusions et perspectives

Combustion supersonique

Statoréacteur à combustion supersonique

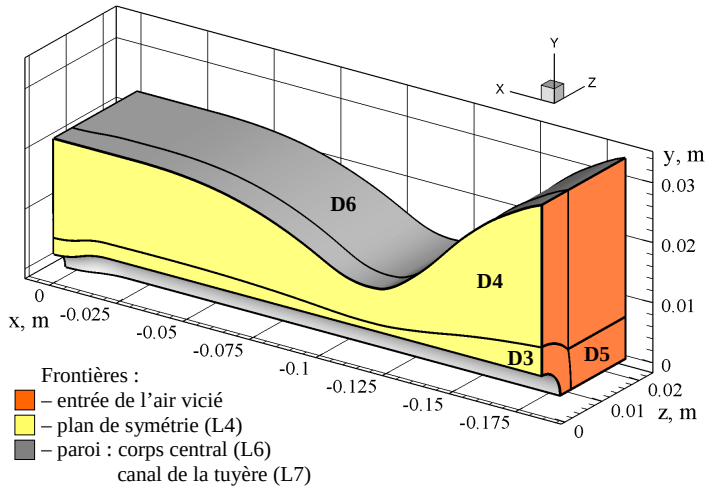


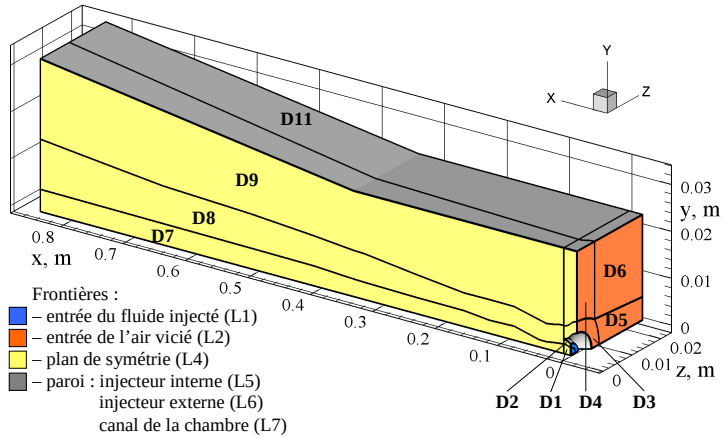
Expérience LAERTE à l'ONERA



Simulation de la combustion de H_2

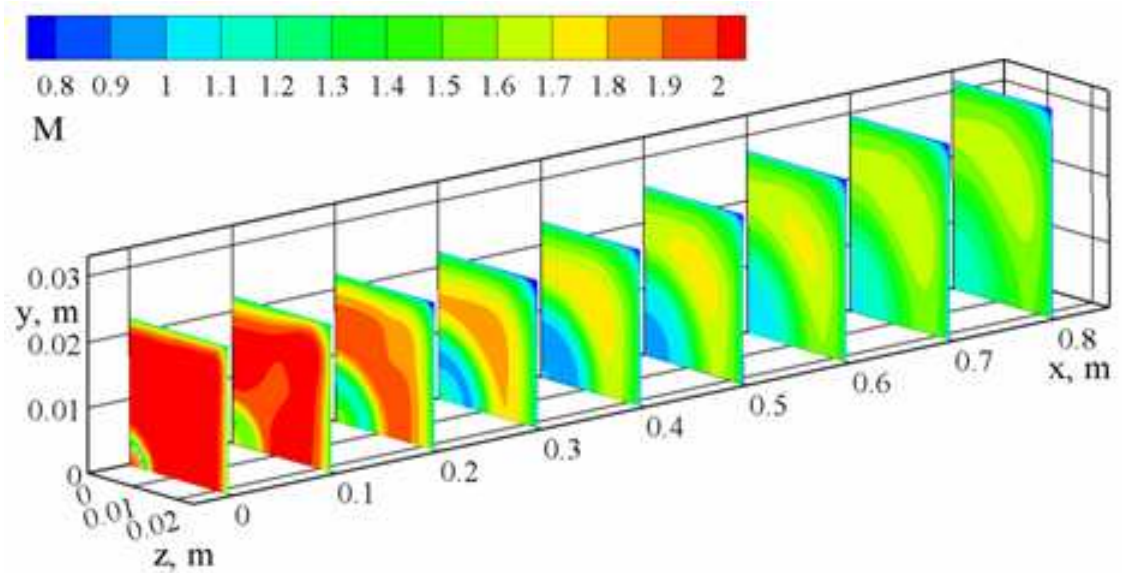
- **Code MSD (ONERA)**
  - Résolution des équations Navier-Stokes en 3D
  - Adaptation à la plate-forme vectorielle et parallélisation MPI par décomposition du domaine de calcul
- **Modèles**
  - Gaz : 7 espèces réactives
  - Chimie : 7 réactions
  - Turbulence : *k*-ε
- **Maillage**
  - Structuré par blocs
  - Tuyère supersonique : 1.1 x 10^6 cellules
  - Chambre de combustion : 4.4 x 10^6 cellules



