Abstract:

We are studying here the numerical simulation of high Reynolds number internal, 2-D flow in a convergent-divergent nozzle, for a compressible, viscous fluid.An implicit finite-difference scheme is used to solve the parabolic approximation of the Navier-Stokes equations, so that the time steps are not severely limited by the small grid sizes needed for the computation of the vicous effects. After the resolution of this problem on a serial computer, we describe the first steps of the parallelization of this problem on a Hypercube (parallel version of the ADI method [7,10]).

Implicit Finite-Difference Simulation of an Internal Flow in a Nozzle: an Example of a Physical Application on a Hypercube

Pierre Porta

Research Report YALEU/DCS/RR-553
August 1987

This work was supported by ONR grant N00014-86-K-0310

1. Introduction:

The resolution of the fluid flow in a nozzle is a problem which has been studied by several authors on sequential [3,4,9,15] and vector [14] computers. At the present time, no complete study of a fluid dynamic problem has been made on a distributed memory parallel machine such as the iPSC Intel Hypercube, and we propose here to study the parallelization of this physical problem and its corresponding speedup on such a machine.

A general technique of resolution of the Navier-Stokes equations [12,13] mainly based on the combination of a general curvilinear coordinate transformation and an implicit method has given some good results. It has already been experimented on the ILLIAC IV [8] and will be used here.

The physical domain (the domain inside the nozzle) is first transformed into a rectangular domain by the resolution of a pseudo-elliptic equation with Dirichlet conditions at the boundaries (Thompson et al method [16]).

The Navier-Stokes equations are then solved in this domain with the Beam-Warming implicit method [1], which leads to the computation of two sets of linear systems (ADI method).

The modelization of the physical problem is descibed in section II, and the numerical generation grid in section III. The boundary and initial conditions are traited in section IV, and we deal with the turbulence modelization in section V. The numerical algorithm is explained in section VI, the results and their discussion appear in section VII. Finally in section VIII, we discuss the parallelization of this problem on a Hypercube.

2. Modelization:

2.1. Navier-Stokes equations:

The motion inside the nozzle (see figure 1) is governed by the unsteady averaged Navier-Stokes equations in the inertial cartesian coordinates (x,y,t). These equations written in the non-dimension form and with the strong conservation law form [12,17] are given by:

$$\partial_t q + \partial_x e + \partial_y f = Re^{-1}(\partial_x g_1 + \partial_y g_2) \tag{1}$$

with:

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ en \end{pmatrix}, e = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(en+p) \end{pmatrix}, f = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(en+p) \end{pmatrix}, g_1 = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ g_{14} \end{pmatrix}, g_2 = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ g_{24} \end{pmatrix} \tag{2}$$

where:

$$\begin{cases} \tau_{xx} = (\lambda + 2\mu)u_x + \lambda v_y \\ \tau_{xy} = \tau_{yx} = \mu(u_y + v_x) \\ \tau_{yy} = (\lambda + 2\mu)v_y + \lambda u_x \\ g_{14} = u\tau_{xx} + v\tau_{xy} + kPr^{-1}(\gamma - 1)^{-1}\partial_x a^2 \\ g_{24} = u\tau_{xy} + v\tau_{yy} + kPr^{-1}(\gamma - 1)^{-1}\partial_y a^2 \end{cases}$$

and:
ρ: density.
u, v: cartesian velocity components.
en: total energy.
p: pressure.
a: sound speed ($a = \sqrt{\gamma RT}$).
T: temperature.
γ: ratio of specific heats (dry air: $\gamma = 1.4$).
μ: dynamic viscosity.
λ: taken with the Stokes hypothesis $\lambda = -(2/3)\mu$. (3)
k: coefficient of thermal conductivity.
R: gas constant.
U: modulus of the velocity ($U = \sqrt{u^2 + v^2}$).
D: specific length.
c_p: specific heat.
Re: Reynolds number ($Re = (UD\rho)/\mu$).
Pr: Prandtl number ($Pr = (\mu c_p/k)$.
M: Mach number ($M = U/a$).

The equations (1) and (2) are completed by the equation defining the pressure:

$$p = (\gamma - 1)[en - 0.5\rho(u^2 + v^2)] \tag{4}$$

As our nozzle is symmetric about the central axis y=0, we can restrict our study to the upper side of the nozzle.

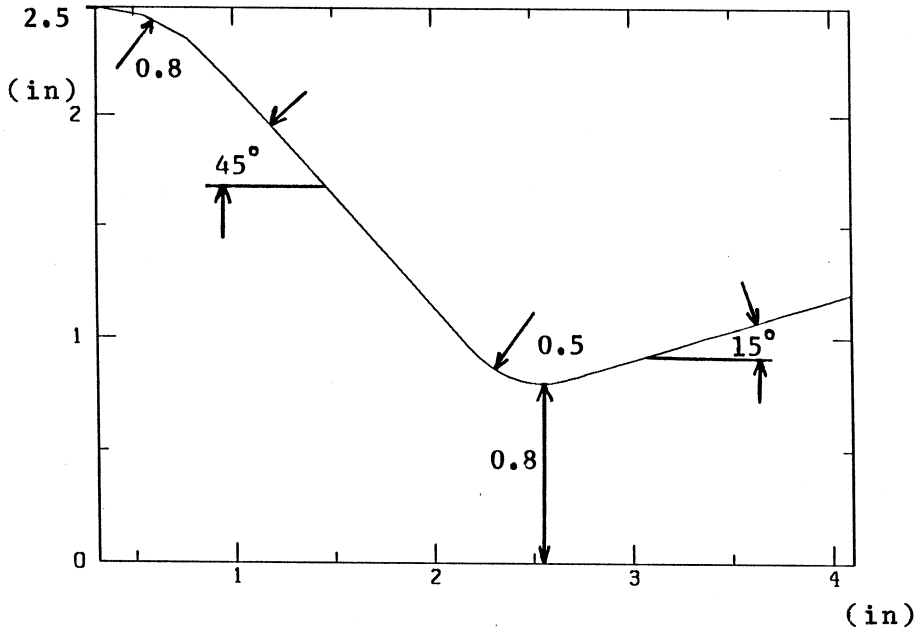

Figure 1: $45^0 - 15^0$ Convergent-Divergent Nozzle

2.2. Transformed Navier-Stokes equations:

2.2.1. General form:

The resolution of equations (1) in the region (R), closed by the boundary (S) (see figure (1)) can be greatly simplified, by transforming these equations (1) to a new body-fitted curvilinear coordinate system [15,16].
More over, if we take for the dependent variables, the Cartesian velocity components, the basis equations (1) can be transformed under the independent-time invertible transformation:

$$\begin{cases} \xi = \xi(x,y) \\ \eta = \eta(x,y) \\ \tau = t \end{cases} \Longleftrightarrow \begin{cases} x = x(\xi,\eta) \\ y = y(\xi,\eta) \\ t = \tau \end{cases} \quad (5)$$

to an arbitrary curvilinear coordinate space (ξ,η,τ) while maintained the strong conservation law form of eq (1) [8,12,17]. The next chapter will present the numerical generation of the curvilinear space (ξ,η,τ), refered as the computational domain.
The resulting equations are given by:

$$\partial_\tau \hat{q} + \partial_\xi \hat{e} + \partial_\eta \hat{f} = Re^{-1}[\partial_\xi(J^{-1}(\xi_x g_1 + \xi_y g_2)) + \partial\eta(J^{-1}(\eta_x g_1 + \eta_y g_2))] \quad (6)$$

