Computational fluid dynamics with irregular grids on the connection machine

Charbel Farhat, Loula Fatima Fezoui, Stephane Lanteri

Rapports de Recherche

N° 1411

Programme 6
Calcul Scientifique, Modélisation et Logiciels numériques

COMPUTATIONAL FLUID DYNAMICS WITH IRREGULAR GRIDS ON THE CONNECTION MACHINE

Charbel FARHAT
Loula FEZOUI
Stéphane LANTERI

Avril 1991

RR.1411

COMPUTATIONAL FLUID DYNAMICS WITH IRREGULAR GRIDS ON THE CONNECTION MACHINE

+Charbel FARHAT , ++Loula FEZOUI
++Stéphane LANTERI

+ Department of Aerospace Engineering
and Center for Space Structures and Controls
University of Colorado
Campus Box 429
BOULDER, COLORADO 80309 (USA)
++ INRIA Sophia-Antipolis
2004, Route des Lucioles
06560 VALBONNE (FRANCE)

COMPUTATIONAL FLUID DYNAMICS WITH IRREGULAR GRIDS ON THE CONNECTION MACHINE

Charbel FARHAT, Loula FEZOUI, Stéphane LANTERI

ABSTRACT : Here we report on our recent effort in solving two dimensional Euler equations on the Connection Machine model CM-2, using a mixed finite volume/finite element method on fully unstructured grids. Because the mesh irregularities inhibit the use of the North-East-West-South (NEWS) fast communication mechanism, we focus on the development and implementation of a communication efficient strategy for mapping thousands of processors arranged in a hypercube topology onto an arbitrary mesh. The main objective is to minimize the interprocessor communication costs that are induced by the ROUTER, while still maintaining a high level of parallelism and load balancing in the computations. We report performance results for internal and external flow problems which indicate that a 16K CM-2 with single precision floating point arithmetic is at least as fast as one processor CRAY-2 and in some cases twice as fast for solving such problems, and this for a virtual processor ratio (VPR) as small as one.

CALCULS EN DYNAMIQUE DES FLUIDES AVEC DES GRILLES IRREGULIERES SUR LA CONNECTION MACHINE

RESUME : On présente ici les résultats de notre travail pour résoudre les équations d'Euler bidimensionnelles sur la Connection Machine CM-2, en utilisant une méthode mixte éléments/volumes finis sur des grilles irrégulières. Le caractère non-structuré des maillages en éléments finis ne permet pas l'utilisation des mécanismes de communication performants en grille (NEWS). Nous nous concentrons donc sur le développement d'une stratégie efficace pour appliquer des milliers de processeurs arrangés en une topologie hypercube, sur un maillage arbitraire. L'objectif recherché est la minimisation des coûts de communication unduits par l'utilisation du ROUTEUR, tout en maintenant un haut degré de parallélisme et une répartition efficicace des calculs locaux. Nous présentons des résultats numériques pour des écoulements internes et externes qui montrent qu'une CM-2 avec 16K processeurs physiques et des accélérateurs de calculs flottants en simple précison, est au moins aussi rapide qu'un CRAY-2 monoprocesseur (double précision) et dans certains cas deux fois plus rapide, et ceci sans la mise en jeu de processeurs virtuels.

Contents

1 Introduction

In our previous study [15], we have reported on our first experiments on Computational Fluid Dynamics on the Connection Machine CM-2. It has been shown that the massively parallel CM-2 is particularly well suited to structured computations. Communication steps on regular grids were performed using the NEWS (North-East-West-South) fast communication mechanism. Consequently, at high virtual processor ratio (VPR), the communication part of the computations did not exceed 10% of the total CPU time; moreover, high efficiency rates of 2 GFLOPS were deduced for the 64K processors machine. We were also motivated by comparisons with a classical supercomputer such as the CRAY-2. We have shown that for non-viscous (Euler equations) and viscous (Navier-Stokes equations) computations on structured grids, the CM-2 was respectively 16 and 24 times faster than the CRAY-2 mono-processor (again these ratios were deduced for the complete configuration).

In the present work we discuss our recent effort in solving two-dimensional Euler equations on the CM-2 using a mixed finite volume/finite element methods on fully unstructured grids. We are interested in both steady and unsteady simulations of two-dimensional compressible flows around arbitrary geometries. The domain of interest is discretized using triangular finite elements yielding to completely unstructured grids. In section 2, we first recall the mathematical modeling of the problem; we then derive first-order and second-order spatial schemes that are characterized by an upwind integration of the convective fluxes. Second order accuracy is obtained through a MUSCL (Monotonic Upwind Scheme for Conservation Laws) technique. Finally time integration is obtained with an explicit (and therefore nicely parallelizable) Runge-Kutta method.

Because the mesh irregularities inhibit the use of the NEWS mechanism, we have to develop an efficient scheme for carrying out communications of an arbitrary pattern. In section 3 we describe a strategy for mapping thousands of processors, arranged in a hypercube topology, onto an unstructured grid. The key elements of this strategy are given by the selection of an appropriate parallel data structure, the partitioning of a given unstructured grid into subgrids, and the mapping of each individual processor onto an entity of these subgrids. The main objective is to minimize the interprocessor communication costs that are induced by the general communication mechanism (the ROUTER), while still maintaining a high level of parallelism and load balancing in the computations. The principal difficulty relates to the fact that the mesh topology is given only at run time (problem dependent) and can hardly be embedded onto a hypercube. The parallel data structure is selected as to introduce a high level of parallelism in the selected algorithm while minimizing the storage requirements per processor and the overall number of redundant operations. Whenever the resulting communication patterns are compiled during the first iteration or time step, the total time elapsed in interprocessor communication using the router is drastically reduced to represent in some cases only 6% of the total CPU time of the simulation.

The performance of the resulting massively parallel schemes is assessed in Section 4 on the Connection Machine CM-2 for various mesh densities and various CM-2 sizes. Performance comparisons of the presented solution algorithms with vectorized versions where scatter/gather manipulations are treated with a coloring technique (Farhat and Crivelli [5]) indicate that for such irregular problems a 16K CM-2 with single precision floating point arithmetic is at least as fast as one processor CRAY-2 and in some cases twice as fast, and this for a VP ratio as small as one.

We also point out that additional work related to Computational Fluid Dynamics (CFD) on the Connection Machine can be found in Jespersen and Levit [14], Long [16], Egolf [4] and Clary, Howell and Karman [2].

2 Numerical solutions of the Euler equations

2.1 Mathematical modelling

In this section we recall the mathematical problem to be solved and set some definitions and notations which are used in the sequel.