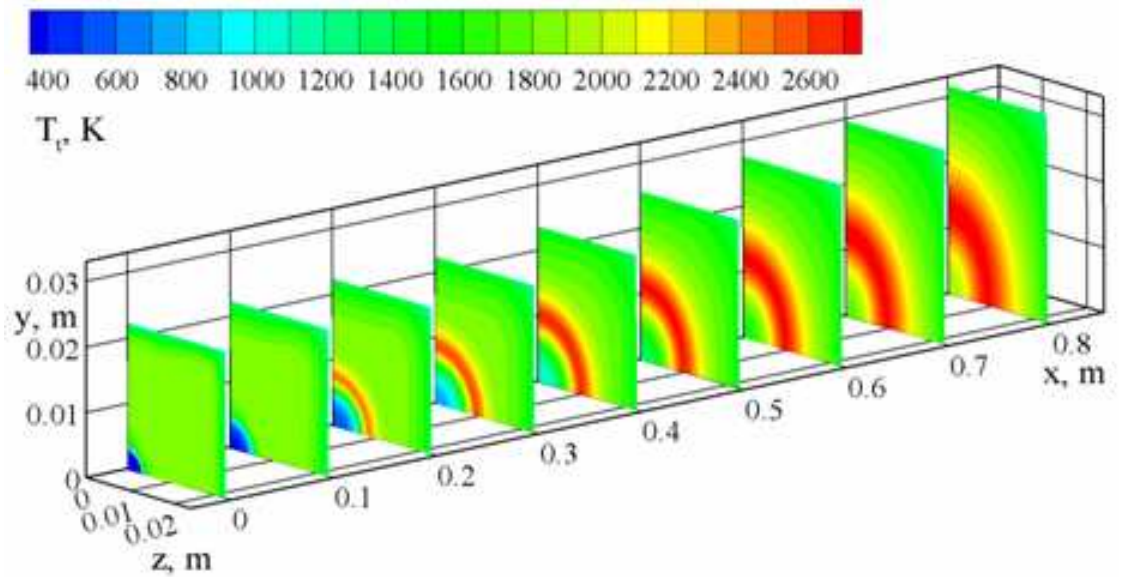


Résultats de la simulation

Champ de nombre de Mach

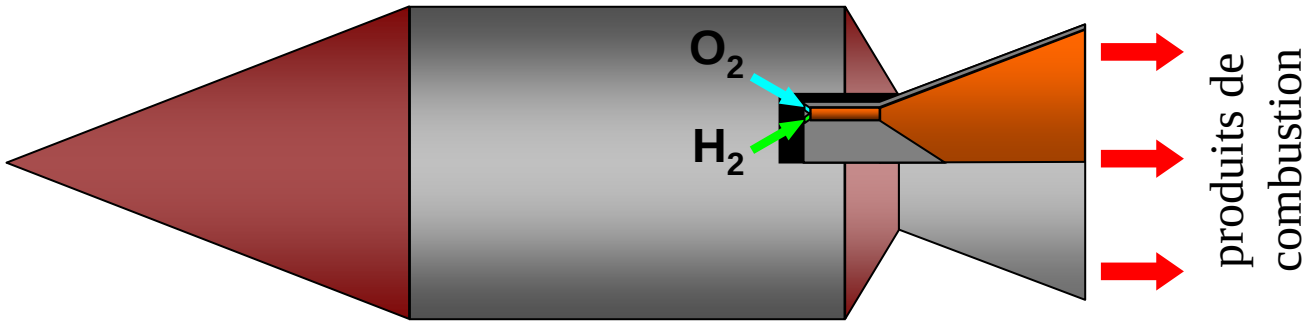


Champ de température totale

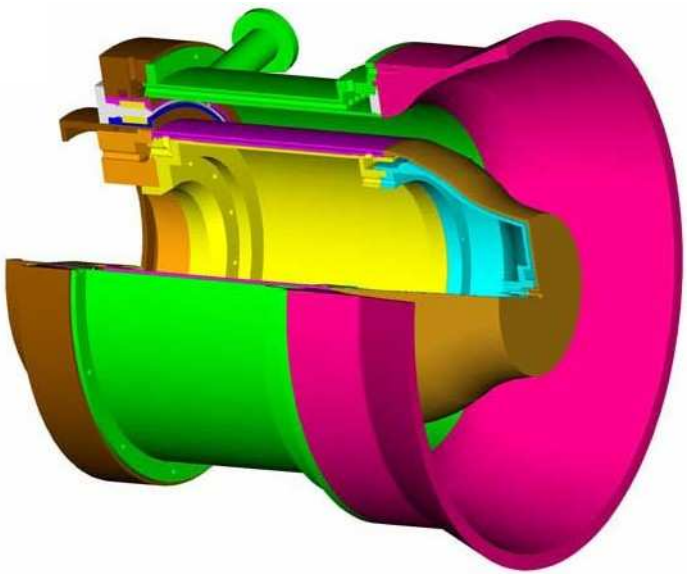


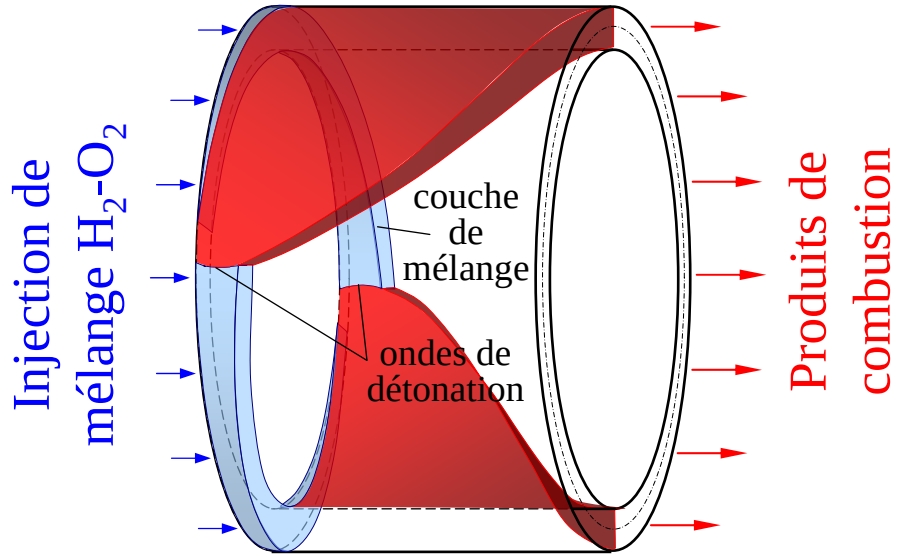
Moteur fusée à détonation continue