In our particular physical case, due to the geometric properties of the nozzle, we can restrain the transformation (ξ,η,τ) to [14]:

$$\begin{cases} \xi = \xi(x) \\ \eta = \eta(x,y) \\ \tau = t \end{cases} \quad (7)$$

The equations are then rewritten as follows:

$$\partial_\tau \hat{q} + \partial_\xi \hat{e} + \partial_\eta \hat{f} = Re^{-1}[\partial_\xi(J^{-1}(\xi_x g_1)) + \partial_\eta(J^{-1}(\eta_x g_1 + \eta_y g_2))] \quad (8)$$

where, the vectors $\hat{q}$, $\hat{e}$ and $\hat{f}$ are expressed by the relations:

$$\begin{aligned} \hat{q} &= J^{-1}q \\ \hat{e} &= J^{-1}(\xi_x e) \\ \hat{f} &= J^{-1}(\eta_x e + \eta_y f) \end{aligned} \quad (9)$$

By the definition of the velocities:

$$\begin{aligned} U &= \xi_x u \\ V &= \eta_x u + \eta_y v \end{aligned} \quad (10)$$

called the contravariant velocity components, and corresponding to the decomposition of the vector velocity along the ξ and η curvilinear coordinates, we get the following expressions:

$$\hat{q} = J^{-1}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ en\end{pmatrix}, \hat{e} = J^{-1}\begin{pmatrix}\rho U\\ \rho uU + \xi_x p\\ \rho vU\\ (en+p)U\end{pmatrix}, \hat{f} = J^{-1}\begin{pmatrix}\rho V\\ \rho uV + \eta_x p\\ \rho vV + \eta_y p\\ (en+p)V\end{pmatrix} \tag{11}$$

J is the transformation Jacobian:

$$J = \xi_x \eta_y = 1/(x_\xi y_\eta) \tag{11}$$

The cartesian derivatives such as u_x are expanded in the (ξ, η) space via chain-rule relation:

$$u_x = \xi_x u_\xi + \eta_x u_\eta \tag{12}$$

And, the metrics ξ_x, η_x, η_y formed from chain-rule extension of x_ξ, y_ξ, y_η are given by the following relations:

$$\begin{aligned}\xi_x &= J y_\eta\\ \eta_x &= -J y_\xi\\ \eta_y &= J x_\xi\end{aligned} \tag{13}$$

2.2.2. Simplified form: the parabolic equations.

Classicaly for high Reynolds number flows, we solve the viscous terms only near the rigid boundaries. We also make the hypothesis of the parabolic approximation that the viscous terms in ξ (along the body) are neglected and only the viscous terms in η are retained.
The equations (8) are then equal to [8,12,15]:

$$\partial_\tau \hat{q} + \partial_\xi \hat{e} + \partial_\eta \hat{f} = Re^{-1}\partial_\eta \hat{g} \tag{14}$$

where:

$$\hat{g} = J^{-1}\begin{pmatrix}0\\ \mu(\eta_x^2+\eta_y^2)u_\eta + (\mu/3)\eta_x(\eta_x u_\eta + \eta_y v_\eta)\\ \mu(\eta_x^2+\eta_y^2)v_\eta + (\mu/3)\eta_y(\eta_x u_\eta + \eta_y v_\eta)\\ [kPr^{-1}(\gamma-1)^{-1}(\eta_x^2+\eta_y^2)\partial_\eta a^2 + \mu(\eta_x^2+\eta_y^2)(u^2+v^2)_\eta/2\\ +\mu/6(\eta_x^2 u_\eta^2 + \eta_y^2 v_\eta^2 + 2\eta_x\eta_y(uv)_\eta)]\end{pmatrix} \tag{15}$$

Note that unlike boundary-layer theory, the pressure p can vary through the viscous layer, and all the inertial terms of the normal momentum equation are retained.

3. Numerical grid generation:

By the means of the generation of (ξ, η, τ), the physical domain is transformed into a rectangular domain with a square grid ($\Delta\xi = \Delta\eta = 1$). All the computations are then done in this rectangular domain (the computational domain) [16], using more simple finite-difference operators for the different derivatives of the Navier-Stokes equations.
Note that the other advantage of this transformation is that the boundary surfaces (here the wall of the nozzle) are mapped into rectangular sufaces, so that the boundary conditions can be computed more easily and more accurately.
Also, this transformation allows grid point clustering near the walls, such that the viscous effects are well computed (see figures $2-3$).

The numerical method of generation of the grid is given here by the Thompson scheme. In this method, the grid in the physical domain is determined by the solution of a Laplace or a Poisson equation [12,16].

$$\partial^2\xi/\partial x^2 + \partial^2\xi/\partial y^2 = 0 \text{ or } f(\xi,\eta) \tag{16.a}$$

$$\partial^2\eta/\partial x^2 + \partial^2\eta/\partial y^2 = 0 \text{ or } f(\xi,\eta) \tag{16.b}$$

where the values of ξ and η are arbitrarily fixed on the boundaries.
Pratically, the equations (16) are transformed into the computational domain and the generation of the grid is then given by the resolution of the pseudo -elliptic equations:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \text{ or } -J^2(\xi_{xx}y_\xi + f(\xi,\eta)y_\eta) \tag{17.a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \text{ or } -J^2(\xi_{xx}y_\xi + f(\xi,\eta)y_\eta) \tag{17.b}$$

In our case, we have arbitrary chosen $x = x(\xi)$ as an exponential law centered in the throat.
The coefficients of the equation (17.b) to solve are [14]:

$$\alpha = y_\eta^2,\ \beta = y_\xi y_\eta,\ \gamma = x_\xi^2 + y_\xi^2, \text{ and } \xi_{xx} = -x_{\xi\xi}/x_\xi^3 \tag{18}$$

The boundary values of y are known and correspond to the values of the desired mesh point on the boundaries of the physical domain.
The function $f(\xi,\eta)$ is determined at every mesh point (ξ,η), and its function is to control the spacing between the grid points in the interior of the physical domain. Here, we have chosen the function given by Strikwerda in [14].
The following figures $(2-3)$ show the grid lines $\xi = \xi_{const}$, and $\eta = \eta_{const}$ in the physical domain for the Laplace and Poisson equations.