2.1.1 Governing equations

Let $\Omega \subset \mathbb{R}^2$ be the flow domain of interest and Γ be its boundary. The conservative law form of the equations describing two-dimensional Euler flows is given by :

$$\frac{\partial W}{\partial t} + \frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial y} = 0 \tag{1}$$

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} = (W^{(k)})_{k=1,4}$$

$$F(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix} , \quad G(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}$$

where ρ is the density, $\vec{U} = (u, v)$ is the velocity vector, E is the total energy per unit of volume, and p is the pressure of the fluid, with the equation of state given for a perfect gas as :

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \parallel \vec{U} \parallel^2) \tag{2}$$

γ is the ratio of specific heats ($\gamma = 1.4$ for air). We introduce the vector

$$\vec{\mathcal{F}}(W) = \begin{pmatrix} F(W) \\ G(W) \end{pmatrix} \tag{3}$$

The conservative system (1) can be rewritten in the following vector form

$$\frac{\partial W}{\partial t} + \vec{\nabla}.\vec{\mathcal{F}}(W) = 0 \tag{4}$$

where $\vec{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$.

We recall that system (1) is a hyperbolic system, so that for every couple of real (α_1, α_2), the Jacobian matrix $P(\alpha_1, \alpha_2, W) = \alpha_1 F'(W) + \alpha_2 G'(W)$ is diagonalizable in the diagonal matrix $\Lambda(\alpha_1, \alpha_2, W) = \text{diag}[\lambda^{(k)}(\alpha_1, \alpha_2, W)]$ and its right eigenvector matrix is denoted by $T(\alpha_1, \alpha_2, W)$. For any real function f, we can define :

$$f(P(\alpha_1, \alpha_2, W)) = T^{-1}(\alpha_1, \alpha_2, W)\text{diag}\left[f(\lambda^{(k)}(\alpha_1, \alpha_2, W))\right] T(\alpha_1, \alpha_2, W)$$

2.1.2 Boundary conditions

We are mostly interested in external flows around airfoils; then the domain of computation Ω is delimited by the boundary $\Gamma = \Gamma_b \cup \Gamma_\infty$. Let $\vec{n}$ be the outward unit normal at any point of the boundary Γ (see Figure 1).

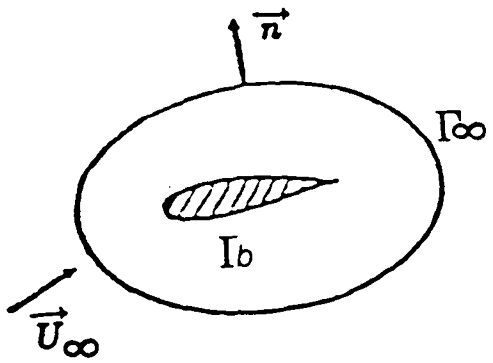

Figure 1 : The computational domain

The flow is assymed to be uniform at infinitiy, and the variables to be non-dimensionalized by the free-stream vector W_∞ given by :

$$\rho_\infty = 1 \ , \ \vec{U}_\infty = \begin{pmatrix} cos\alpha \\ sin\alpha \end{pmatrix} \ , \ p_\infty = \frac{1}{\gamma M_\infty^2} \tag{5}$$

where α is the angle of attack and M_∞ denotes the free-stream Mach number. On the wall boundary Γ_b, we assume the slip condition :

$$\vec{U}.\vec{n} = 0 \tag{6}$$

2.1.3 Definitions

We assume that Ω is a polygonal bounded domain of $\mathbb{R}^2$. Let $\mathcal{T}_h$ be a standard triangulation of Ω and h the maximal length of the edges of the triangles of $\mathcal{T}_h$. We need to introduce the following notations.

For every vertex $S_i (i = 1, ..., ns)$ of $\mathcal{T}_h$, the cell C_i is the union of the subtriangles resulting from the subdivision by means of the medians of each triangle of $\mathcal{T}_h$ and having S_i as a vertex (see Figure 2). The boundary of C_i is denoted by ∂C_i and the unit vector of the outward normal to ∂C_i by $\vec{\nu}_i = (\nu_{ix}, \nu_{iy})$. The union of all these cells constitutes a partition of domain Ω.

$$\Omega = \bigcup_{i=1}^{ns} C_i$$

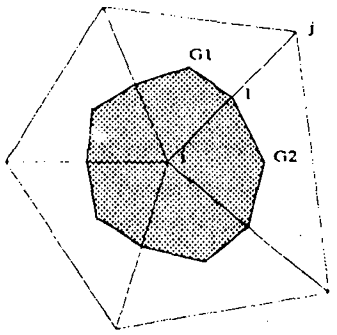

Figure 2 : Control volume in an unstructured grid

For every vertex S_i, $K(i)$ is the set of neighboring nodes of S_i. We recall that Ψ_i is the characteristic function of the cell C_i defined as :

$$\Psi_i(\vec{X}) = \begin{cases} 1 & \text{if } \vec{X} \in C_i \\ 0 & \text{otherwise} \end{cases}$$

We introduce the following discrete spaces (where P_1 is the space of polynomials in two variables of degrees 1) :

$$\begin{aligned} \mathcal{V}_h &= \{v_h \mid v_h \in C^0(\Omega), v_h\,|_T \in P_1, \forall T \in \mathcal{T}_h\} \\ \mathcal{W}_h &= \{v_h \mid v_h \in L^2(\Omega), v_h\,|_{C_i} = v_i = \text{const}; i = 1, ..., ns\} \end{aligned}$$

Any function φ belonging to $\mathcal{V}_h$ is uniquely determined by its values $\varphi(S_i)$ at each vertex S_i and if we note $(N_i)_{i=1}^{ns}$ the basis set of $\mathcal{V}_h$, we have :

$$\varphi(\vec{X}) = \sum_{i=1,ns} \varphi(S_i) N_i(\vec{X})$$

There exists a natural bijection between spaces $\mathcal{V}_h$ and $\mathcal{W}_h$ defined by :

$$\forall \varphi \in \mathcal{V}_h \quad , \quad S(\varphi(\vec{X})) = \sum_{i=1,ns} \varphi(S_i) \Psi_i(\vec{X})$$

2.2 Upwind approximations

First, a systematic procedure is developed for extending any 3-point scheme defined by a numerical flux function to the case of irregular mesh triangulations. Next, an extension to high order approximations is proposed.

2.2.1 First-order accurate scheme

Consider a 1-dimensional system of conservation laws with a flux function given by $F : \mathbb{R}^m \to \mathbb{R}^m$, where m is the dimension of the system :

$$\frac{\partial W}{\partial t} + A(W)\frac{\partial W}{\partial x} = 0 \quad W \in \mathbb{R}^m \tag{7}$$

The matrix $A(W) = dF/dW$ is the Jacobian matrix of the flux function. It is well known that a 3-points conservative finite difference scheme is characterized by its numerical flux function $\Phi_F(U,V)$ which depends on two variables U and V and satisfying the consistency equation $\Phi_F(U,U) = F(U)$.

