STUDIA UNIV BABEŞ-BOLYAI, MATHEMATICA, XXXVIII, 3, 1993

SERIAL AND PARALLEL ALGORITHMS FOR SOLVING A PROBLEM OF CONVECTION IN POROUS MEDIUM

Ioana CHIOREAN*

Dedicated to Professor Emil Muntean on his 60th anniversary

Received August 5, 1993
AMS subject classification 65Y05, 68Q22

REZUMAT. - Algoritmi seriali şi paraleli pentru rezolvarea unei probleme de convexitate în mediu poros. Scopul acestei lucrări este să se facă o comparaţie între algoritmii seriali şi paraleli, pentru a rezolva o problemă dată în mediu poros Sunt studiate în lucrare performanţele algoritmilor paraleli care au ca scop creşterea vitezei de calcul şi a eficienţei lor

Abstract. The main purpose of this paper is to make a comparison between a serial and a parallel algorithm for solving a given problem of convection in porous medium The performances of the parallel algorithm, established by means of speed-up and efficiency, are studied

NOMENCLATURE

g	gravitational acceleration
V	velocity of the fluid
p	pressure of fluid
T	temperature of fluid
K	permeability of the saturated porous medium
k	thermal conductivity of porous medium
S	rate of internal heat generation of porous medium
Ra	internal Rayleigh number
L	characteristic length of the porous medium
t	time

* *"Babeş-Bolyai" University, Faculty of Mathematics and Computer Science, 3400 Cluj-Napoca, Romania*

u,v	velocity components
x,y	coordinates

Greek symbols

ρ	density of fluid
μ	viscosity of fluid
$(pc)_f$	heat capacity of fluid
$(pc)_p$	heat capacity of porous medium
β	thermal expansion coefficient
ψ	dimensionless stream function
ϕ	angular coordinate

Superscripts

dimensional variables

Subscripts

0	value at reference temperature and density

1. Introduction. The problem under consideration is that of $2D$ steady laminar convection in a porous layer bounded by an inclined squre box with four rigid walls of constant temperature (fig 1) Heat is generated by a uniformily distributed energy sources within the cavity The porous layer is isotropic, homogeneous and saturated with an incompressibile fluid The heat generation creates a temperature gradient across the layer, and thereby provides a driving mechanism for natural convection within the cavity

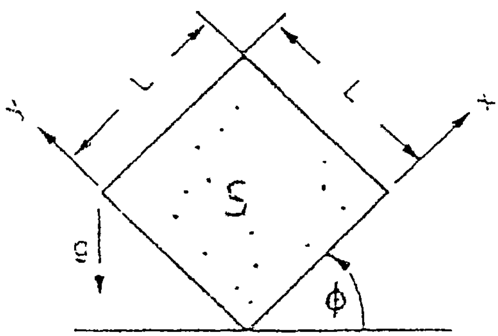

Fig 1 Schematic diagram of the enclosure

In the present study, the saturated porous medium is treated as a continuum, with the solid and fluid phases in local thermodynamic equilibrium Also, the saturated fluid and the porous matrix are supposed incompressible and all physical properties of the medium, except the fluid density are taken to be constant

2. Converning equations. The fluid motion obeys the equations Darcy-Oberbeck-Boussineq For the case of volumetric heating considered here, the governing equations can be written as

$$\nabla \cdot V' = 0, \tag{1}$$

$$V' = \frac{K}{\mu}(\rho' g - \nabla p'), \tag{2}$$

$$(\rho c)_p \frac{\partial T'}{\partial t'} + (\rho c)_f (V' \cdot \nabla) T' = k \nabla^2 T' + S', \tag{3}$$

$$\rho' = \rho_0' [l - \beta(T' - T_0')] \tag{4}$$

The four equations may be written

$$\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} = 0, \tag{1'}$$

$$u' = \frac{k}{\mu}\left(-\rho' g \sin\phi - \frac{\partial p'}{\partial x'}\right), \tag{2'}$$

$$v' = \frac{K}{\mu}\left(-\rho' g \cos\phi - \frac{\partial p'}{\partial y'}\right), \tag{2''}$$

$$(\rho c)_p \frac{\partial T'}{\partial t'} + (\rho c)_f \left(u' \frac{\partial T'}{\partial x'} + v' \frac{\partial T'}{\partial y'}\right) = k\left(\frac{\partial^2 T'}{\partial x'^2} + \frac{\partial^2 T'}{\partial y'^2}\right) + S', \tag{3'}$$

$$\rho' = \rho_0 [l - \beta(T' - T_0')] \tag{4'}$$

Derivating (2') after y' and (2") after x' and taking into account that the temperature function has the form $T'(x' y')$, it is obtained

$$\frac{\partial u'}{\partial y'} = \frac{K}{\mu}\left(g \sin\phi \, \rho_0' \beta \frac{\partial T'}{\partial y'} - \frac{\partial^2 p'}{\partial x' \partial y'}\right), \tag{5}$$

$$\frac{\partial v'}{\partial x'} = \frac{K}{\mu}\left(g\cos\phi\,\rho_0'\,\beta\,\frac{\partial T'}{\partial x'} - \frac{\partial^2 p'}{\partial x'\,\partial y'}\right) \tag{6}$$

Subtracting (6) from (5) we get

$$\frac{\partial u'}{\partial y'} - \frac{\partial v'}{\partial x'} = \frac{K}{\mu}g\,\rho_0'\beta\left(\sin\phi\,\frac{\partial T'}{\partial y'} - \cos\phi\frac{\partial T'}{\partial x'}\right) \tag{7}$$

Using the dimensionless variables

$$t = \frac{kt'}{(\rho c)_p L^2};\ u = \frac{(\rho c)_f L u'}{k},\ v = \frac{(\rho c)_f L v'}{k},\ x = \frac{x'}{L},\ y = \frac{y'}{L},\ T = \frac{k(T' - T_0')}{S' L^2}$$

