PC Cluster Implementation of a Mass Transport Two-Dimensional Model

Ricardo Vargas Dorneles[1,2], Rogério Luis Rizzi[1,3], Cesar Albenes Zeferino[1], Tiaraju A. Diverio[1], Philippe O. A. Navaux[1], Altamiro A. Susin[1], Sergio Bampi[1]

[1] Programa de Pós-Graduação em Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, CP 15064, 91501-970, Porto Alegre, RS, Brasil.

[2] Departamento de Informática, Centro de Ciências Exatas e Tecnologia, Universidade de Caxias do Sul, Rua Francisco Getúlio Vargas, 1130, 95001-970, Caxias do Sul, RS, Brasil

[3] Departamento de Matemática e Estatística, Universidade Estadual do Oeste do Paraná, Campus de Cascavel, Rua Universitária, 2069, 85801-110, Cascavel, PR, Brasil.

{cadinho, rizzi, zeferino, diverio, navaux, susin, bampi} @inf.ufrgs.br

Abstract
This work presents a parallel solution for the continuity and horizontal momentum equations, and for the mass transport equation. They constitute a model for contaminant transport simulation. The first ones were solved applying Krylov-Schwarz method and for the last one it was used a pipelined Thomas method. The computational model was implemented on a PC cluster using MPI message passing library. Semi-implicit numerical schemes were developed over a staggered grid.

Keywords— **Cluster computing, Krylov-Schwarz method, pipelined Thomas algorithm, Mass Transport Equation, water quality.**

I. Introduction

Shallow Water Equations (briefly, SWEs) and 2-D Mass Transport Equation (briefly, MTE) constitute, under some conditions, a model to describe the hydrodynamics and dispersion behaviors in both water bodies and atmosphere. Although they are a simplification of real models, important issues of estuaries, rivers and coastal seas can be well represented. Also, they are a 2-D version of 3-D models that govern the hydrodynamics and contaminants transport in a real fluid. For these reasons, they are frequently used as benchmarks to test new numerical and computational methods, and parallelization strategies.

The correct treatment of boundary conditions is crucial for both mathematical and numerical issues of SWEs. In [AGO 94], Agoshkov presents some boundary terms based on a consistent mathematical analysis. SWEs and MTE approaches emphasizing the domain decomposition and the application of Krylov-Schwarz solvers can be found in [CHE 98], [GOO 93] and [SIL 97]. Some computational models include Casalas' sequential version [CAS 96] and the parallel implementations done by Vollebregt [VOL 97] and Kaplan [KAP 98].

The computational simulation of water bodies has several practical applications, such as in evaluations of the conditions for bath, navigation and water supplying. With the development and implementation of a high-resolution computational model, it is possible to perform a detailed simulation of hydrodynamics and mass transport behaviors, aiming to aid the definition of actions such as the choice of outflow points for the sewerage system and the correspondent impact in the environment.

The Group of Computational Mathematics and the Group of Parallel and Distributed Processing of UFRGS have been working with the study and parallel implementation of numerical methods applied to simulation of real problems on distributed memory machines.

Based on these studies, some parallel computation models were implemented on a PC cluster to simulate the circulation in Guaíba River and the mass transport in some specific sites. Guaíba River bathes the metropolitan region of Porto Alegre City, at the southern of Brazil. With 470 km^2 of surface, it receives the outflow of Jacuí delta, which is formed by the confluence of Jacuí, Caí, Sinos and Gravataí rivers, and flows into Lagoa dos Patos. It is about 50 km long and 15 km wide in some sections, and is sited between the 50° and 55° West parallels and 28° and 35° South latitudes [CAS 85].

Guaíba River is quite important for water supplying, fluvial transport and soil irrigation for the cities on its region. However, it receives a lot of industrial and domestic contaminants.

The major goal of this work is to present a water quality simulation parallel model applied to well-mixed water bodies. The parallel implementation and application of a mathematical model for the vertically integrated two-dimensional mass transport are presented. Levels and velocities for every time step are obtained from a two-dimensional hydrodynamics model.