The expression of the semi-discretized approximation of (7) is given by :

$$\frac{\partial W_h}{\partial t} + \frac{1}{\Delta x}\left(\Phi_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}}\right) = 0 \tag{8}$$

with

$$\Phi_{i+\frac{1}{2}} = \Phi_F(W_i, W_{i+1})$$

$$\Phi_{i-\frac{1}{2}} = \Phi_F(W_{i-1}, W_i)$$

where W_i is the value of the numerical approximation W_h of the solution of (7) at $x = i\Delta x$, Δx being the discrete spatial increment. A two-dimensional extension of this class of schemes is constructed as follows. First, a variational approach of equation (1) is derived as :

$$\text{Find } W_h \in (\mathcal{V}_h)^m, \quad \forall \varphi_h \in \mathcal{V}_h$$

$$\iint\limits_{\Omega} \frac{\partial W_h}{\partial t}\varphi_h dxdy + \iint\limits_{\Omega} \vec{\nabla}.\vec{\mathcal{F}}(W_h)S(\varphi_h)dxdy = 0 \tag{9}$$

The above formulation can be called a finite volume Galerkin (FVG) approximation. Treating also the left-hand-side integral of (9) with the operator S leads to a mass-lumped variant of the variational approach of equation (1). Choosing the function φ_h as a shape function N_i associated to the node S_i and integrating by parts the above equation on each cell C_i transforms the problem into :

$$\text{Find } W_h \in (\mathcal{W}_h)^m$$

$$\iint\limits_{C_i} \frac{\partial W_h}{\partial t} dxdy = -\int\limits_{\partial C_i} \vec{\mathcal{F}}(W_h).\vec{\nu}_i d\sigma \tag{10}$$

Equality (10) above expresses the flux balance for the control volume C_i. Specifying the approximation for computing the right-hand-side integral in (10) completes the description of the spatial integration scheme. For that purpose, the boundary ∂C_i of the cell C_i is split in bi-segments ∂C_{ij} which join the middle point of the segment $[S_iS_j]$ to the centroids of the triangle having S_i and S_j as common vertices (see Figure 3). Let us introduce the notations :

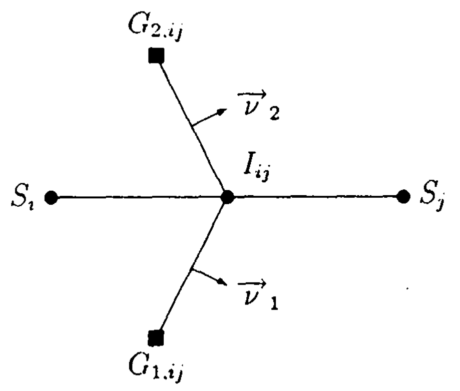

Figure 3 : Definition of ∂C_{ij}

After the following quantities are defined :

$$\begin{aligned}
\mathcal{F}(U,\vec{\nu}_{ij}) &= \vec{\mathcal{F}}(U). \int_{\partial C_{ij}} \vec{\nu}_{ij} d\sigma \\
\mathcal{A}(U,\vec{\nu}_{ij}) &= F'(U) \int_{\partial C_{ij}} \nu_{ij}^{x} d\sigma + G'(U) \int_{\partial C_{ij}} \nu_{ij}^{y} d\sigma \\
&= \frac{\partial}{\partial U}[\vec{\mathcal{F}}(U)]. \int_{\partial C_{ij}} \vec{\nu}_{ij} d\sigma
\end{aligned}$$

problem (10) becomes :

Find $W_h \in (\mathcal{W}_h)^m$

$$
(11)\quad \iint_{C_i} \frac{\partial W_h}{\partial t} dxdy = \begin{aligned}[t] & - \sum_{j\in k(i)} \int_{\partial C_{ij}} \vec{\mathcal{F}}(W_h).\vec{\nu}_{ij} d\sigma && <1> \\ & - \int_{\partial C_i \cap \Gamma_b} \vec{\mathcal{F}}(\bar{W}_h).\vec{n}_i d\sigma && <2> \\ & - \int_{\partial C_i \cap \Gamma_\infty} \vec{\mathcal{F}}(\bar{W}_h).\vec{n}_i d\sigma && <3> \end{aligned}
$$

Let $H_{ij}^{(1)}$ denote the first term $<1>$ of the right-hand side of (10). The computation of this term will involve the numerical flux function Φ of a first-order accurate upwind scheme described in (8) by :

$$H_{ij}^{(1)} = \Phi_{\mathcal{F}_{ij}}(W_i, W_j, \vec{\nu}_{ij})$$

where $W_i = W_h(S_i)$ and $W_j = W_h(S_j)$. We recall the definition (by their numerical flux function) of the first-order accurate schemes we have used in this study :

- *Van Leer's flux vector splitting* [20] (see also [11] for a complete description of this flux in the case of finite-elements mesh)

$$(12)\qquad \Phi_{\mathcal{F}}^{VL}(U,V,\vec{\nu}) = \| \vec{\nu} \| \; R_\theta^{-1} \left(F^+(\hat{U}) + F^-(\hat{U}) \right)$$

- *Steger and Warming's scheme* [18]

$$(13)\qquad \Phi_{\mathcal{F}}^{SW}(U,V,\vec{\nu}) = \mathcal{A}^+(U,\vec{\nu})\,U + \mathcal{A}^-(V,\vec{\nu})\,V$$

2.2.2 A second-order extension

The numerical integration with an upwind scheme as descrided previously leads to approximations which are only first-order accurate. We present a modification of problem (11) in order to get a second-order accurate solution without changing the approximation space. The key ingredients fro constructing such a second-order accurate approximation are :

(1) A dissipative first-order accurate (quasi-monotone) scheme,

(2) A second-order scheme derived from the previous one by using linear interpolations in the computations of the fluxes,

(3) A limiting procedure which reduces the oscillations of the solution.

A construction of this type is introduced in [19]; It is developed in the present context as an extension of Van Leer's M.U.S.C.L. (Monotonic Upwind Scheme for Conservation Laws) method [21] to the case of unstructured meshes.