(7) becomes

$$\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = \frac{K L^3 S' \rho_0' g \beta (\rho c)_f}{\mu k^2}\left(\sin\phi\frac{\partial T}{\partial y} - \cos\phi\frac{\partial T}{\partial x}\right) \tag{7'}$$

Taking $Ra = \dfrac{KL^3 S' g\beta}{\alpha\nu k}$, where $\nu = \mu/\rho_0'$ and $\alpha = (\rho c)_f/k$ as the Rayleigh number, (7') becomes

$$\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = Ra\left(\sin\phi\,\frac{\partial T}{\partial y} - \cos\phi\,\frac{\partial T}{\partial x}\right) \tag{7''}$$

Analogously, using the dimensionless variables, (1') and (3') become

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{1''}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \nabla^2 T + 1 \tag{4''}$$

Equation (4") is verified by the streamfunction ψ where

$$u = \frac{\partial\psi}{\partial y},\ v = -\frac{\partial\psi}{\partial x} \tag{8}$$

So, imtroducing (8) in (4") and (7") we get the finally system of two equations with two unknowns (the temperature function T and the stream function ψ)

$$\begin{cases} \dfrac{\partial T}{\partial t} + \dfrac{\partial\psi}{\partial y}\dfrac{\partial T}{\partial x} - \dfrac{\partial\psi}{\partial x}\dfrac{\partial T}{\partial y} = \nabla^2 T + 1, \\ \nabla^2\psi = Ra\left(\sin\phi\,\dfrac{\partial T}{\partial y} - \cos\phi\,\dfrac{\partial T}{\partial x}\right) \end{cases} \tag{9}$$

We solve this system beeig situated in an enclosure with unit square section ($L = 1$), eith the initial conditions

$$T_0 = \psi_0 = 0 \tag{10}$$

and the boundary conditions

$$T = \psi = 0 \quad \text{for } x = 0 \text{ and } 1, \; y = 0 \text{ and } 1 \tag{11}$$

Numerical results.

3.1. The Steady Problem In the steady case, our system of equations is

$$\begin{cases} \dfrac{\partial\psi}{\partial y}\dfrac{\partial T}{\partial x} - \dfrac{\partial\psi}{\partial x}\dfrac{\partial T}{\partial y} = \nabla^2 T + 1, \\ \nabla^2\psi = Ra\left(\sin\phi\,\dfrac{\partial T}{\partial y} - \cos\phi\,\dfrac{\partial T}{\partial x}\right) \end{cases} \tag{12}$$

In order to obtain the solution for the system (12) with the conditions (10) and (11), we used the Multigrid method [4] with a Gauss-Seidel smoother The space derivatives were approximated in the following manner the first order derivatives with the Euler forward formula and the second order derivatives with the centered differences, accordind to [6] The discretized solution for the temperature and stream functions was obtained working on an equidistant grid Ω_l (where l indicates the level of grid), defined in the following manner

$$\Omega_l = \left\{ (ih_l, jh_l) \mid 0 \le i, j \le N_l, h_l = 1/N_l, N_l = 2^{l+1} \right\}$$

Denoting $T_{i,j} = T(ih_l, jh_l)$, $\psi_{i,j} = \psi(ih_l, jh_l)$ for every $0 \le i, j \le N_l$, l being one grid the system becomes

$$\begin{cases} \dfrac{\psi_{i,j+1}-\psi_{i,j}}{h_l}\dfrac{T_{i+l,j}-T_{i,j}}{h_l} - \dfrac{\psi_{i+l,j}-\psi_{i,j}}{h_l}\dfrac{T_{i,j+l}-T_{i,j}}{h_l} = \dfrac{T_{i+1,j}+T_{i-1,j}+T_{i,j+1}+T_{i,j-1}-4T_{i,j}}{h_l^2}+1, \\ \dfrac{\psi_{i+1,j}+\psi_{i-1,j}+\psi_{i,j+1}+\psi_{i,j-1}-4\psi_{i,j}}{h_l^2} = Ra\left(\sin\phi\dfrac{T_{i,j+1}-T_{i,j}}{h_l} - \cos\phi\dfrac{T_{i+1,j}-T_{i,j}}{h_l}\right) \end{cases} \tag{13}$$

The solution of system (13) was obtained in two ways first, as the output of an serial algorithm and second, as the output of a parallel algorithm.

3.1.1. The serial algorithm The algorithm which solves (13) by means of up to seven grids ($l = 7$) contains the following steps

1 Solve the first equation of system (13) using (10) and (11); results T new;

2 Solve the second equation of system (13) using T new just determined, ψ_0 and ψ at the boundary; results ψ new;

3 Solve the first equation using ψ new and (11); results T new;

4 Repeats Steps 3 and 4 until "CONDITION" (When it is accomplish, the steady solution is obtained)

Note In our case, "CONDITION" means that the difference between two succesive approximation is less that 10^{-6} In other words, if we denoted, e g. F^{old} anf F^{new} two succesive approximations (where F represents T or ψ), "CONDITION" will be

$$\| F^{new} - F^{old} \| \leq 10^{-6}$$

where $\|\cdot\|$ denotes the Euclidean norm [4]. Fig 2a and b indicate the decreasing of error during ten repetitions of Steps 3 and 4 (Fig 2b detailes more the error at temperature function)

Concerning the results, our observations are the followings: the steady temperature has form like in Fig 3 and is not influenced by Ra number or ϕ angle Also, the general shape of the function (and this note is valable for the stream function, too) does not change with the numbers of grid points