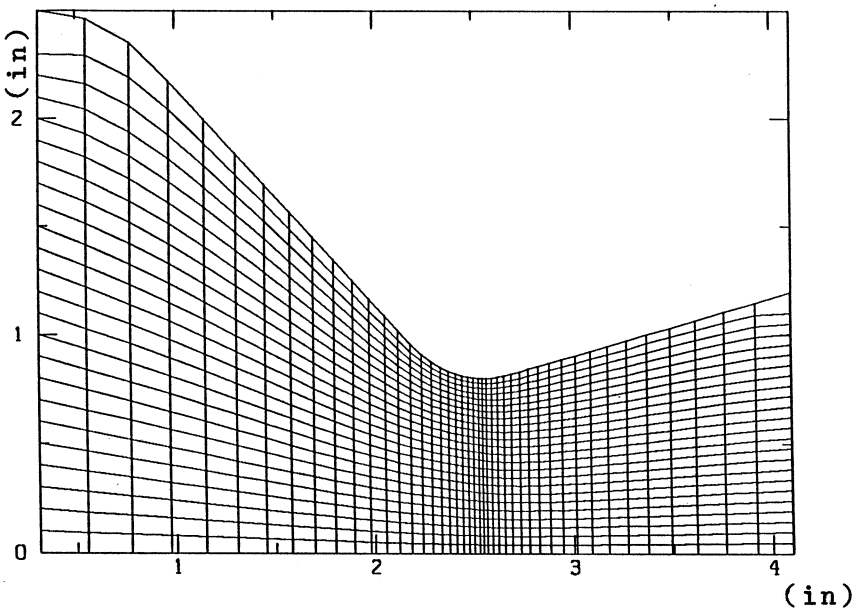

Figure 2: Grid lines (ξ, η) in the physical domain (for the Laplace equation).

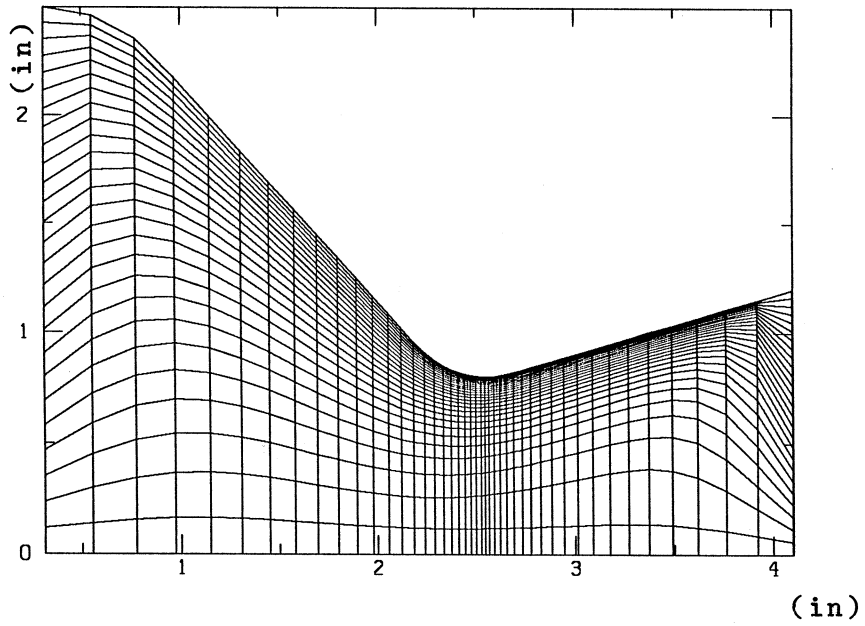

Figure 3: Grid lines (ξ, η) in the physical domain (for the Poisson equation).

4. Boundary conditions- Initial condition:

The equations (14) have to be solved in the computational domain defined before, subject to different conditions on the boundaries.
The curvilinear coordinate system (ξ,η) is defined such that the inflow and outflow boundaries are two ξ-coordinate lines, and the wall and the symmetric axis are two η-coordinate lines.

4.1. Wall boundary condition:

At the nozzle wall, we have for the velocity the no-slip condition:

$$u = v = 0 \tag{19.a}$$

and, for the temperature we can take the adiabatic condition:

$$\vec{n} \cdot \nabla T = 0 \tag{19.b}$$

with: $\vec{n}$ the vector normal to the surface. In our case, for a surface $\eta = \eta_{const}$, we have:

$$\vec{n} = \nabla \eta \tag{19.c}$$

and equation (19.b) becomes:

$$\nabla \eta . \nabla T = 0 \tag{19.d}$$

Under the parabolic approximation T_ξ is neglected and we have [15]:

$$(\nabla \eta \cdot \nabla \eta) T_\eta = 0 \tag{19.e}$$

4.2. Symmetric axis y=0:

Because of the symmetry of the nozzle, we consider the axis y=0 as one of the boundaries of this problem (see figure (1)).
In order to limit the overloading of the code, we just limit ourselves to a condition on the vertical component of the velocity v :

$$v = 0 \tag{20}$$

as it is an odd function of y.

4.3. Upstream inflow boundary:

The pressure p and the temperature T are constant during all the computation, and are equal to their stagnation correspondants i-e:

$$p = p_o \tag{21.a}$$

$$T = T_o \tag{21.b}$$

The components u and v of the velocity are given by the relation:

$$u = v \tan \theta(y) \tag{21.c}$$

where $\theta(y)$ is a function given by Holder et al [5].

4.4. Downstream outflow boundary:

For a supersonic flow ($M > 1$), no boundary conditions are required. The variables are determined by extrapolation from the interior points.

4.5. Initial condition:

All the variables at the beginning of the computation were computed under the 1-D approximation method (isentropic flow, perfect fluid).
In such an approach, the values of the variables at a specified section of the nozzle are given by the geometric ratio of the aera of the section over the throat section, and by the chosen stagnation conditions [6,11].
In particular, we have taken here:

$$p_o = 6.2\ 10^5 N/m^2$$

$$T_o = 300K$$

And the Reynolds number for this initial condition was then equal to: $Re = 1.5\ 10^6$

5. Turbulence Modelling:

For $Re >> 1$, the domain is divided into two main regions: the boundary-layer and the core. The modelization of the turbulence has to separate these parts:

5.1. Internal and external wall boundaries layers:

The boundary layer flow along the wall is modelized by using a two-layers eddy viscosity model. Under this approach, the turbulent stresses τ_t in the boundary-layer are modeled in terms of the eddy viscosity μ_t by:

$$\tau_t = \mu_t \partial U / \partial y \tag{22}$$

where U is the velocity component parallel to the wall, and y is the coordinate normal to and measured from the wall [2,15].