In the one-dimensional case, the method is based on a linear interpolation of W in each interval $I_i = [(i - \frac{1}{2})\Delta x, (i + \frac{1}{2})\Delta x]$ as :

$$\begin{aligned} W(x) &= W_i + (x - x_i)P_i \quad \text{for} \quad x \in I_i \\ P_i &= \frac{W_{i+1} - W_{i-1}}{2\Delta x} \end{aligned}$$

The fluxes are computed with the values :

$$\begin{aligned} W^n_{i+\frac{1}{2}^-} &= W_i + \frac{\Delta x}{2} P_i \\ W^n_{i+\frac{1}{2}^+} &= W_{i+1} - \frac{\Delta x}{2} P_{i+1} \end{aligned}$$

of W at each side of the interface $x_{i+\frac{1}{2}}$. The spatially second-order accurate version of (8) is then given by :

$$\left(\frac{\partial W_h}{\partial t}\right)_i + \frac{1}{\Delta x}(\Phi_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}}) = 0 \tag{14}$$

with

$$\begin{aligned} \Phi_{i+\frac{1}{2}} &= \Phi_F(W_{i+\frac{1}{2}^-}, W_{i+\frac{1}{2}^+}) \\ \Phi_{i-\frac{1}{2}} &= \Phi_F(W_{i-\frac{1}{2}^-}, W_{i-\frac{1}{2}^+}) \end{aligned}$$

2.2.3 Interpolation in the two-dimensional case

In the two-dimensional case, a second-order accurate approximation requires the evaluation of the gradient of the solution at each vertex. Clearly, the gradient of a function v_h of V_h is constant in each element and discontinuous in the domain. Following the M.U.S.C.L. method, one way to reach the second-order spatial accuracy is to evaluate fluxes with extrapolated values W_{ij}, W_{ji} at the interface $\partial C_i \cap \partial C_j$. The resulting scheme is a direct extension of the first-order accurate scheme (11) and is formulated as :

$$\text{Find } W_h \in (\mathcal{W}_h)^m$$

$$(15) \qquad \begin{aligned} \iint\limits_{C_i} \frac{\partial W_h}{\partial t} dxdy = & - \sum_{j \in K(i)} H_{ij}^{(2)} \\ & - \int\limits_{\partial C_i \cap \Gamma_b} \vec{\mathcal{F}}(\bar{W}_h).\vec{n}_i d\sigma \\ & - \int\limits_{\partial C_i \cap \Gamma_\infty} \vec{\mathcal{F}}(W_h).\vec{n}_i d\sigma \end{aligned}$$

where

$$(16) \qquad \begin{aligned} H_{ij}^{(2)} &= \Phi_{\mathcal{F}_{ij}}(W_{ij}, W_{ji}, \vec{\nu}_{ij}) \\ W_{ij} &= W_i + \frac{1}{2}(\vec{\nabla} W)_i . \vec{S_iS_j} \\ W_{ji} &= W_j - \frac{1}{2}(\vec{\nabla} W)_j . \vec{S_iS_j} \end{aligned}$$

and where the approximate nodal gradients $(\vec{\nabla} W)_i$ are obtained from the Galerkin gradients as follows :

$$(17) \qquad (\vec{\nabla} W)_i = \frac{1}{\text{area}(C_i)} \iint\limits_{C_i} \nabla W \mid_T dxdy$$

The outcome is a half-upwind (Fromm-like) scheme which is spatially second-order accurate but may present spurious oscillations in the solutions, expressing a loss of monotony. One way to circumvent this problem is to apply a slope limitation procedure.

2.2.4 The limiting procedure

In order to measure the upstream and downstream variations of the unknown on the segment $\vec{S_iS_j}$, fictitious values W_{ij}^* , W_{ji}^* are introduced in addition to the nodal values W_i , W_j. These fictitious values are indeed nodal values at points $i+1$, $i-2$ if a structured grid is used ($j = i-1$). W^*_{ij} and W^*_{ji} are derived from the nodal gradients (17) as follows :

$$(18) \qquad \begin{cases} W^*_{ij} & = W_i - 2(\vec{\nabla} W)_i . \vec{S_i S_j} + (W_j - W_i) \\ \\ W^*_{ji} & = W_j - 2(\vec{\nabla} W)_j . \vec{S_j S_i} + (W_i - W_j) \end{cases}$$

The limited slopes are obtained using the van Albada limiter which can be written as :

$$(19) \qquad \begin{cases} dW_i & = Average\left(W_j - W_i, W_i - W^*_{ij}\right) \\ \\ dW_j & = Average\left(W_i - W_j, W_j - W^*_{ji}\right) \end{cases}$$

where

$$(20) \qquad Average\,(a,b) = \begin{cases} \dfrac{a(b^2+\varepsilon^2)+b(a^2+\varepsilon^2)}{a^2+b^2+2\varepsilon^2} & \text{if } a.b > 0 \\ \\ 0 & \text{otherwise} \end{cases}$$

Finally the limited arguments for the numerical flux function Φ are computed as :

$$(21) \qquad \begin{cases} W^{lim}_{ij} & = W_i + \frac{1}{2} dW_i \\ \\ W^{lim}_{ji} & = W_j + \frac{1}{2} dW_j \end{cases}$$

The resulting scheme (15) is now second-order accurate (except at the extrema) and the solutions it generates are oscillation-free in most cases (transonic flows). However, positiveness may be lost with more severe conditions such as high Mach numbers. More details about such techniques for unstructured grids and other limitations procedure may be found in [10].

2.2.5 Boundary conditions

The second term $< 2 >$ and the third term $< 3 >$ of the right-hand side of (15) contain the physical boundary conditions. These are represented by the vector $\bar{W}_h$ which involves quantities that depend on the interior value W_h and quantities that are determined by the physical boundary conditions.

Wall boundary : the vector $\bar{W}_h$ is assumed to satisfy the slip condition thus term $< 2 >$ is computed as :

$$(22) \qquad \int_{\partial C_i \cap \Gamma_b} \mathcal{F}(\bar{W}_h).\vec{n}_i d\sigma = \int_{\partial C_i \cap \Gamma_b} \begin{pmatrix} 0 \\ \bar{p}.n_i^x \\ \bar{p}.n_i^y \\ 0 \end{pmatrix} d\sigma$$

where $\bar{p}$ is taken as the interior pressure. (Note that in this procedure, the slip condition is applied in a weak variational way, as in cell-centered finite volume formulations).