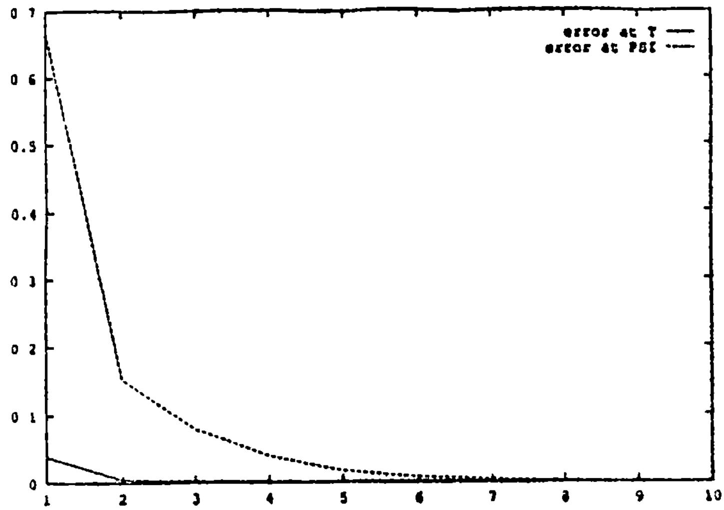

Fig 2a. The error

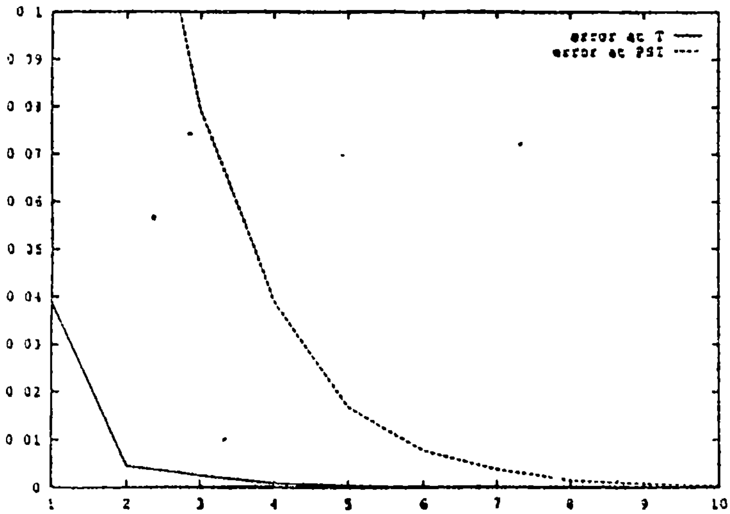

Fig.2b The error (detailed)

The stream function modifies according with the Rayleigh number and has the shape as in Fig 4

The stream function modifies also according with the angle of enclosure (see Fig 5a-c)

3.1.2. The parallel algorithm. The parallel algorithm was implemented on the INMOS Transputer System from University of Heidelberg, under PARIX operating system The main

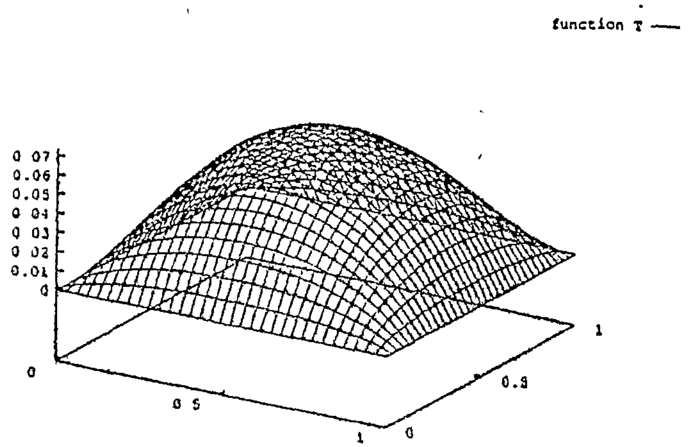

Fig 3 Temperature in steady case with Ra=500 and ϕ=0

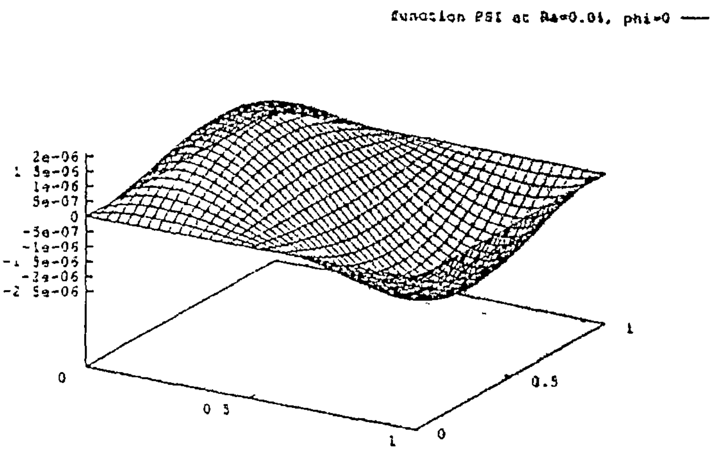

Fig.4 Stream function in steady case with Ra=0 01 and ϕ=0

ideea in solving our problem is that of [3], but with chages due to the convective terms (first equation) and the right-hand-side (second equation) from (12) We use a rectangular grid with $(N_l - 1) \times (N_l - 1)$ unknowns, then each processor is assigned to a subset of unknowns (data partitioning) In an one-dimensional arrengement of n processors caled a ring configuration of length n, processor $p, p \in \{0, \ldots, n-1\}$ is assigned to the grid points $\{(i,j) | \max(1, pN_l \backslash n) \leq i \leq (p+1)N_l \backslash n,\ 1 \leq j \leq N_l\}$ If the sidelength of the grid is not divisible by the number of