Moteur fusée à détonation continue rotative



Chambre de combustion du moteur à détonation

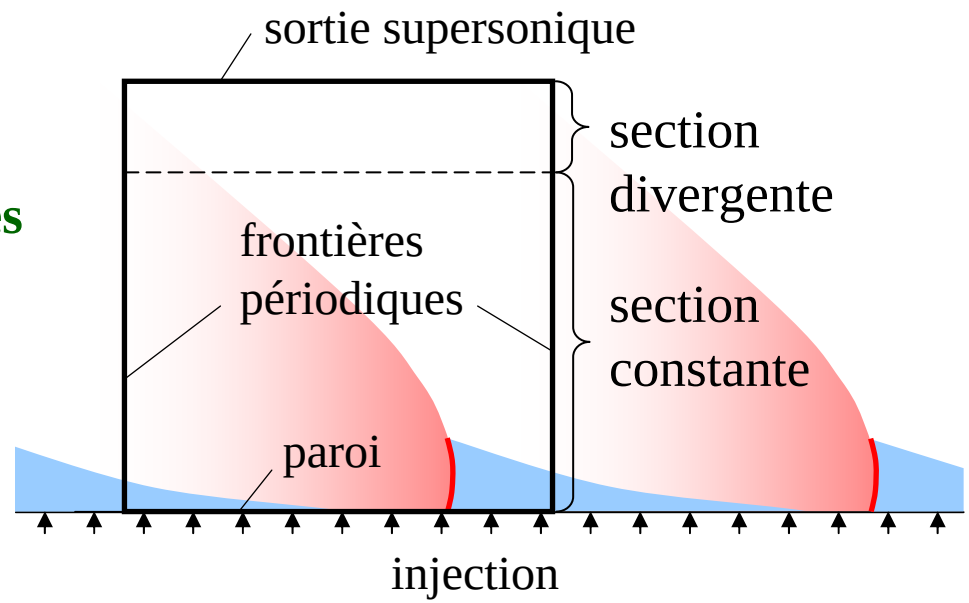




Simulation de la détonation rotative

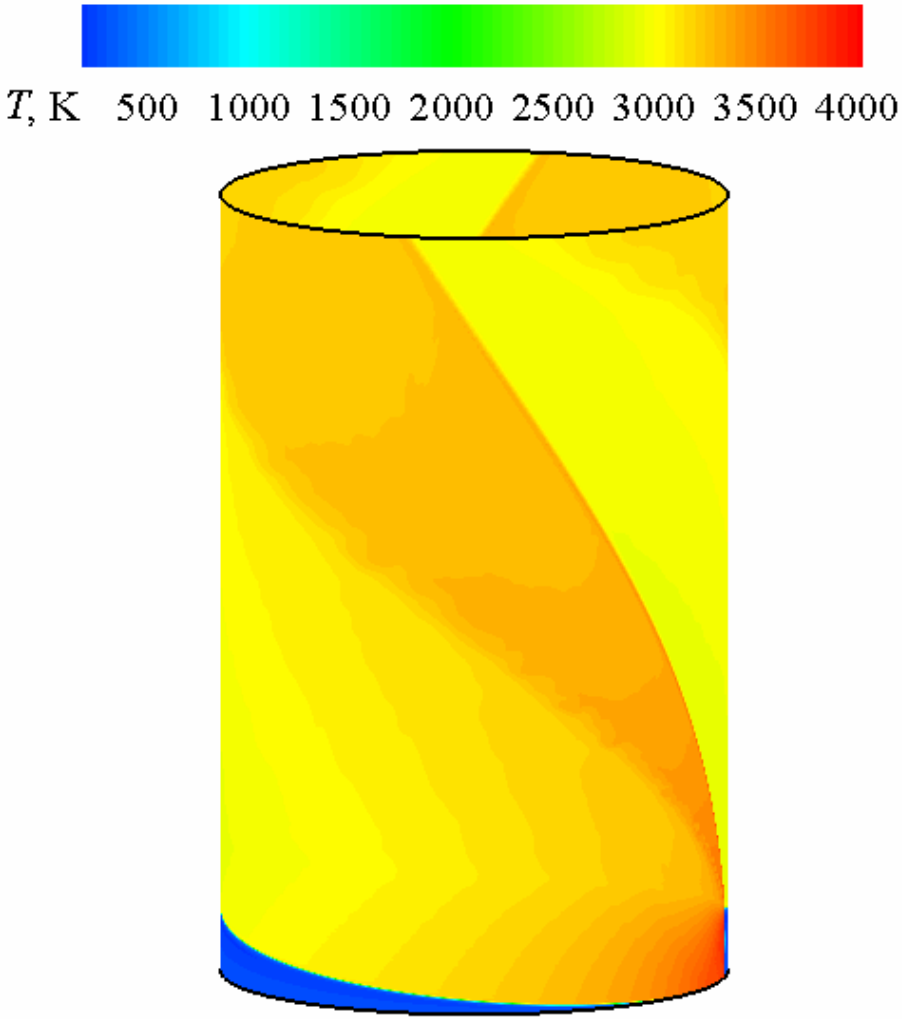
- **Code WENO (ICARE)**
  - Résolution des équations Euler non stationnaires en 2D
  - Schémas numériques haute résolution
  - Parallélisation MPI par décomposition du domaine de calcul
- **Modèles**
  - Gaz : 6 espèces réactives
  - Chimie : 7 réactions
- **Maillage**
  - Structuré orthogonal
  - De 4×10^5 à 5×10^5 cellules