Inflow and outflow boundaries : at these boundaries, a precise set of compatible exterior data which depend on the flow regime and the velocity direction, is to be specified. For this purpose a *plus-minus* flux splitting is applied between exterior data and interior values. More precisely, the boundary integral $< 3 >$ is evaluated using the flux-splitting of Steger and Warming [18] :

$$(23) \qquad \int_{\partial C_i \cap \Gamma_\infty} \mathcal{F}(\bar{W}_h).\vec{n}_i d\sigma = \mathcal{A}^+(W_i, \vec{\nu}_{i\infty}).W_i + \mathcal{A}^-(W_i, \vec{\nu}_{i\infty}).W_\infty$$

2.3 Time integration

Once the spatial discretization is accomplished we obtain the following system of ordinary differential equations :

$$\frac{dW}{dt} + \psi(W) = 0$$

Because it lends itself to massive parallelism, the explicit Runge-Kutta method is selected for integrating the above equations. A 3-step variant is used here. It is summarized as :

$$\begin{cases} W^{(0)} &= W^{n} \\ W^{(1)} &= W^{(0)} - \dfrac{\Delta t}{3}\psi(W^{(0)}) \\ W^{(2)} &= W^{(0)} - \dfrac{\Delta t}{2}\psi(W^{(1)}) \\ W^{(3)} &= W^{(0)} - \Delta t\psi(W^{(2)}) \\ W^{n+1} &= W^{(3)} \end{cases}$$

The above scheme is often referred to as the low-storage Runge-Kutta method as only the solution at substep $\alpha - 1$ is used to compute the one at substep α. It is third-order accurate in the linear case but only second-order accurate in our case.

3 Parallel implementation on the CM-2

Clearly, discrete expressions from the previous section, reveal that both the spatial and temporal integrations are in principle nicely parallelizable. In this section, our interest lies in investigating the most efficient way to implement these computations on a massively parallel Single Instruction Multiple Data (SIMD) computer such as the Connection Machine CM-2. Here, efficiency relates to both performance and software architecture flexibility issues. In particular, we would like to implement all of first and second-order schemes for inviscid and viscous flows, within the same code. Special care is given to interprocessor communication because mesh irregularities inhibit the exploitation of the NEWS grid and require the use of the relatively slow router.

3.1 The parallel data structure on the CM-2

Behind the performance of any parallel algorithm lies the choice of the corresponding parallel data structure. The latter is closely related to both the entity and the task to be assigned to each processor. Therefore, all of the computational, communication and memory requirements should be considered before the distributed data structure is determined. For the mixed finite volume/finite element computations presented here, the node (vertex, grid point), element, edge, and cell data structures are all possible candidates.

While regular grids are most often characterized by their number of nodes, irregular grids can be characterized also by their number of elements or edges. Here, we assume for simplicity that $\mathcal{T}_h$ is characterized by its number of nodal points. Euler's relations for a triangulation state that :

$$\begin{aligned} number\_of\_nodes \; & + number\_of\_elements \\ & - number\_of\_edges = 1 \\ 2 \times number\_of\_edges \; & - number\_of\_boundary\_nodes \\ & = 3 \times number\_of\_elements \end{aligned}$$

which leads to :

$$\begin{aligned} number\_of\_elements \; & \approx 2 \times number\_of\_nodes \\ number\_of\_edges \; & \approx 3 \times number\_of\_nodes \end{aligned}$$

Therefore, if $\mathcal{T}_h$ is designed, for example, so that its number of nodes matches a given Connection Machine size, the VP ratio, operation count and communication requirements associated with each possible data structure are as follows :

	Node	Element	Edge	Cell
VPR	1	2	3	1
Operation count	2x	2x	x	2x
Communication	y	y	2y	y

Table 1:Parallel data structures

In Table 1 above, x and y are introduced only to highlight the relative requirements associated with the candidate data structures. Clearly, only the edge-based one does not generate redundant flux computations. However, this data structure and the node-based one are not practical, and therefore unacceptable, because they do not accommodate the evaluation of the Galerkin gradients (17) and would not handle future extensions to viscous flow computations. Moreover, the edge-based distribution requires twice as much communication as the other choices. On the other hand, the cell-based and element based data structures are suitable for all components of the first-order and second-order calculations and involve a minimum amount of communication. They do however result in redundant computations as each flux across an edge $[S_iS_j]$ shared by two triangles would be computed twice, once in the cell centered on S_i and once in the cell centered on S_j. Finally, the cell parallel data structure is preferred over the element-based one because it requires a smaller VP ratio.

It should be noted that in theory, it is possible to select a cell-based data structure and yet avoid some of the redundant flux computations. Basically, one could label the edges that are shared by more than one cell so that their corresponding fluxes are computed by an appropriate processor in one and only one cell. However for highly irregular meshes, the solution of the labeling problem may introduce a load unbalance, that is, it may assign a different number of edges to each processor that is mapped onto a cell. More importantly, the fluxes which would not be computed in a cell would need to be communicated to that cell. Therefore, redundant computations can be avoided only at the expense of additional communication that is characterized by an irregular pattern, which is not necessarily beneficial on a massively parallel processor such as the CM-2. Consequently, we have chosen the cell-based parallel data structure and have decided to accept the additional cost of redundant flux computations. The members of this parallel data structure are the coordinates of the cell center and cell nodes, the connectivity of the cell elements, the state variables at the cell center, the addresses of neighboring cells for guiding the router during irregular interprocessor communication, and various other cell related attributes. At each step, all computations are performed in parallel and only the state variables at the center of neighboring cells are communicated between their assigned processors (for a second order scheme, the nodal gradients at the center of neighboring cells are also communicated). The optimization of the communication phase is carried out in a two-step approach as follows.

3.2 The grid decomposition for the CM-2

Efficiency in arbitrary communication on the CM-2 requires the minimization of both the "hammering" on the router, that is, wire contention, and the distance that information has to travel, that is, the number of hops between the sender and receiver processors. Here, this implies that :

- adjacent cells must be assigned, as much as possible, to directly connected processors or processors that are lying in directly connected chips,
- contention for the wire connecting neighboring chips must be reduced.

In a first step, the unstructured grid is decomposed into a series of subgrids each containing 16 adjacent numerical cells. Each subgrid is assigned to a certain CM-2 chip that is subsequently identified, so that adjacent cells within a subgrid are assigned to directly connected processors lying in the same chip. As a result, off-chip communication is needed only across the subgrid boundaries. Wire contention is reduced if each of the defined subgrids is surrounded by the largest possible number of neighboring subgrids. Indeed, wherever a subgrid boundary is shared with several other subgrids, off-chip communication is split between distinct chips and is distributed across several of the available inter-chip wires (see Figure 4). On the other hand, if for example a subgrid is adjacent only to two other subgrids, a maximum of two wires can be used during off-chip communication, which may create a severe wire contention that would serialize communication and significantly increase its cost. Here, we use the mesh decomposer of Farhat [6] which has proven to be very effective at reducing wire contention on the CM-2 (Farhat, Sobh and Park [7]).