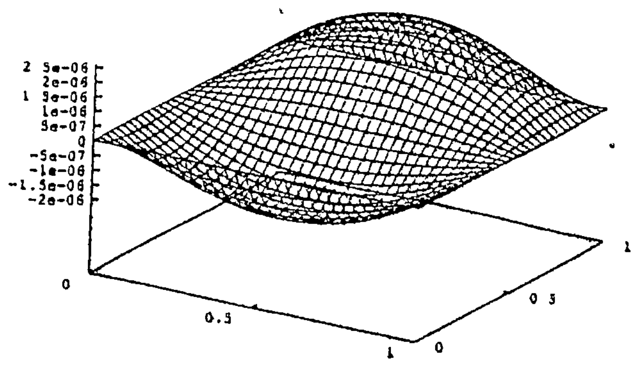

Fig 5a Stream function, with Ra=0.01 and ϕ=90

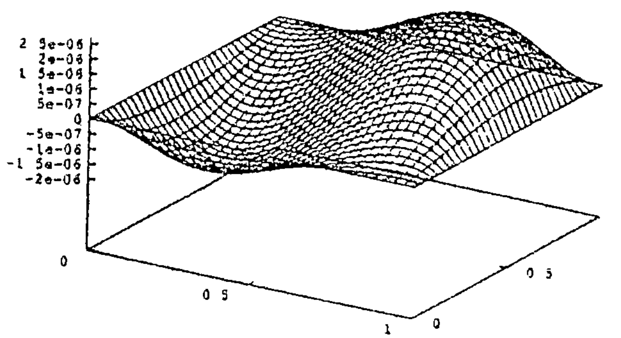

Fig 5b Stream function with Ra=0.01 and ϕ=145

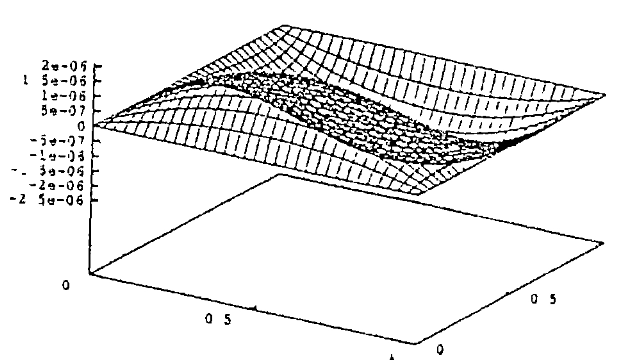

Fig 5c Stream function with Ra=0.01 and ϕ=270

processors, then some of them will be assigned more unknowns than others, generating an unequal load balance, which is one source for loss of efficiency Taking into account the way of disposing the grid points on processors and denoting by $x_{min}(p)$ and $x_{max}(p)$ the leftmost, respectively the rightmost grid point column stored by processor p, each processor will executes simultaneously the following steps

1 Computes the convective terms for the first equation of (12),

2 In case of an overlapping, sends values to the leftside processor (if it exists) and receives values from the righside processor (if it exists),

3 For every j from 1 to n_j do

3 1 Receives values from the leftside processor (if it exists),

3 2 For every i from $x_{min}(p)$ to $x_{max}(p)$ do

Computes Gauss-Seidel iterations,

3 3 Sends values to the rightside processor (if it exists)

After processing the previous steps, with step 3 repetead till the steady solution for temperature is obtained (we have noticed that it happened after 10 iterations), we proceed analogously to solve the second ecuation of (12).

In order to compare the results obtained with the serial and the parallel code, we used, like in [1] and [3], *the speed-up*, defined as

$$S(n) = \frac{T_{Mono}}{T_{Multi}(n)} \tag{14}$$

where T_{Mono} is the time needed for obtaining the solution the with the serial code and T_{Multi} is the time took by the parallel code, using n processors, and *the efficiency* which is defined

$$E(n) = \frac{S(n)}{n} \tag{15}$$

Table 1 presents the execution times (in sec) for the serial and the parallel code, when a different number of processors was used So, we can notice that the increasing of time for the serial code is deeply connected with the numbers of grid points (on a coarse grid, the execution takes a few seconds, the execution takes a few seconds, on a fine grid it takes more that an hour!) and the execution time decreases according with the number of processors used, with the observation that for the coarse grid 32 x 32 the situation is like in Fig 7

Table 1 Execution time

Nr proc/Nr pc	32×32	64×64	128×128	256×256	321×321
1	16 6029	67 9188	276 725	1116 65	1677 73
7	7 86234	19 2012	53 5307	167 708	240 032
11	7 472	17 1044	43 5171	126 039	173 89
15	7 59328	16.6198	39 6177	107 014	141 814
19	7 40141	16 0309	36 4428	95.6118	128 211
23	7 49709	15 6065	35 5842	89 002	116 689

Fig 6 visualises the information from Tabel 1, meanwhile Fig 7 indicates only an unconcludent situation when more that one processors are used

Table 2 Speed-up

Nr proc/Nr pc	32×32	64×64	128×128	256×256	312×312
7	2 11	3 53	5 16	6 65	6 98
11	2 21	3 97	6 35	8 85	9 5
15	2 18	4 08	6 95	10 43	11 8
19	2 24	4 23	7 59	11 67	13 08
23	2 21	4 35	7 76	12 54	14 3

The speed-up for all operations carried out on a fixed grid depends heavily on the number of unknowns per processor, because a larger proportion of computing time is spent on communication and the effects of unequal load distribution are more pronounced if the number of grid points per processor is small This means that a high speed-up can be