This paper is organized as follows. Section II presents the mathematical model for the mass transport. In Section III, the discretization applied to approximate the mathematical models for hydrodynamics and mass transport is presented. Section IV describes the domain decomposition method used in this work, while Section V details the numerical grid partitioning. Sections VI and VII present the numerical algorithms applied to solve, respectively, hydrodynamics and mass transport equations. Finally, Sections VIII and IX show the results and conclusions.

II. MASS TRANSPORT EQUATION

The mathematical model applied for hydrodynamics was based on SWEs, which are obtained from the vertical integration of the water velocity, resulting in a 2-D model with averaged horizontal velocities. SWEs are a set of nonlinear hyperbolic Partial Differential Equations (briefly, PDEs) for an incompressible fluid with a free surface, including horizontal momentum and continuity equations.

The governing equation for the advection-diffusion of constituent concentration for the case of well-mixed water bodies is MTE, which can be described as [LEE 71].

$$\frac{\partial(CH)}{\partial t}+\frac{\partial(CHU)}{\partial x}+\frac{\partial(CHV)}{\partial y}-\frac{\partial}{\partial x}\left(HD_x\frac{\partial C}{\partial x}\right) -\frac{\partial}{\partial y}\left(HD_y\frac{\partial C}{\partial y}\right)-KHC-FH=0 \qquad (1)$$

where:

- C is the concentration integrated on the vertical direction;
- $H = h + h$ is the distance from bottom to water surface, where $h = h(x,y,t)$ is the elevation of the water above the mean level and h represents the depth bellow this level;
- $U = U(x,y,t)$ and $V = V(x,y,t)$ are the velocity components on the (X,Y) coordinate system;
- $D_x = a_L \cos^2 q + a_T \sin^2 q$ and $D_y = a_L \sin^2 q + a_T \cos^2 q$ are the tensor dispersion components on X and Y directions, respectively, where a_L and a_T are the longitudinal and transverse dispersion coefficients, respectively, and q is the angle formed between X axis and vector velocity;
- F is the source term;
- $K = ln\ 0.1/T_{90}$, where T_{90} is the time for the removal of 90 percent of organisms, i.e., the decay constant.

If the changes in constituent concentration yielded by the solution of the transport equation cause negligible variation in water density, the SWEs and MTE can be solved independently. If the movement of mass predicted by the MTE causes significant change in water density, the equations must be solved as a coupled set [ZHE 95]. For coliform bacteria, as is the situation in this work, the equations are solved decoupled, and in this case, the SWEs give the level and velocities, input data for the mass transport equation. A description of the discretization model and SWEs implementation is done in [RIZ 00].

One can observe that equation (1) is a particular case for only one constituent concentration, which is coliform bacteria in this work. For the general case, there is a reaction matrix [K]. These and other details can be found in Leenderste, [LEE 70] and [LEE 71].

Initial conditions and boundary conditions are an integral part of the mathematical model. The initial condition can be written as $C(x,y,0) = C_0(x,y)$, where $C_0(x,y)$ indicates a known concentration distribution in the domain. There are three types of boundary conditions in transport models: concentrations are specified along a boundary, called Dirichlet condition; concentration gradients are specified across a boundary, called Neumann condition; and both concentrations along a boundary and concentration gradients across that boundary are specified, called Cauchy condition [ZHE 95].

The boundary conditions here specified allow the use of open and closed boundaries with Dirichlet and Neumann type boundary conditions. For SWEs, it was imposed null velocity and zero level on the closed boundaries, and defined a constant velocity on the open boundaries. For MTE, it was imposed that both concentration and gradient concentration are null on the closed boundaries. On the open boundaries, it was defined a constant concentration in boundary inflow and a specified gradient in boundary outflow.

III. DISCRETIZATION

The governing equations (hydrodynamics and mass transport) are discretized by centered finite difference. The numerical scheme was defined over a space-staggered grid Arakawa class C. The decoupling of the variables that do not need to be defined on the same point of the grid results in a reduction of the processing time while the truncation error remains the same. Furthermore, this grid conserves the physical properties [MES 98].

For the discretization of SWEs and MTE, and building of the numerical scheme, a semi-implicit approach was used. Applying this approach and Casulli's strategy [CAS 90], the algebraic linear equations systems (ALES) resulted from the discretization process have structured (5-diagonal), sparse and symmetric positive defined (SPD) matrices. These matrices can be efficiently solved using Krylov subspace iterative methods, such as conjugate gradient (briefly, CG).