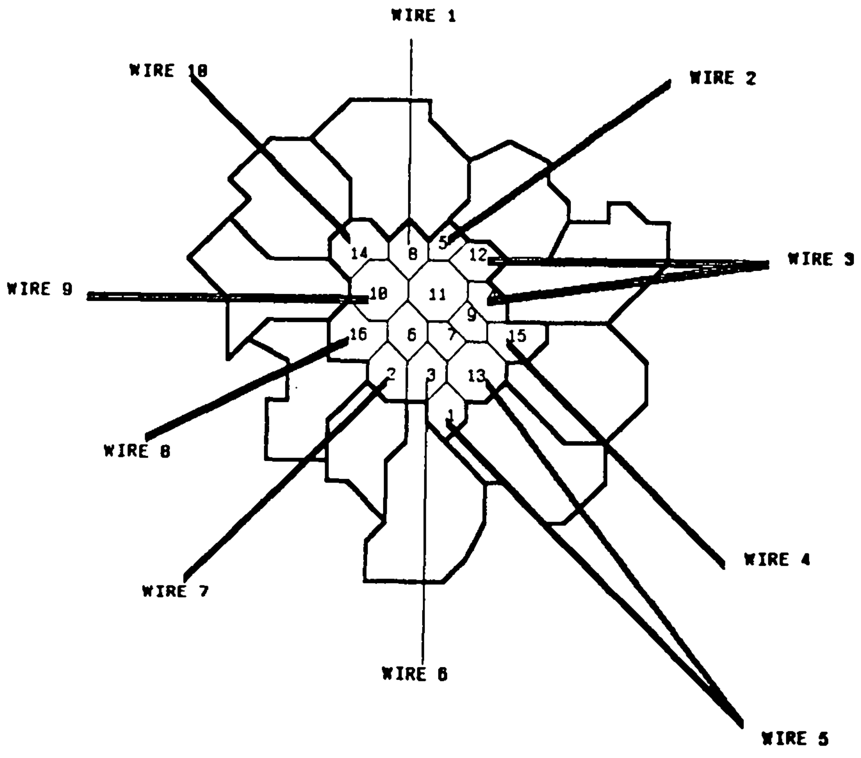

Figure 4 : Grid decomposition with reduced wire-contention

Next, we address the problem of mapping a specific chip onto a specific 16-cell subgrid.

3.3 Mapping and compiling

The next step is to reduce the distance that information has to travel during off-chip communication, that is when data is exchanged between centers of cells that are assigned to processors lying on different chips. This can be achieved by assigning adjacent subgrids as far as possible to directly connected chips. A combinatorial optimization-like procedure known as *Simulated Annealing* (see, for example, Flower, Otto and Salama [12]) is probably the most popular technique for tackling this mapping problem. However, it is a very expensive procedure which has often proved to be impractical. Alternative heuristic-based schemes have been developed by several authors including Bokhari [1], Farhat [8], and recently Hammond and Schreiber [13]. In this work, we have adopted the mapper of reference [8]. It is based on a combined greedy/divide and conquer approach and is tuned for hypercube topologies.

A detailed analysis of interprocessor communication on the CM-2 for unstructured grids can be found in Farhat, Sobh and Park [7] and [9]. In these references, it is shown that mesh irregularities induce an MIMD (Multiple Instruction Multiple Data) style of programming for the communication phase which dominates the cost of communication. It is also suggested that since the irregular pattern of communication is fixed in time, a considerable improvement can be achieved if that pattern is evaluated during the first time step (iteration), then compiled or stored in the CM-2 for re-use in sbsequent time steps (iterations). However, no software was available at that time for validating the proposed communication strategy. Recently, a communication compiler prototype has become available (see Dahl [3]) and can be used for storing the routing pattern. In Section 4, we report on its performance.

4 Performance results

Four test cases corresponding to two different problems are considered. Problem *P1* corresponds to a simplified modeling of the strong acoustic waves that may occur during an engine knock. The flow domain $\mathcal{D}$ is a $[0,1]\times[0,1]$ square with adiabatic slipping walls (see Figure 5). Different initial conditions are imposed in the 0.4×0.4 subdomain $\mathcal{D}_1$ and in the remaining part $\mathcal{D}_0$ of $\mathcal{D}$. These are :

$$\begin{array}{lll} \rho_0 = 1 & u_0 = v_0 = 0 & p_0 = 0.375 \\ \rho_1 = 1 & u_1 = v_1 = 0 & p_1 = 1.0 \end{array}$$

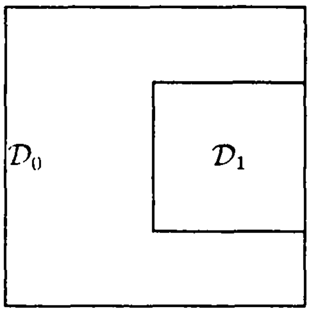

Figure 5 : Flow domain for problem P1

This test case has already been considered in [15]; Two test cases, *T1.1* and *T1.2* are associated with problem *P1*. Case *T1.1* corresponds to a mesh triangulation which results in 8192 grid points and case *T1.2* corresponds to a finer triangulation which delivers 16384 grid points. Both triangulations are regular but do not fit in a NEWS grid. Moreover, the regularity in the triangulation is not explicitly exploited in the parallel code.

Problem *P2* is associated with the simulation of a transonic flow around a NACA0012 airfoil at $M_\infty = 0.85$ and $\alpha = 0°$. Cases *T2.1* and *T2.2* correspond to two irregular triangulations which result in 3114 (Figure 6) and 12284 grid points, respectively. Note that neither the first mesh nor the second one are optimal for any size of the CM-2. These meshes were designed for earlier simulations on a CRAY-2 processor.

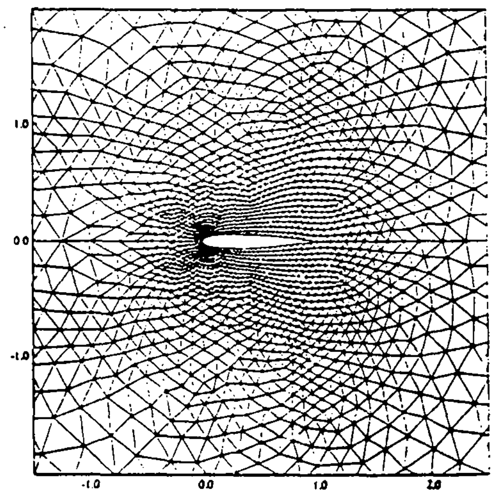

Figure 6 : Irregular triangulation for a NACA0012 flow domain

This test case results in a steady transonic flow; the steady iso-mach lines are depicted in Figure 7.