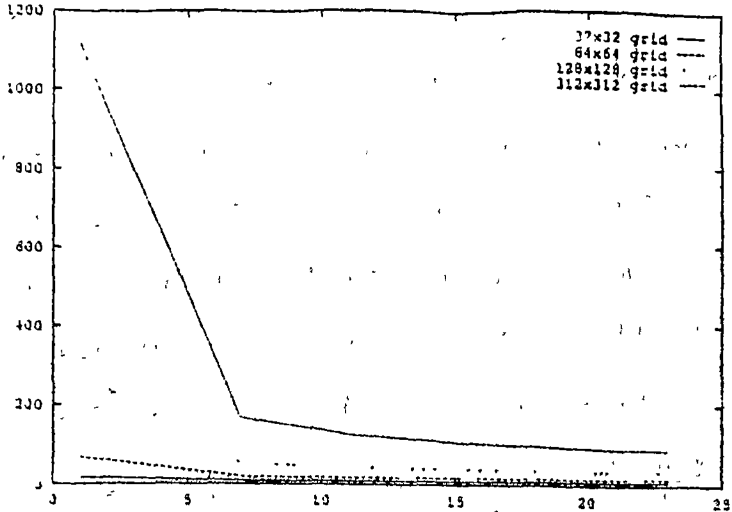

Fig 6 Execution time

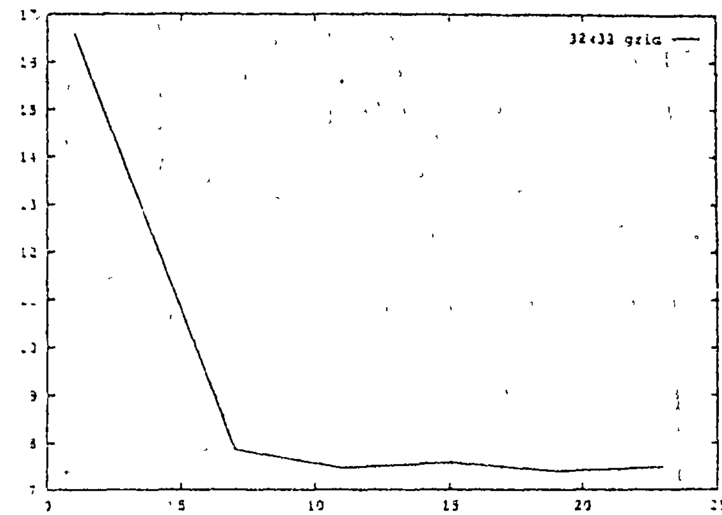

Fig 7 Execution time working with several processors on a coarse grid

achieved on the fine grids (assuming a large number of grid points per processor on the fine grids) like in Fig.9 whereas the speed-up deteriorates on the coarser grids (see Fig 10) Table 2 contains the values which sustained these observations and on which Fig 8 and 9 are based

Working with several processors on a coarse grid, the improving of speed-up is not concludent, as we can see from Fig 10 Next, accordind with (15), Table 3 contains the values

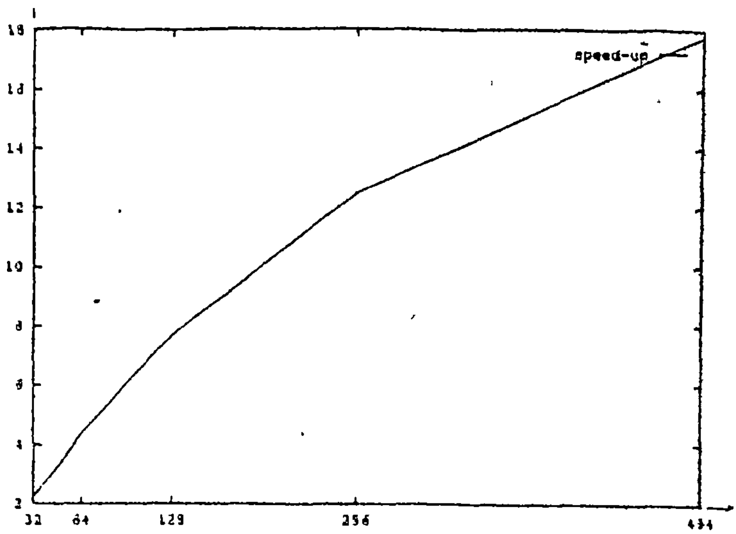

Fig 8 Speed-up (from coarser to finer grids with p=23)

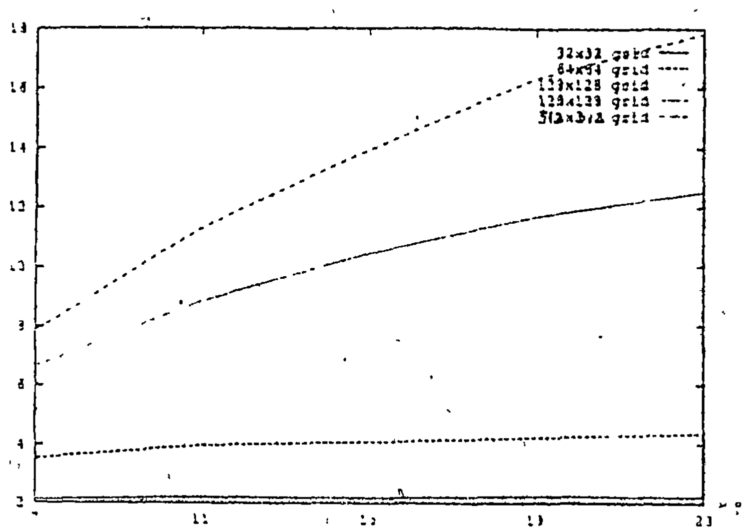

Fig 9 Speed-up

which indicate how efficiency depends on the number of processors and on the number of grids points