For MTE, the time step splitting strategy was applied and its equation was discretized using Alternating Direction Implicit (ADI) method with two time steps, resulting in two sets of ALES, one for each direction. These ALES have non-symmetrical 3-diagonal matrices, which are efficiently solved applying Thomas method.

Alternatively, one could do this discretization to obtain non-symmetrical 5-diagonal matrices, which can be solved by GMRES iterative method.

IV. DOMAIN DECOMPOSITION

The method of domain decomposition with overlapping is called Overlapping Schwarz Method (OSM). This method is characterized by the decomposition of the original domain W into subdomains W_i, where $W \subseteq \bigcup_{i=1}^{p} W_i$.

For ALES written in the matricial form $Ah = f$, the solutions are obtained solving a sequence of subproblems $A_i h_i = f_i$ defined in W_i. The partial solutions h_i form the global solution h.

The major motivation to apply domain decomposition method is the optimal parallelism obtained, because the subproblems can be solved concurrently and the communications are based only on the information exchange on the subdomains overlapping. The choice of how these exchanges occur on these artificial boundaries defines the OSM type: additive or multiplicative.

In the additive version, applied in this work, all the subdomains use the solutions of the last iteration as boundary conditions, so that each subdomain can be solved independently. In the multiplicative version, each subdomain uses the solution of other already evaluated subdomains as boundary condition. Because this last procedure is inherently sequential, to get parallelism, it is necessary to apply schemes such as multicolor, where independent subproblems can be introduced and the number of sequential phases can be minimized [CHA 94] [SMI 96].

A. Schwarz Additive Method

To build the subproblems and the overlapping regions, the formal procedure consists in considering a matrix R_i for each subregion W_i, which, when applied to the points of a grid, returns only the values associated with the points of this subgrid. This matrix is called extension map. In the same way, it is considered an extension operator R_i^T, which extends by zero a vector of nodal values in W_i [CHA 94].

The values of a generic submatrix A_i are obtained considering $A_i = R_i A R_i^T$, where A is the global matrix. One can observe that in computational implementations, the matrices R_i and R_i^T are never formed explicitly. They are introduced to express the several types of algorithms in a similar and concise manner [SMI 96].

Thus, the parallelization approach applied in this work was based on decomposing the domain in p subdomains W_i, and consequently, in p overlapped submatrices A_i. In the generation of the overlapping, where the grid points correspond to the subdomain boundary cells, there is a data dependency on the boundary of the adjacent subdomains.

This requires the exchange of some grid points among the processors in each time step. To do this, it was considered a 2-row overlapping, as shown in figure 1. Such choice is based on the discretization process applied to the SWEs.