For the inner layer, the eddy viscosity μ_t is given by the Van Driest formulation with Cebeci damping:

$$\mu_t = 0.016\rho y^2[1 - exp(y^+/A^+)]^2 \partial U/\partial y \tag{23}$$

where the damping constant A^+ is function of the pressure gradient parameter P^+ as:

$$A^+ = 26(1 + 11.8P^+)^{-1/2} \tag{24}$$

where:

$$P^+ = \mu/(\rho^2 u_\tau^3).dp/ds \tag{25}$$

with the "friction" velocity given by:

$$u_\tau = \sqrt{\tau_w/\rho_w} \tag{26}$$

where the index w means that the values are taken at the wall.
For the outer layer, the eddy viscosity is computed from the Clauser approximation:

$$\mu_\tau = 0.0168\rho \int_0^\delta (U_c - U)\, dy \tag{27}$$

where:
U_c : core flow velocity at the edge of the boundary layer.
δ : thickness of the boundary layer.

Pratically, the boundary-layer thickness δ is defined here as the distance from the wall so that the velocity U approches the corresponding value of the freestream.

5.2. Core flow:

The effect of the turbulence is also approximated in this part from the modelization of an eddy viscosity, constant for all the core flow and deducted from the jet theory [15]:
we have here:

$$\mu_t = 0.0256 b \rho U_o \tag{28}$$

with:
U_o: the average inlet velocity.
b: the nozzle inlet radius.

Finally, a constant turbulent Prandtl number of $Pr_t = 0.9$ is used in the energy equation, which leads to the following definition of the turbulent eddy conductivity k_t:

$$Pr_t = \mu_t cp/k_t = 0.9 \tag{29}$$

6. Numerical Method:

We solve the equations (14) in the computational domain (ξ, η) subject to the boundary conditions seen before, with the implicit delta-form, approximate-factorization, Beam-Warming algorithm.

An implicit numerical method is employed here in order to avoid the severely restrictive stability conditions of an explicit method, when small grid spacings are used. Such a situation is needed near the wall for an accurate computation of the viscous effects.

In the basic Beam-Warming algorithm, the spatial derivative terms in the Navier-Stokes equations are approximated by standard second-order accurate central-difference operators, and the implicit θ-method of Richtmyer and Morton is taken for the time differencing [8].

The computation of the boundary points can be computed directly by the numerical resolution of the equations (see [15]) or by extrapolation from the interior points at the end of each time step (see [12]). The second case degrades the time accuracy on the boundary to a zero-order but gives a more simple scheme.

The method proposed here is to mix these two approaches: the variables at the inflow boundary are updated by extrapolation (due to its particular conditions), and for the other boundaries, we use a direct numerical resolution. This approach will give us the correct steady state solution but will need more time steps.

6.1. Time Differencing:

For both the interior and boundary points, we define the same time differencing. The equations of Navier-Stokes in their final form (14) are rewritten here as:

$$\partial_\tau \hat{q} = -[\partial_\xi \hat{e} + \partial_\eta \hat{f} - Re^{-1}\partial_\eta \hat{g}] = \hat{r} \tag{30}$$

Using n for the time index and h for the time step, we apply the Richtmyer and Morton method, and we obtain [8]:

$$\hat{q}^{n+1} = \hat{q}^n + h[(1-\theta)\hat{r}^{n+1} + \theta \hat{r}_n] \tag{31}$$

This method is first-order accurate in time for $\theta = 0$ (Implicit Euler method), and is second-order accurate in time for $\theta = 1/2$.

Since we seek only the asymptotic steady state solution, we can employ the first-order accurate in time method. The accuracy of the solution is given by the spatial difference operators [1].

So we have:

$$\hat{q}^{n+1} = \hat{q}^n + h\hat{r}^{n+1} \tag{32}$$

In order to define the non-linear term $\hat{r}_{n+1}$, we must locally linearized the terms $\hat{e}$, $\hat{f}$ and $\hat{g}$ in terms of $\hat{q}$. This is done by using the Taylor series expansions:

$$\hat{e}^{n+1} = \hat{e}^n + E^n(\hat{q}^{n+1} - \hat{q}^n) + O(h^2).$$

$$\hat{f}^{n+1} = \hat{f}^n + F^n(\hat{q}^{n+1} - \hat{q}^n) + O(h^2). \tag{33}$$

$$\hat{g}^{n+1} = \hat{g}^n + G^n(\hat{q}^{n+1} - \hat{q}^n) + O(h^2).$$

where E, F and G are the flux Jacobian matrices:

$$E = \partial\hat{e}/\partial\hat{q},\ F = \partial\hat{f}/\partial\hat{q}\ ,\ G = \partial\hat{g}/\partial\hat{q} \tag{34}$$

defined more precisely as following:

$$E = [E_{i,j}] = \partial\hat{e}_i/\partial\hat{q}_j,\ F = [F_{i,j}] = \partial\hat{f}_i/\partial\hat{q}_j\ ,\ G = [G_{i,j}] = \partial\hat{g}_i/\partial\hat{q}_j \tag{35}$$

By definition, the flux vectors $\hat{e}$ and $\hat{f}$ are both linear combinations of e and f (see eq(9)) i.e:

$$\hat{e} = J^{-1}(\xi_x e)$$

$$\hat{f} = J^{-1}(\eta_x e + \eta_y f) \tag{36}$$

so the inviscid flux Jacobian matrices E and F are expressed as follows:

$$E = \partial\hat{e}/\partial\hat{q} = \xi_x \partial e/\partial q$$

$$F = \partial\hat{f}/\partial\hat{q} = \eta_x \partial e/\partial q + \eta_y \partial f/\partial q \tag{37}$$

The E or F matrices are then given by [8,12]:

$E, F =$

$$\begin{pmatrix} 0 & K_1 & K_2 & 0 \\ K_1\phi^2 - u\theta & \theta - K_1(\gamma-2)u & K_2 u - (\gamma-1)K_1 v & K_1(\gamma-1) \\ K_2\phi^2 - v\theta & K_1 v - K_2(\gamma-1)u & \theta - K_2(\gamma-2)v & K_2(\gamma-1) \\ \theta(2\phi^2 - \gamma(en/\rho)) & [K_1[\gamma(en/\rho) - \phi^2] - (\gamma-1)u\theta] & [K_2(\gamma(en/\rho) - \phi^2) - (\gamma-1)v\theta] & \gamma\theta \end{pmatrix}$$