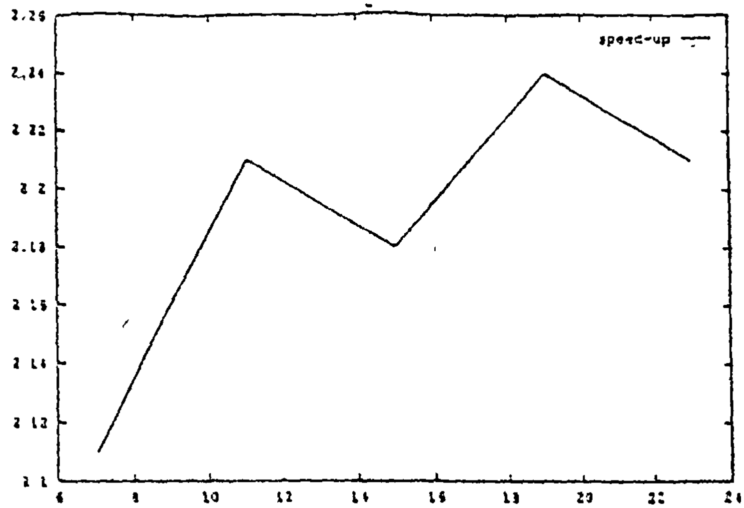

Fig 10 Speed-up on a coarse grid, with several processors

Table 3 Efficiency evolution

Nr proc/Nr pc	64×64	128×128	256×256	312×312
7	0 50	0 73	0 95	0 99
11	0 36	0 57	0 80	0 86
15	0 27	0 46	0 69	0 78
19	0 22	0 34	0 61	0 68
23	0 18	0 33	0 54	0 62

Based on Table 3 , Fig 11 shows the increasing of efficiency when finer grids are used

3.2.The Unsteady Problem Solving the unsteady problem means to solve the system in the original form (9) In order to do this, we use the same finite difference formulas to discretize the space derivatives, as in 3 1 The time deribative will be discretized with the backward Euler formula ([6]) We denote by dt the timestep, which is considered fix, by L^{ell} the Laplace operator and by G_x and G_y the gradient operators ([2]) Let T^k be the temperature function at the moment of time $t_k = kdt$ Then the first equation of system (9) can be written

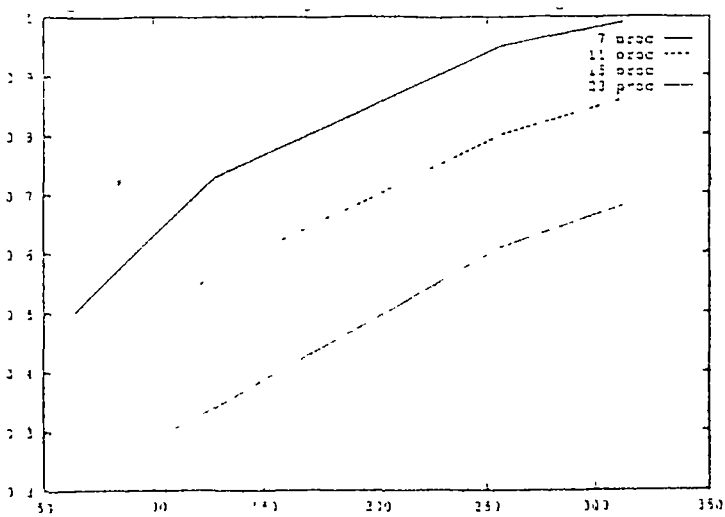

Fig 11 Efficiency

in the following manner

$$\frac{T^{k+1}-T^{k}}{dt}+G_{y}\psi^{k+1}G_{x}T^{k+1}-G_{x}\psi^{k+1}G_{y}T^{k+1}=L^{ell}T^{k+1}+1 \qquad (16)$$

For a fixed time interval $[t_k, t_{k+m}]$, denoting with I the Identity operator and based on (16), to solve the parabolic equation of system (9) means to solve the following bidiagonal blok-structured system

$$As = b$$

where

$$A=\begin{bmatrix} \frac{1}{dt}I+G_{y}\psi G_{x}\cdot+G_{x}\psi G_{y}\cdot-L^{ell}\cdot & & 0 \\ \frac{-1}{dt}I & \frac{1}{dt}I+G_{y}\psi G_{x}\cdot+G_{x}\psi G_{y}\cdot-L^{ell}\cdot & \\ 0 & \frac{-1}{dt}I & \frac{1}{dt}I+G_{y}\psi G_{x}\cdot+G_{x}\psi G_{y}\cdot-L^{ell}\cdot \end{bmatrix}$$

$$s = \begin{bmatrix} T^{k+1} \\ T^{k+2} \\ \\ T^{k+m} \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 1 + \frac{1}{dt} IT^k \\ 1 \\ \\ 1 \end{bmatrix}$$

We observe that at every moment of time the relation which gives the temperature function is fully implicit and we have to solve, as the first equation of system (9), the following

$$\frac{T_{i,j}^{k+1}}{dt} + \frac{\psi_{i,j+1}^{k+1} - \psi_{i,j}^{k+1}}{h_i} \frac{T_{i+1,j}^{k+1} - T_{i,j}^{k+1}}{h_i} - \frac{\psi_{i+1,j}^{k+1} - \psi_{i,j}^{k+1}}{h_i} \frac{T_{i,j+1}^{k+1} - T_{i,j}^{k+1}}{h_i} -$$

$$- \frac{T_{i+1,j}^{k+1} + T_{i-1,j}^{k+1} + T_{i,j+1}^{k+1} + T_{i,j-1}^{k+1} - 4T_{i,j}^{k+1}}{h_i^2} = 1 + \frac{T_{i,j}^k}{dt} \tag{17}$$

Equation (17) together with the second equation of system (13) will form the problem we have to solve in this case As in the paragraph 3 1 , the Multigrid method was used and the general scheme of solving is the following

Step 1 Solve equation (17) at the moment of time t^{k+1} based on T^k (where T^0, the initial temperature is given), results T^{k+1}