Domaine de calcul 2D périodique

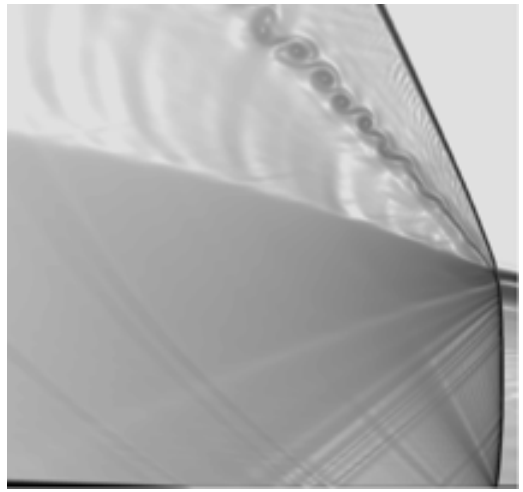


Résultats de la simulation

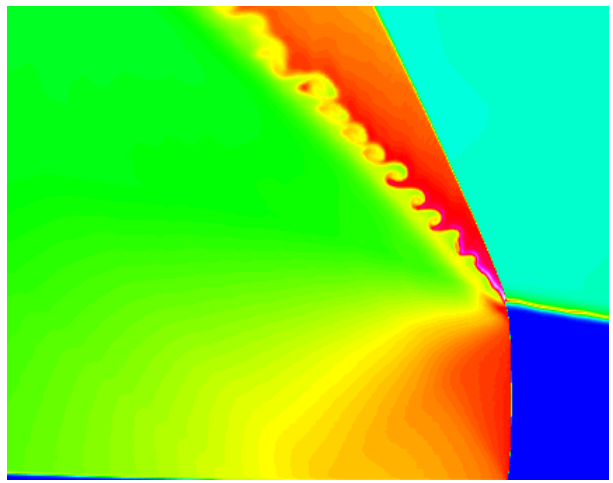
Champ de température



Gradient de densité



Fraction du radical O



Plan de l'exposé

I. Introduction

II. Plates-formes parallèles et principes de parallélisation

III. Quelques tests comparatifs

IV. Simulations parallèles d'écoulements réactifs à l'ICARE-CNRS

V. Conclusions et perspectives

Conclusions et perspectives

- **Simulations complexes grâce à l'utilisation efficace de plates-formes parallèles**
  - **Configurations 2D et 3D en échelle réelle**
  - **Conditions pratiquement intéressantes en terme de pression et de température**
  - **Systèmes réactionnels détaillés**
- **Amélioration d'outils de simulation**
  - **Création d'une plate-forme de calcul scientifique à l'ICARE**
  - **Application des techniques de tabulation pour le traitement de systèmes réactifs complexes (hydrocarbures)**
  - **Intégration de techniques d'adaptation dynamique de maillage (*Adaptive Mesh Refinement*)**