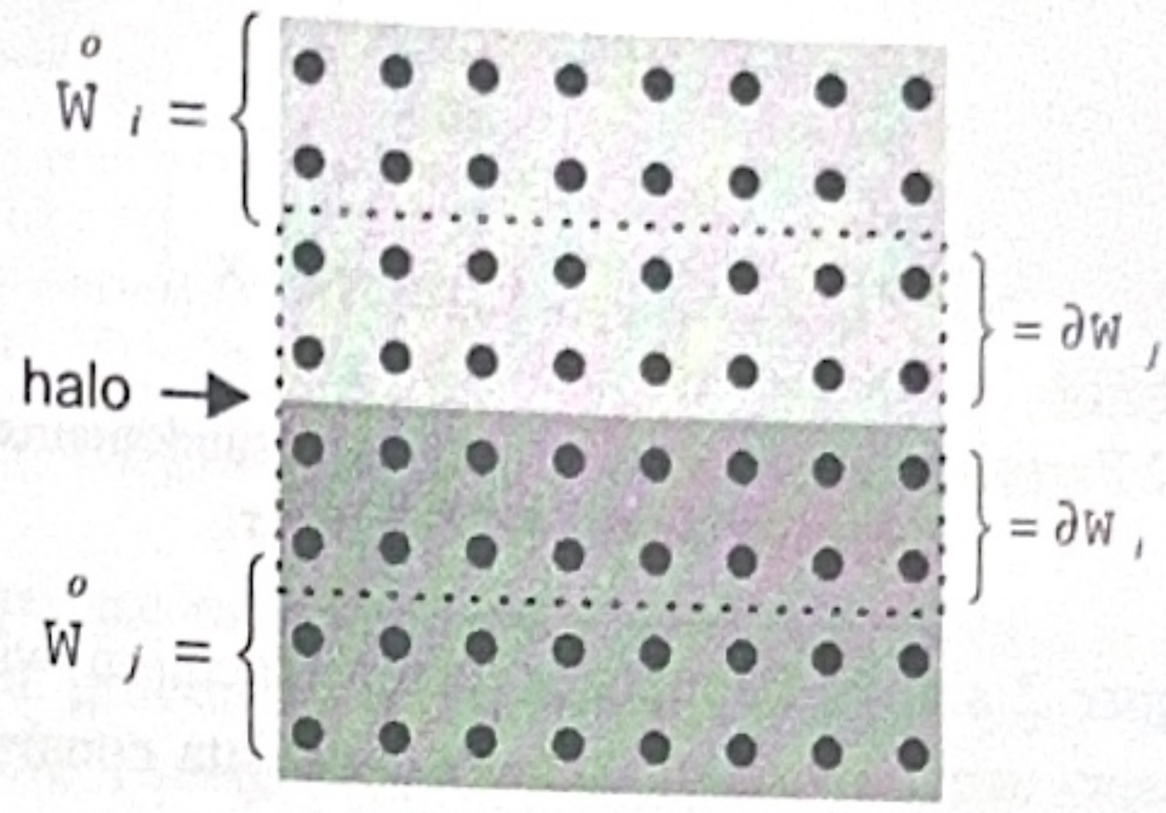


Fig. 1 Artificial boundary introduced by domain decomposition.

In figure 1, ∂W_k and $\overset{o}{W}_k$ are, respectively, the boundary and the interior of a specific subdomain.

V. NUMERICAL GRID PARTITIONING

Partitioning the numerical grids is a NP-complete problem. Therefore, the heuristic algorithms are the only alternative for this class of problems [NAS 96].

A good partitioning strategy must consider the load balancing among the processors and the minimization of the communication requirements. It must also treat complex geometry and arbitrary discretizations [DIE 97]. Recent works aim to develop strategies and parallel algorithms for dynamic balancing of grids [NAS 96]. Furthermore, partitioning can also influence the efficiency of OSM, modifying the numerical convergence rate [CIA 94].

In this work, only domain static balancing was implemented. In the hydrodynamics case, the algorithm splits the domain in stripes considering a uniform load distribution among the subdomains. Thus, in the SWEs case, it was considered a criterion based only on the load balancing, because communication occurs only on the boundaries.

For MTE, ADI method and Thomas algorithm are applied, finding the numerical solution by doing sweeps on rows and columns. For sake of simplicity, it was considered that no row is split among two or more processors, even at the cost of a small load unbalancing.

Since no other partitioning algorithms were not implemented and the applied grid refinement procedure was the heuristic above described, the global domain was decomposed in stripes on the Y direction, as in [CHE 98].

This approach generated subdomains for Guaíba River as shown in figure 2.