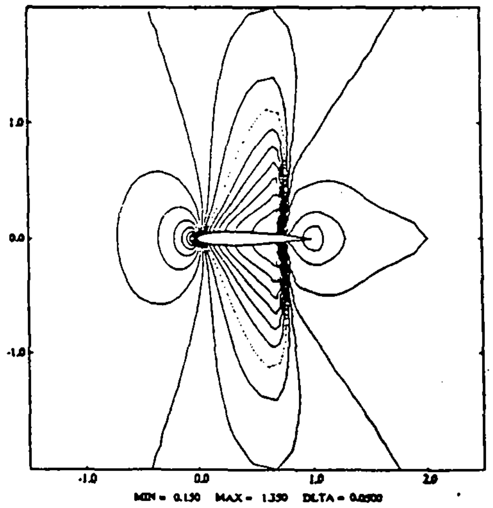

Figure 7 : Iso-mach lines for the NACA0012 flow

All computations on the CM-2 are performed with 32-bit Weitek floating point accelerators. The code is implemented using the new version of the C^*

language; some routines involving intensive computation steps are written in PARIS. The first set of performance results are given in Table 2 for the first-order accurate algorithm, and in Table 3 for the second-order version. For each case, the timings correspond to 200 iterations on an 8K CM-2 using a random processor mapping. "Comm." and "Comp." refer to the communication and computation elapsed time, respectively, "Tot." being the total CPU time.

Case	VPR	Tot. time	Comp. time	Comm. time	Comm./Tot.
T1.1	1	22.8 sec	12.1 sec	10.7 sec	47%
T1.2	2	48.8 sec	22.7 sec	26.1 sec	54%
T2.1	1	23.8 sec	12.1 sec	11.7 sec	49%
T2.2	2	47.4 sec	21.2 sec	26.2 sec	55%

Table 2 : Performance results on an 8K CM-2
First-order algorithm - **Random mapping**

Case	VPR	Tot. time	Comp. time	Comm. time	Comm./Tot.
T1.1	1	192 sec	95 sec	97 sec	51%
T1.2	2	407 sec	169 sec	238 sec	59%
T2.1	1	201 sec	94 sec	107 sec	54%
T2.2	2	401 sec	168 sec	233 sec	58%

Table 3 : Performance results on an 8K CM-2
Second-order algorithm - **Random mapping**

The above results indicate that the total elapsed time is almost equally split between computations and interprocessor communications, with communication being slightly dominant for the second-order accurate algorithm. Next, we report the performance results that are achieved when the decomposer/mapper module summarized in Section 3 is used.

Case	VPR	Tot. time	Comp. time	Comm. time	Comm./Tot.
T1.1	1	20.9 sec	12.1 sec	8.8 sec	42%
T1.2	2	40.7 sec	22.7 sec	18.0 sec	44%
T2.1	1	17.7 sec	12.1 sec	5.6 sec	32%
T2.2	2	34.6 sec	21.2 sec	13.4 sec	39%

Table 4 : Performance results on an 8K CM-2
First-order algorithm - **Our mapping**

Case	VPR	Tot. time	Comp. time	Comm. time	Comm./Tot.
T1.1	1	175 sec	95 sec	80 sec	45%
T1.2	2	327 sec	169 sec	158 sec	48%
T2.1	1	145 sec	94 sec	51 sec	35%
T2.2	2	296 sec	168 sec	128 sec	43%

Table 5 : Performance results on an 8K CM-2
Second-order algorithm - **Our mapping**

The effect of our mapper on the performance results is illustrated in Table 6 and Table 7 below.

Case	Unmap. Comm.	Map. Comp.	Unmap. Tot.	Map. Tot.
T1.1	10.7 sec	8.8 sec	22.8 sec	20.9 sec
T1.2	26.1 sec	18.0 sec	48.8 sec	40.7 sec
T2.1	11.7 sec	5.6 sec	23.8 sec	17.7 sec
T2.2	26.2 sec	13.4 sec	47.4 sec	34.6 sec

Table 6 : Mapped/Unmapped performance results on an 8K CM-2
First-order algorithm

Case	Unmap. Comm.	Map. Comp.	Unmap. Tot.	Map. Tot.
T1.1	97 sec	80 sec	192 sec	175 sec
T1.2	238 sec	158 sec	409 sec	327 sec
T2.1	107 sec	51 sec	201 sec	145 sec
T2.2	233 sec	128 sec	401 sec	226 sec

Table 7 : Mapped/Unmapped performance results on an 8K CM-2
Second-order algorithm

Clearly, the decomposer/mapper discussed in Section 3 decreases communication costs by 20 to 33% in Problem *P1* (regular triangulation) and by roughly 50% in problem *P2* (irregular triangulation). For a VP ratio of one, we were able to successfully use the communication compiler FASTGRAPH (FG) written by Dahl [3]. For higher VP ratios, FG constantly caused the system to crash. We hope that future releases of the communication compiler will resolve this problem. The improvement in communication performance due to FG and our mapper are highlighted in Table 8 below.

Case	VPR	Random	Mapper	FG	FG + Mapper
T1.1	1	10.7 sec	8.8 sec	2.70 sec	1.32 sec
T2.1	1	11.7 sec	5.6 sec	1.64 sec	0.82 sec

Table 8 : Communication performance results on an 8K CM-2
First-order algorithm

Clearly, the improvements due to FASTGRAPH are quite impressive. Basically, the cost of interprocessor communication is reduced by a factor of 4 in problem *P1* (regular triangulation) and by a factor of 7 in problem *P2* (irregular triangulation). The superposition of our mapper to FG results in an improvement factor of 8 for *P1* and in an improvement factor of 14 for *P2*. Table 9 summarizes the best performances for cases *T1.1* and *T2.1*.

Case	VPR	Tot. time	Comp. time	Comm. time	Comm./Tot.
T1.1	1	13.4 sec	12.1 sec	1.32 sec	10%
T2.1	1	12.9 sec	12.1 sec	0.82 sec	6%

Table 9 : Performance results on an 8K CM-2
First-order algorithm - **Our mapper + FASTGRAPH**

Finally, the performance of the proposed parallel algorithms on the CM-2 is compared to that of a highly vectorized CRAY-2 version where scatter/gather manipulations are preprocessed with a coloring scheme. The CRAY-2 code is identical to that of the CM-2 except for the fact that no redundant flux computation is performed; all fluxes are computed on an *edge-by-edge* basis. Performance results are compared for a 16K CM-2 using 32-bit arithmetic, our mapper and FASTGRAPH, and a mono processor CRAY-2 using 64-bit arithmetic (see Table 10).