Step 2 Solve the second equation of system 9130 at the moment of time t^{k+1} based on T^{k+1} just determined results ψ^{k+1}

Step 3 Repeat Steps 1 and 2 until "CONDITION 1"

Note "CONDITION 1" indicates the number of time steps we have to execute until the steady solution is obtained, normally, thi depends on the value of dt For instance, if dt = 0 1, the steady solution is attain in mostly 10 steps, but for dt = 0 001 we need almost 180 time iterations to get it Fig 12a-2 show the evolution in time of the temperature function, for Ra = 500 , ϕ = 0 and dt = 0 001 After 180 time steps, the temperature is stationary (in order to compare, see Fig 3)

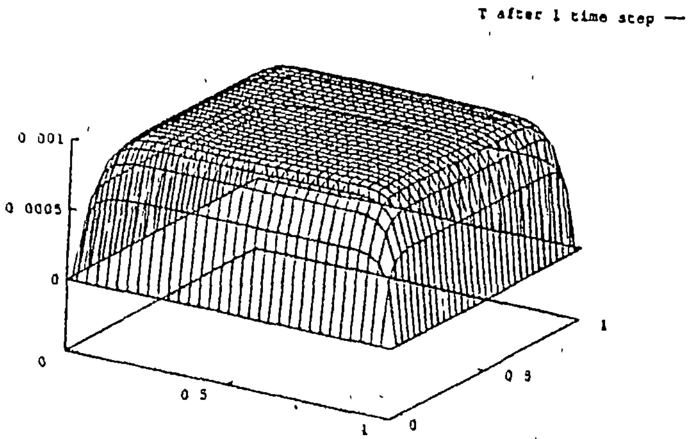

Fig 12a The temperature after 1 time step

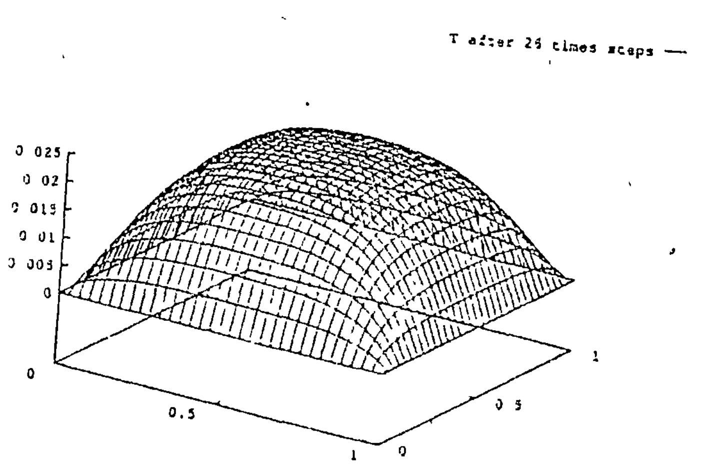

Fig 12b The temperature after 26 time steps

The following graphics show how the temperature function evoluates up to the steady case

In the same conditions (but for Ra = 125), Fig 13a-c present the evolution in time of the stream function