(38)

where:

$$\phi^2 = 0.5(\gamma-1)(u^2 + v^2)$$

$$\theta = K_1 u + K_2 v \tag{39}$$

with the following definition of the coefficients K_1 and K_2:

$$\text{for } E\text{: } K_1 = \xi_x,\ K_2 = 0 \tag{40.a}$$

$$\text{for } F\text{: } K_1 = \eta_x,\ K_2 = \eta_y \tag{40.b}$$

And for the viscous flux Jacobian term, we have:

$$G^n = J^{-1}\begin{pmatrix} 0 & 0 & 0 & 0 \\ g_{21} & \alpha_1\partial_\eta(1/\rho) & \alpha_2\partial_\eta(1/\rho) & 0 \\ g_{31} & \alpha_2\partial_\eta(1/\rho) & \alpha_3\partial_\eta(1/\rho) & 0 \\ g_{41} & g_{42} & g_{43} & \alpha_4\partial_\eta(1/\rho) \end{pmatrix} J \tag{41}$$

where:

$$\begin{pmatrix} g_{21} \\ g_{31} \end{pmatrix} = -Q \cdot \begin{pmatrix} \partial_\eta(u/\rho) \\ \partial_\eta(v/\rho) \end{pmatrix}, \text{ with } Q = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_2 & \alpha_3 \end{pmatrix} \tag{42}$$

and:

$$\begin{aligned} \alpha_1 &= \mu[(4/3)\eta_x^2 + \eta_y^2] \\ \alpha_2 &= (\mu/3)\eta_x\eta_y \\ \alpha_3 &= \mu[\eta_x^2 + (4/3)\eta_y^2] \\ \alpha_4 &= \gamma k Pr^{-1}(\eta_x^2 + \eta_y^2) \end{aligned} \tag{43}$$

By means of the local linearizations (see eqs (33)) the equation (32) is then rewritten:

$$[I + h(\partial_\xi E^n + \partial_\eta F^n - Re^{-1}\partial_\eta G^n)]\Delta\hat{q}^n = h\hat{r}^n \tag{44}$$

with:

$$\Delta\hat{q}^n = \hat{q}^{n+1} - \hat{q}^n \tag{45}$$

where the linear operator notation is to be interpreted as following:

$$(\partial_\xi E^n)\Delta\hat{q}^n = \partial_\xi(E^n\Delta\hat{q}^n) \tag{46}$$

The left hand side of equation (44) can be factorized into a product of two one-dimensional operators, with the same order of accuracy [1].
This results in the factorized form of equation (44):

$$([I + h\partial_\xi E^n][I + h(\partial_\eta F^n - Re^{-1}\partial_\eta G^n)])\Delta\hat{q}^n = h\hat{r}^n. \tag{47}$$

which is said to be the "delta-form", because the left hand side contains the factor $\Delta\hat{q}^n$.
The r.h.s of equation (44) is defined as:

$$\hat{r}^n = -[\partial_\xi\hat{e}^n + \partial_\eta\hat{f}^n - Re^{-1}\partial_\eta\hat{g}^n]. \tag{48}$$

Finally, the vector solution $\hat{q}^{n+1}$ is given by the following ADI sequence:

$$[I + h\partial_\xi E^n]\Delta\hat{q}^{*n} = h\hat{r}^n \tag{49.a}$$

$$[I + h(\partial_\eta F^n - Re^{-1}\partial_\eta G^n)]\Delta\hat{q}^n = \Delta\hat{q}^{*n} \tag{49.b}$$

$$\hat{q}^{n+1} = \hat{q}^n + \Delta\hat{q}^n \tag{49.c}$$

For convenience, we have omitted the spatial subscript notations, all the terms were taken here at the point (i, j) of the computational domain.
But from now on, we will use the spatial subscripts again.

Note that by the choice of a central-difference scheme for the space derivatives, each step of the ADI sequence (49) involves the solution of a linear system of equations having a block-tridiagonal coefficient matrix.

6.2. Space Differencing:

6.2.1. Interior points:

The resolution of the ADI sequence $(49.a, 49.b, 49.c)$ is completed by the choice of the finite difference operators δ for the spatial derivatives ∂_ξ and ∂_η.

We use here to approximate the convective derivatives the standard central-difference second order accurate operator. For example we have for the first derivative of equation $(49.a)$:

$$\delta_i E_{i,j} = (\partial E_{i,j}/\partial\xi) = (E_{i+1,j} - E_{i-1,j})/2\Delta\xi \tag{50}$$

where by definition of the grid in the computational domain: $\Delta\xi = \Delta\eta = 1$. From now on, the increments $\Delta\xi$ and $\Delta\eta$ will be replaced by 1 in all the spatial difference formulas.
First we get the expressions:

$$\delta_i E_{i,j} = (E_{i+1,j} - E_{i-1,j})/2$$

$$\delta_j F_{i,j} = (F_{i,j+1} - F_{i,j-1})/2. \tag{51}$$

From the general form of the viscous term written as follows:

$$\partial[(\alpha_{i,j}\partial\beta_{i,j}/\partial\eta)]/\partial\eta \tag{52}$$

we see that we have to approximate the viscous derivative in a more complicated way.
If, we define the following central-difference operator for a general viscous term h:

$$(\partial h_{i,j}/\partial\eta) \approx h_{i,j+1/2} - h_{i,j-1/2} \tag{53}$$

its application on the viscous term defined in equation (52) gives the generalized three-points central-difference second-order accurate operator:

$$\partial[(\alpha_{i,j}\partial\beta_{i,j}/\partial\eta)]/\partial\eta =$$
$$1/2[(\alpha_{i,j+1} + \alpha_{i,j})\beta_{i,j+1} - (\alpha_{i,j+1} + 2\alpha_{i,j} + \alpha_{i,j-1})\beta_{i,j} + (\alpha_{i,j} + \alpha_{i,j-1})\beta_{i,j-1}] \tag{54}$$

corresponding to the following notation : $\delta'_j \hat{g}_{i,j}$ and $\delta'_j G_{i,j}$
From these definitions, the ADI sequence is rewritten for the interior points as follows:

$$[I + h\delta_i E^n_{i,j}]\Delta\hat{q}^{*n}_{i,j} = h\hat{r}^n_{i,j} \tag{55.a}$$

$$[I + h(\delta_j F^n_{i,j} - Re^{-1}\delta'_j G^n_{i,j})]\Delta\hat{q}^n_{i,j} = \Delta\hat{q}^{*n}_{i,j} \tag{55.b}$$

$$\hat{q}^{n+1}_{i,j} = \hat{q}^n_{i,j} + \Delta\hat{q}^n_{i,j} \tag{55.c}$$

with:

$$\hat{r}^n_{i,j} = -[\delta_i \hat{e}^n_{i,j} + \delta_j \hat{f}^n_{i,j} - Re^{-1}\delta'_j \hat{g}^n_{i,j}] \tag{56}$$

6.2.2. Boundary grid points:

At the inflow boundary, both the pressure p, the temperature T, and the density ρ remain constant during all the computation. To take advantage of this characteristic, we assume that for all the points of this boundary, the increment $\Delta\hat{q}^n$ is equal to 0, during the computation of the ADI sequence. Then u, v, and en are updated by extrapolation from the interior points at the end of each time step.