Case	VPR	CM-2 Total time	CRAY-2 Time
T1.2	1	112 sec	204 sec
T2.2	1	110 sec	130 sec

Table 10 : Performance comparisons : 16K CM-2/CRAY-2
Second-order algorithm
CM-2 : **Our mapper + FASTGRAPH**
CRAY-2 : Highly vectorized

By comparing the respective costs of test case *T2.2* in Tables 3 and 10, it can be seen how crucial is the problem of the optimization of the communication part when one want to get the best efficiency for finite element computations on fully unstructured grids. Moreover, the above results demonstrate that for a VPR as small as one, a 16K CM-2 is at least as fast as a mono processor CRAY-2, and in some cases twice as fast. However as stated earlier, these results are for a single precision CM-2 (which is the only machine that was available to us).

5 Conclusion

In this report, we have described the implementation on the Connection Machine CM-2 of a parallel mixed finite volume/finite element method for solving the two-dimensional Euler equations on fully unstructured grids. Second-order accuracy is achieved through a Monotonic Upwind Scheme for Conservation Laws (MUSCL) technique. The resulting semi-discrete equations are time integrated with an explicit Runge-Kutta method.

Even though the results of our previous study on massively parallel Computational Fluid Dynamics (see [15]) have shown that the CM-2 was well suited to structured computations, it was not obvious that this will yet be true in the more general case of unstructured computations. Moreover, the mesh irregularities are now inhibiting the use of the fast NEWS communication mechanism. Therefore, if one want to obtain the best performance when using a massivelly parallel computer for such computations, it becomes essential to reduce the communication costs. For this purpose, we have developped a strategy for mapping thousands of processors onto an unstructured grid. Its key elements are given by the selection of an appropriate parallel data structure, the partitioning of a given unstructured grid into subgrids. and the mapping of each individual processor onto an entity of these subgrids. Whenever the communication patterns are compiled during the first iteration or time step, the total time elapsed in interprocessor communication using the router is drastically reduced to represent in some cases only 6% of the total CPU time of the simulation.

The performance of the proposed massively parallel schemes is assessed on various CM-2 sizes for various mesh densities. Performance comparisons of the presented solution algorithms with vectorized versions where scatter/gather manipulations are treated with a coloring technique indicate that a 16K CM-2 with single precision floating point arithmetic is at least as fast as one processor CRAY-2 and in some cases twice as fast, and this for a virtual processor ratio (VPR) as small as one.

Finite element methods on unstructured grids are well suited to the simulations of flows around or inside complex geometries such as the ones involved in real life problems. On the other part, the Navier-Stokes equations represent the more general model for Computational Fluid Dynamics as they introduce diffusive and heat conducting effects. Future extensions of our work will include the resolution of full Navier-Stokes equations; according to our choice for the data parallel structure (the cell) we can now and henceforth say that no more communication steps will be needed to take into account the diffusive fluxes. Therefore the resulting algorithm is suppose to be more efficient.

References

[1] BOKHARI S. H., *On the Mapping Problem*, IEEE Trans. Comp., Vol C-30, No 3, pp. 207-214, (1981).

[2] CLARY J. S. - HOWELL G. A. - KARMAN S. L., *Benchmark Calculations with an Unstructured Grid Flow Solver on a SIMD Computer*, ACM pp. 32-41, (1989).

[3] DAHL E. D., *Mapping and Compiled Communication on the Connection Machine System*, Proceed. of Distr. Mem. Comp. Conf., Charleston, (1990).

[4] EGOLF T. A., *Scientific Applications of the Connection Machine at the United Technologies Research Center*, Scient. Applic. of the Connection Machine, ed. by SIMON H., pp. 38-63, (1988).

[5] FARHAT C. - CRIVELLI L., *A general Approach to Nonlinear Finite Element Computations on Shared Memory Multiprocessors*, Comp. Meth. Appl. Mech. Eng., Vol 72, No 2, pp. 153-172, (1989).

[6] FARHAT C., *A Simple and Efficient Automatic Finite Element Mesh Domain Decomposer*, Comp. and. Struct., Vol 28, No 5, pp. 579-602, (1988).

[7] FARHAT C. - SOBH N. - PARK K. C., *Transcient Finite Element Computations on 65536 Processors : The Connection Machine*, Intern. Journ. Numer. Meth. Eng., Vol 30, pp. 27-55, (1990).

[8] FARHAT C., *On the Mapping of Massively Parallel Processors Onto Finite Element Graphs*, Comp. and. Struct., Vol 32, No 2, pp. 347-354, (1989).

[9] FARHAT C. - SOBH N. - PARK K. C., *Dynamic Finite Element Simulations on the Connection Machine*, Scient. Applic. of the Connection Machine, ed. by SIMON H., pp. 217-233, (1988).

[10] FEZOUI L. - DERVIEUX A., *Finite Element Non Oscillatory Schemes for Compressible Flows*, Comp. Math. and Applic. 8th France-U.S.S.R.-Italy Joint Sympos. Pavia, (1989).

[11] FEZOUI L. - STEVE H., *Décomposition de Flux de van Leer en éléments finis*, INRIA Report No 830, (1988).

[12] FLOWER J. W. - OTTO S. W. - SALAMA M. C., *A Preprocessor for Irregular Finite Element Problems*, CalTech/JPL Report C3P-292, (1986).

[13] HAMMOND S. - SCHREIBER R., *Mapping Unstructured Grid Problems to the Connection Machine*, RIACS Technical Report 90.22, (1990).

[14] JESPERSEN D. - LEVIT C., *A Computational Fluid Dynamics Algorithm on a Massively Parallel Computer*, AIAA Paper 89-1936-CP, AIAA 9th Comp. Fluid Dynam. Conf, pp. 79-88, (1989).

[15] LANTERI S. - FARHAT C. - FEZOUI L., *Structured Compressible Flow Computations on the Connection Machine*, INRIA Report No 1322, (1990).

[16] LONG L. N., *A Three-Dimensional Navier-Stokes Method for the Connection Machine*, Scient. Applic. of the Connection Machine, ed. by SIMON H., pp. 64-93, (1988).

[17] ROE P. L., *Approximate Riemann Solvers, Parameters Vectors and Difference Schemes*, Journ. of Comp. Phys., **43**, pp. 357-371, (1981).

[18] STEGER J. - WARMING R.F., *Flux vector splitting for the inviscid gas dynamic with applications to finite-difference methods*, Journ. of Comp. Phys., **40**, (2), pp. 263-293, (1981).

[19] FEZOUI L. - STOUFFLET B., *A Class of Implicit Upwind Schemes for Euler Simulations with Unstructured Meshes*, Journ. of Comp. Phys., **84**, pp. 174-206, (1989).

[20] VAN LEER B., *Flux-Vector Splitting for the Euler Equations*, Lect. Notes in Phys., Vol 170 (1982).

[21] VAN LEER B., *Towards the Ultimate Conservative Difference Scheme V : a Second-Order Sequel to Gudonov's Method*, Journ. of Comp. Phys., Vol 32, (1979).

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399