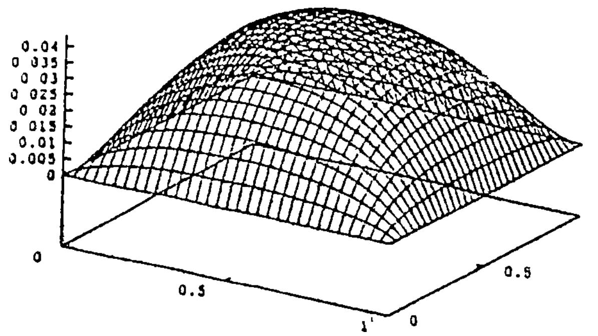

Fig 12c The Temperature after 51 time steps

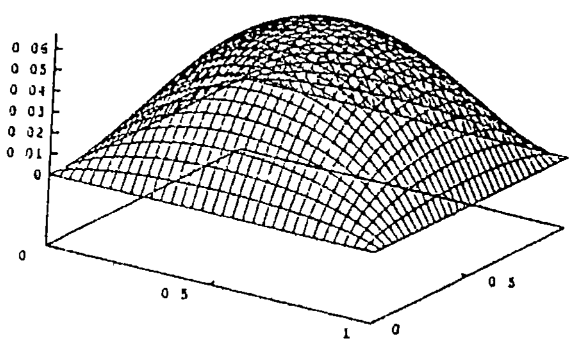

Fig 12d The Temperature after 131 time steps

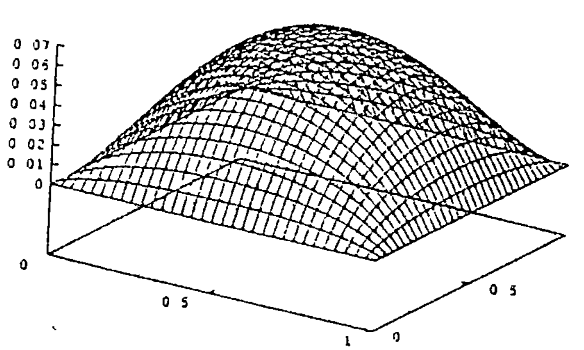

Fig 12e The Temperature after 180 time steps

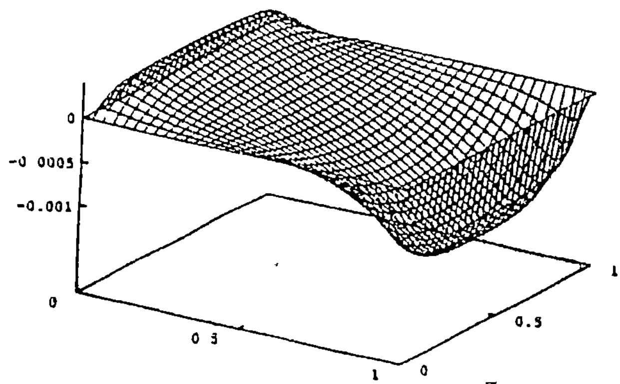

Fig 13a The Stream Function after 1 time step

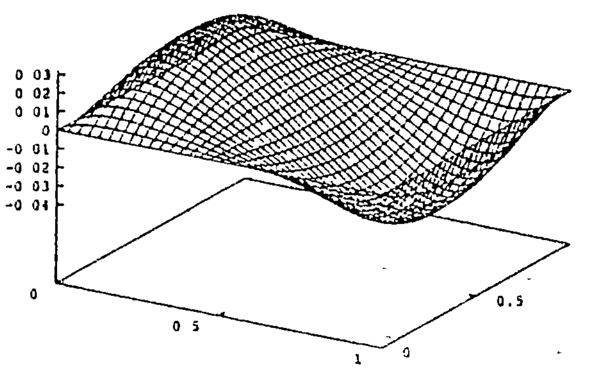

Fig 13b The Stream Function after 26 time steps

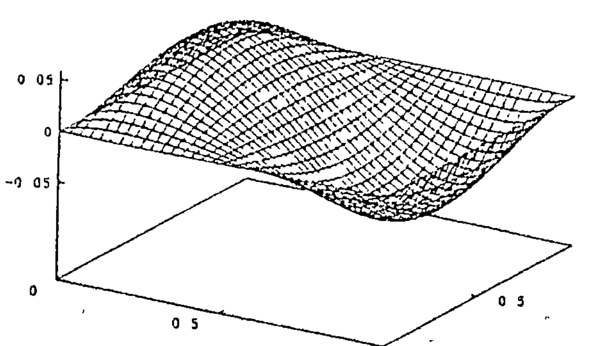

Fig 13c The Stream Function after 51 time steps

After 180 time steps, the stream function becomes steady (Fig 4), as we can see from the following graphics

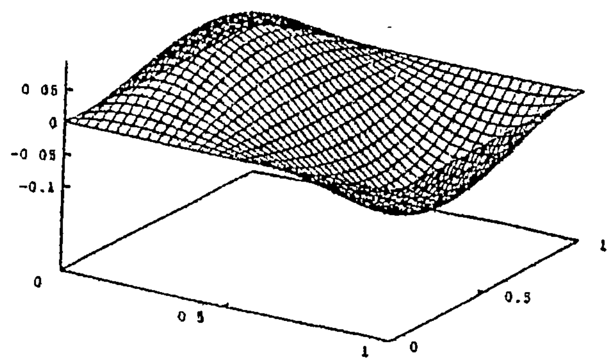

Fig 13d The Stream Function after 131 time steps

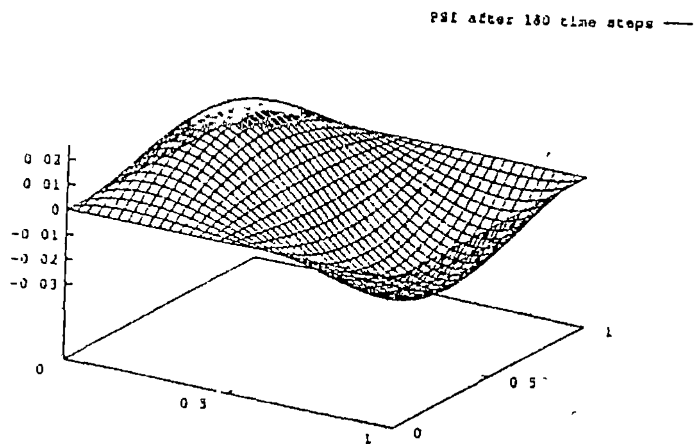

Fig 13e The Stream Function after 180 time steps

4.Conclusions. The main goal of this research was to show that transputer system can efficiently solve large computational problems with good performance We made study on a problem of interest in the computational fluid dynamics field, which generated a parabolic problem expressed by a PDE system In order to verify the results, we solve first, in serial and in parallel, the steady problem The outputs of this two different codes were almost the sama Based on the steady solution, we solved then the original problem, indicating by means of many graphics the evolution in time, up to the steady state, of the solution functions

Acknowledgements. The author would like to especially thank to Prof Willi Jäger for his constant support Also, expresses her thanks to Peter Bastian and all the coleagues from IWR, University of Heidelberg, for their help in elaborating this paper

REFERENCES

1 Bader,G , Gehrke E , *On the performance of transputer networks for solving linear systems of equations*, Parallel Computing 17(1991), pp 1397-1407

2 Bastian,P , Burmeister,J , Horton,G , *Implementation of a parallel multigrid method for parabolic partial differential equations*, Proceeding of the Sixth GAMM-Seminar, Kiel, Jan 19-21, 1990, pp 18-27

3 Bastian,P , Horton,G , *Parallelization of Robust Multigrid Method ILU Factorization and Frequency Decomposition Method*, SIAM J SCI STAT COMP , vol 12, No 6, pp 1457-1470, nov 1991

4 Hackbusch,W , *Multi-Grid Methods and Applications*, Springer Verlag, Berlin, Heidelberg, 1985

5 May,H O , *A numerical study on natural convection in an inclined square enclosure containing internal heat sources*, Int J Heat Mass Transfer, Vol 34, No 4, pp 919-928/1991

6 Roache,J ,P , *Computational Fluid Dynamics*, Hermosa, Albuquerque, New-Mexico, 1985

7 Vasseur,P , Hung Nguyen,T , Robillard,L , Tong Thi,V K , *Natural convection between horizontal concentric cylinders filled with a porous layer with internal heat generation*, Int J Heat Mass Transfer, vol 27, no 3, pp 337-349/1984