For the other boundaries, we used a method of resolution quasi-similar to that of the interior points. In this case, we have to adapt certain derivative operators in order to use only the points of the computational domain.

We will employ here the forward difference operator Δ, and the backward difference operator ∇ where:

$$\begin{aligned}
\Delta_i \hat{e}_{i,j} &= \hat{e}_{i+1,j} - \hat{e}_{i,j} \\
\Delta_j \hat{e}_{i,j} &= \hat{e}_{i,j+1} - \hat{e}_{i,j} \\
\nabla_i \hat{e}_{i,j} &= \hat{e}_{i,j} - \hat{e}_{i-1,j} \\
\nabla_j \hat{e}_{i,j} &= \hat{e}_{i,j} - \hat{e}_{i,j-1}
\end{aligned} \tag{57}$$

This approach has been described in detail in [15], and is presented below in a synthetic way:

6.2.2.1. Outflow boundary (for supersonic condition):

Only the first step of the ADI sequence (49.a) has to be modified. The spatial derivatives along the ξ-axis are approximated by the backward difference:

$$\nabla_i E_{i,j} = E_{i,j} - E_{i-1,j} \tag{58}$$

Due to the link between the finite-difference and finite volume methods, we associate each point of the outflow boundary with a half-cell situated $\Delta\xi/2$ before this boundary. The time derivative is applied at the centroid of this half-cell (placed $\Delta\xi/4$ from the outflow boundary). The first step of the ADI sequence is then rewritten [15]:

$$[I + h\nabla_i(E^n_{i,j} - \alpha I)]\Delta\hat{q}^{*n}_{i,j} = h\hat{r}^{n'}_{i,j} \tag{59}$$

where in order to keep the second order accuracy in space, the coefficient α is chosen as follows:

$$\alpha = 1/(4h) \tag{60}$$

with the r.h.s of eq (59) given by:

$$\hat{r}^{n'}_{i,j} = -[\nabla_i \hat{e}^n_{i,j} + \delta_j \hat{f}^n_{i,j} - Re^{-1}\delta'_j \hat{g}^n_{i,j}] \tag{61}$$

6.2.3. Wall boundary:

The method here is to approximate the spatial derivative along the η-axis by the backward difference operator ∇ for the convective derivative, and by ∇' for the viscous derivative, where:

$$\nabla'_j G^n_{i,j} = 2(G^n_{i,j} - G^n_{i,j-1/2}) \tag{62}$$

The ADI sequence becomes:

$$[I + h\delta_i E^n_{i,j}]\Delta\hat{q}^{*n}_{i,j} = h\hat{r}^{n''}_{i,j} \tag{63.a}$$

$$[I + h(\nabla_j F^n_{i,j} - \alpha I) - hRe^{-1}\nabla'_j G^n_{i,j}]\Delta\hat{q}^n_{i,j} = \Delta\hat{q}^{*n}_{i,j} \tag{63.b}$$

$$\hat{q}^{n+1}_{i,j} = \hat{q}^n_{i,j} + \Delta\hat{q}^n_{i,j} \tag{63.c}$$

with the time derivative taken at the points placed $\Delta\eta/4$ from the wall boundary.
With the same assumption for the r.h.s, we get:

$$\hat{r}^{n''}_{i,j} = -[\delta_i \hat{e}^n_{i,j} + \nabla_j \hat{f}^n_{i,j} - Re^{-1}\nabla'_j \hat{g}^n_{i,j}] \tag{64}$$

6.2.4. Symmetric axis y=0:

The partial derivative along the η-axis is given by the forward difference operator Δ, for the convective derivative:

$$\Delta_j F^n_{i,j} = F^n_{i,j+1} - F^n_{i,j} \tag{65}$$

and by Δ' for the viscous derivative, where:

$$\Delta'_j G^n_{i,j} = 2[G^n_{i,j+1/2} - G^n_{i,j}] \tag{66}$$

The vector solution $\hat{q}_{n+1}$ is then given for the symmetric axis by the equations:

$$[I + h\delta_i E^n_{i,j}]\Delta\hat{q}^{*n}_{i,j} = h\hat{r}^{n'''}_{i,j} \tag{67.a}$$

$$[I + h(\Delta_j F^n_{i,j} - Re^{-1}\Delta'_j G^n_{i,j})]\Delta\hat{q}^n_{i,j} = \Delta q^{*n}_{i,j} \tag{67.b}$$

$$\hat{q}^{n+1}_{i,j} = \hat{q}^n_{i,j} + \Delta\hat{q}^n_{i,j} \tag{67.c}$$

where:

$$\hat{r}^{n'''}_{i,j} = -[\delta_i \hat{e}^n_{i,j} + \Delta_j \hat{f}^n_{i,j} - Re^{-1}\Delta'_j \hat{g}^n_{i,j}] \tag{68}$$

6.2.5. Inflow boundary:

During all the computation, the pressure p, the temperature T, and by the state law for a perfect gas, the density ρ are constant and equal to their respective stagnation values:

$$p = p_o,\ T = T_o,\ \rho = \rho_o \tag{69}$$

The value of u is updated at the end of each time step by extrapolation from the interior points by the mass conservation equation between the first two sections of the nozzle.

Then, the new value of v is computed by the relation:

$$v = u \tan\theta(y) \tag{70}$$

with the function $\theta(y)$ given by Holder et al [5].

Also, from the new values of the components u and v of the velocity, the total energy en is updated by equation (4).

6.3. Smoothing:

Finally, the numerical stability of this method of simulation of a high Reynolds number flow is here controlled by adding to the r.h.s of equation (49.a) a fourth-order dissipation term and to the l.h.s of eqs (49.a) and (49.b) a second-order dissipation term [1,8,12]. The effect is for example to modify the ADI sequence (55.a, 55.b, 55.c) for the interior points to the form:

$$[I + h\delta_i E^n_{i,j} - \varepsilon_{imp} J^{-1}_{i,j}(\nabla_i\Delta_i)J_{i,j}]\Delta\hat{q}^{*n}_{i,j} = h\hat{r}^n_{i,j} - \varepsilon_{exp} J^{-1}_{i,j}[(\nabla_i\Delta_i)^2 + (\nabla_j\Delta_j)^2]J_{i,j}\hat{q}^n_{i,j}. \tag{71.a}$$

$$[I + h(\delta_j F^n_{i,j} - Re^{-1}\delta'_j G^n_{i,j}) - \varepsilon_{imp} J^{-1}_{i,j}(\nabla_j\Delta_j)J_{i,j}]\Delta\hat{q}^n_{i,j} = \Delta\hat{q}^{*n}_{i,j}. \tag{71.b}$$

$$\hat{q}^{n+1}_{i,j} = \hat{q}^n_{i,j} + \Delta\hat{q}^n_{i,j}. \tag{71.c}$$

where the values of the coefficients ε_{exp} and ε_{imp} are controlled by the linear stability conditions.We take here the values:

$$\varepsilon_{exp} = h,\ \varepsilon_{imp} = 3h \tag{72}$$

We proceed in the same way for the wall, the symmetric axis and the outflow boundaries. The operators Δ and ∇ are the forward and backward difference operators defined before (see eqs (57)).

7. Results:

The numerical method described here has been used in the study of the flow of a compressible viscous fluid in a two-dimensional convergent-divergent nozzle (see figure 1).

The numerical generated grid used for this problem is shown in figure 3. The number of grid points was fixed to 1125 in order to reduce the size of the linear systems to be solved on the parallel machine.

The mesh was of size 45×25, where 45 points were used in the horizontal direction and, 25 points in the vertical direction.

The combination of the numerical algorithm, grid mapping and boundary conditions were here as a first step tested on the VAX 8600, before its application on the iPSC Intel Hypercube.

The computations were stopped when the maximum relative change in the Mach number in the throat and downstream region was sufficiently small.

From the initial condition and the inflow boundary conditions, the Reynolds number at the beginning of this simulation was equal to $1.5\ 10^6$, being evaluated on the symmetric axis, using the throat radius as the reference length.

The solution proposed was obtained for a CPU time of about 35 mn.The main properties of this solution are here well caracterised by the representation of the equi-Mach number curbs along the nozzle (see figure 4), and by the ratio of the pressure at the symmetric axis of each section over the stagnation pressure (see figure 5).

We get accurate results comparable to the numerical solution proposed by Cline in [3]. In addition, the capacity of adaptability of the numerical method used here to a large variety of problems gives more power and more interest to this general approach [12].

Also, we can note that by the definition of an ADI sequence, this numerical algorithm has intrinsically a highly parallel character for its application on a distributed memory parallel machine [7].We propose now a first reflexion on the parallelization of this problem.

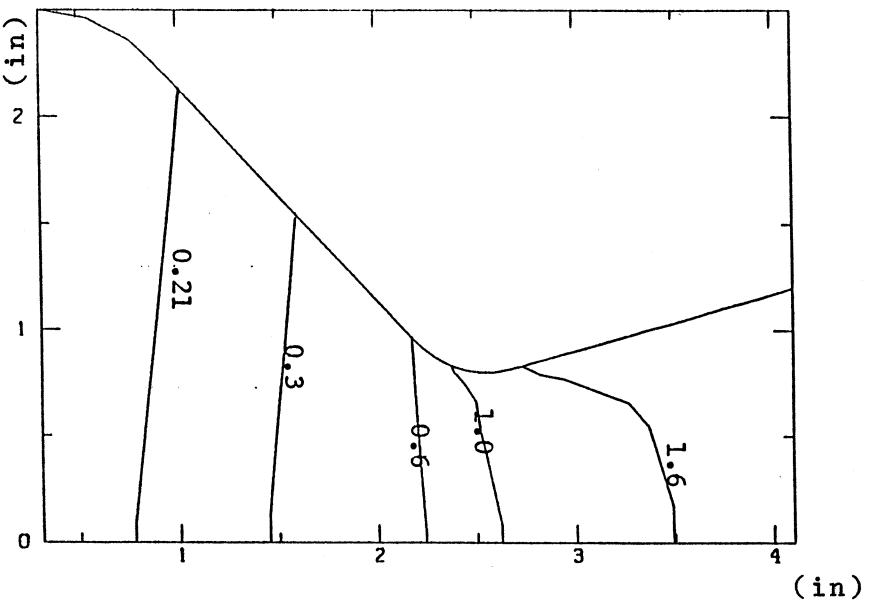

Figure 4: Contour plot of Mach number

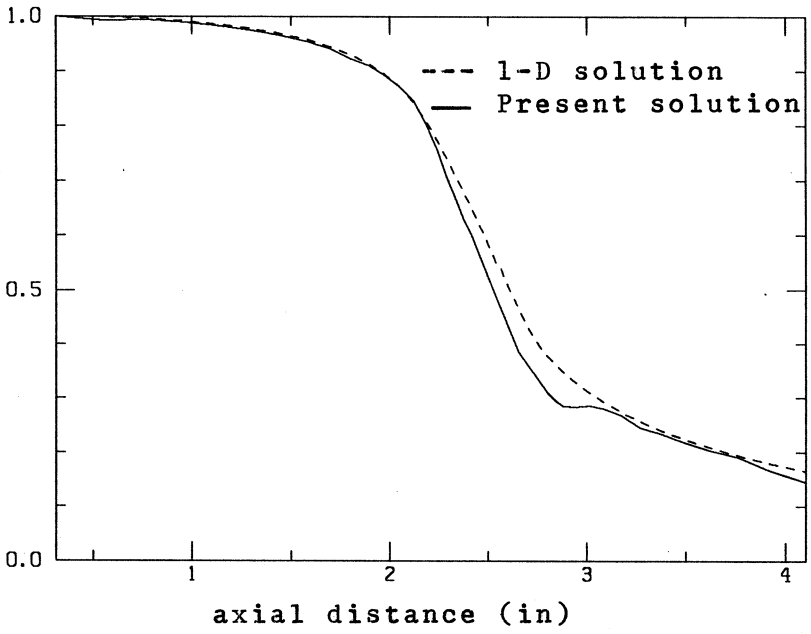

Figure 5: Axis pressure ratio

8. Parallelization:

Let us now study the numerical simulation of this fluid flow on the Hypercube, from the existing sequential code.

In particular, we have seen that the numerical algorithm for the solution of the 2-D Navier-Stokes equations leads to an ADI sequence, which has already been study both theorically and pratically on the Hypercube (see [7,10]).

First, we recall the ADI sequence (see eqs (71)), that we can express here as follows:

$$A^n_{i,j}\Delta\hat{q}^{*n}_{i,j} = b^n_{i,j} \quad \text{for } j = 1,\ldots,25 \tag{73.a}$$

$$C^n_{i,j}\Delta\hat{q}^n_{i,j} = \Delta\hat{q}^{*n}_{i,j} \quad \text{for } i = 2,\ldots,45 \tag{73.b}$$

with the vector solution $\hat{q}^{n+1}_{i,j}$ given by:

$$\hat{q}^{n+1}_{i,j} = \hat{q}^n_{i,j} + \Delta\hat{q}^n_{i,j} \tag{73.c}$$

where:

$$\begin{cases} A^n_{i,j} = I + h\delta_i E^n_{i,j} - \varepsilon_{imp} J^{-1}_{i,j}(\nabla_i\Delta_i)J_{i,j} \\ C^n_{i,j} = I + h(\delta_j F^n_{i,j} - Re^{-1}\delta'_j G^n_{i,j}) - \varepsilon_{imp} J^{-1}_{i,j}(\nabla_j\Delta_j)J_{i,j} \\ b^n_{i,j} = h\hat{r}^n_{i,j} - \varepsilon_{exp} J^{-1}_{i,j}[(\nabla_i\Delta_i)^2 + (\nabla_j\Delta_j)^2]J_{i,j}\hat{q}^n_{i,j} \end{cases} \tag{74}$$

for the definition of $E^n_{i,j}$, $F^n_{i,j}$, $G^n_{i,j}$ of size 4×4 , and $\hat{r}^n_{i,j}$ see chapter VI.

The purpose of this work is to study the parallelization of the specific ADI sequence given before (eqs(73)) for the computational domain of size 44×25 defined in chapter III (the first column of the initial domain of size 45×25, corresponding to the inflow boundary is computed by extrapolation, see chapter VI).

More precisely, we solve the 25 equations (73.a) (respectively 44 equations (73.b)) with i varying from 2 to 45 (respectively with j varying from 1 to 25) which leads to the computation of 25 linear systems of size 176 (44×4) (respectively 44 linear systems of size 100 (25×4)).

Pratically, we use a one dimensional domain decomposition embedded on the Hypercube of $k=2^n$ processors where n is the dimension of the Hypercube.

The solution method chosen here can be described as follows:
For the first half step of the ADI sequence (eqs (73.a)), the computational domain of size 44×25 is decomposed in k horizontal strips, where each strip is assigned to one processor. Every processor has then to solve locally $25/k$ linear systems.

The matrix solution $\Delta\hat{q}^{*n}_{i,j}$ is then transposed among the processors in order to solve the second half step of the ADI sequence (eqs (73.b)), with the maximun level of parallelism, i.e. $44/k$ systems to solve per processor.

Also, since all linear systems are solved locally, and involve banded matrices, they can be solved using a band solver from LINPACK.

The incremental solution $\Delta\hat{q}^{n}_{i,j}$ is moved back to its original distribution for the computation of $\hat{q}^{n}_{i,j}$ by equation (73.c). Finally, we compute the updated values of the matrices $A^{n+1}_{i,j}$, $C^{n+1}_{i,j}$, and the vector $b^{n+1}_{i,j}$. We can then begin a new time step of the ADI sequence (73.a), (73.b) and (73.c).

Such an approach has already been made in the study of the Schrodinger equation by an ADI sequence on the iPSC Intel Hypercube (see [10]) and will be developped further for our physical problem.

9. Conclusion:

An implicit finite-difference serial computer program has been developped to solve the parabolic approximation of the 2-D Navier-Stokes equations for a compressible fluid.

Its application for the simulation of a flow in a convergent-divergent nozzle has provided an accurate solution. Further more, this numerical method, essentially based on an ADI sequence is intrinsically parallelizable, which gives to this general technique more interest in the study of fluid dynamics problem, on a parallel machine of the Hypercube family.

The complete resolution of this problem on the Hypercube is currently studied and will be detailed in a next paper.

Acknowledgement:

I would like to thank Tiba for her help and Faisal for our useful discussions.

References

[1] R.M.Beam; R.F.Warming (1976) *An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form.* Journal of Computational Physics, vol.22, pp 87-110.

[2] T.Cebeci; A.M.O.Smith; G.Mosinskis (1970) *Calculation of Compressible Adiabatic Turbulent Boundary Layers.* AIAA J, vol.8, no.11, pp 1974-1982.

[3] M.C.Cline (1974) *Computation of Steady Nozzle Flow by a Time-Dependent Method.* AIAA J, vol.12, no.4, pp 419-420.

[4] M.C.Cline (1976) *Computation of Two-Dimensional, Viscous Nozzle Flow.* AIAA J, vol.14 no.3, pp 295-296.

[5] D.W.Holder; D.C.Macphail; J.S.Thompson (1953) *Modern Developpement in Fluid Dynamics High Speed Flow.* vol II, Ed L.Howarth, Oxford University Press.

[6] B.W.Imrie (1973) *Compressible Fluid Flow.* A Halsted Press Book, ed. John Wiler & Sons.

[7] S.L.Johnsson; Y.Saad; M.H.Schultz (1985) *Alternating Direction Methods on Multiprocessors.* Technical Report YALEU/DCS/RR-382, Yale University,Dept. of Computer Science.

[8] H.Lomax; T.H.Pulliam (1982) *A Fully Implicit Factored Code for Computing Three-Dimensional Flows on the ILLIAC IV* in G.Rodrigue *Parallel Computations.* Academic Press. pp 217-250.

[9] W.J.Rae (1971) *Some Numerical Results on Viscous Low-Density Nozzle Flows in the Slender-Channel Approximation.* AIAA J, vol.9, no.5, pp 811-820.

[10] F.Saied; C.T.Ho; S.L.Johnsson; M.H.Schultz (1987) *Solving Schrödinger Equation on the Intel iPSC by the Alternating Direction Method,* in *Hypercube Multiprocessors 1987,* ed. Michael Heath, SIAM.

[11] A.H.Shapiro (1953) *The Dynamics and Thermodynamics of Compressible Fluid Flow.* vol I, The Ronald Press Company, New York.

[12] J.L.Steger (1978) *Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries.* AIAA J, vol.16, no.7, pp 679-686.

[13] T.H.Pulliam;J.L.Steger (1980) *Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow.* AIAA J, vol.18, no.2, pp 159-167.

[14] J.C.Strikwerda (1982) *A Time-Split Difference Scheme for the Compressible Navier-Stokes Equations with Applications to Flows in Slotted Nozzles.* in G.Rodrigue *Parallel Computations.* Academic Press. pp 251-267.

[15] P.D.Thomas (1979) *Numerical Method for Predicting Flow Characteristics and Performance of Nonaxisymmetric Nozzles-Theory.* NASA CR-3147.

[16] J.F.Thompson; F.C.Thames; C.W.Mastin (1974) *Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Fields Containing any Number of Arbitrary Two-Dimensional Bodies.* Journal of Computational Physics, vol.15, pp 299-319.

[17] M.Vinokur (1974) *Conservation Equations of Gasdynamics in Curvilinear Coordinate Systems.* Journal of Computational Physics, vol.14, pp 105-125.