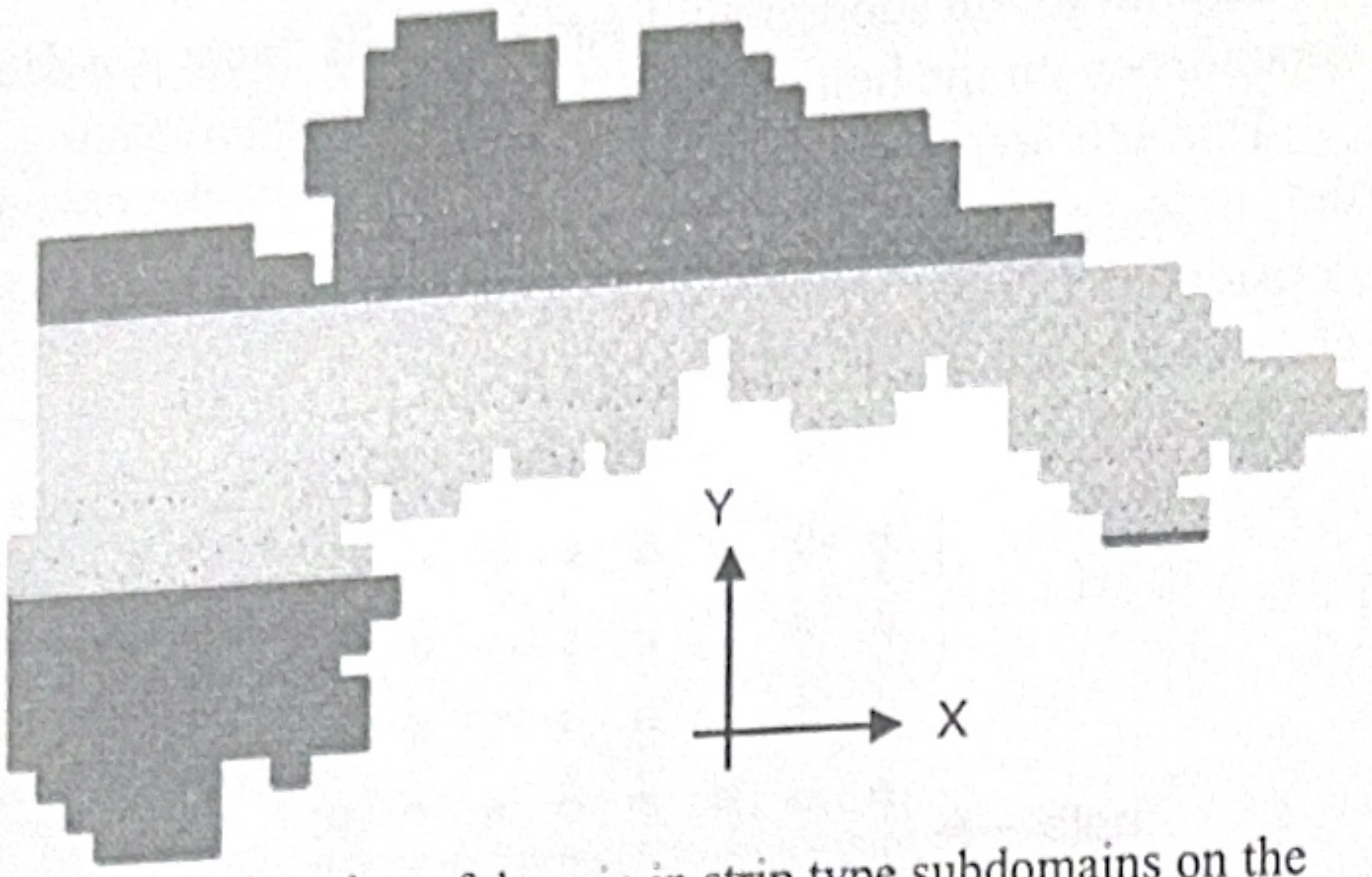


Fig. 2 Partitioning of domain in strip type subdomains on the *Y* direction for Guaíba River.

Figure 2 shows the results of an execution with three processors using a 2×2 refinement. For data consistency, a 2-row overlapping is used at boundaries.

VI. KRYLOV-SCHWARZ FOR HYDRODYNAMICS MODEL

The ALES for SWEs can be written in the form $A\text{h} = f$, where A is the coefficient matrix which is 5-diagonal and SPD, h is the level vector and f is the independent terms vector. To speed up the convergence of Krylov subspace methods, it is necessary to preconditionate the matrices of the ALES. The preconditioner used in the work was built applying the Schwarz method, which consists in using an approximation for the local inverse. This approach results in the Krylov-Schwarz methods.

A. *Conjugate Gradient Algorithm*

Since the coefficient matrix is SPD, square and sparse, the CG algorithm can be applied. The version used in this work can be found in [SHE 94] and is shown in figure 3.

```
i        0
r        f - Ah
d        M^1 r
d_new    r^T d
d_0      d_new
while i < i_max and d_new > e^2 d_0 do
    q        Ad
    a        d_new / d^T q
    h        h + ad
    If i is divisible by 50
             r        f - Ah
    else
             r        r - aq
    s        M^1 r
    d_old    d_new
    d_new    r^T s
    b        d_new / d_old
    d        r + bd
    i        i + 1
```

Fig. 3 Preconditioned CG algorithm.

CG algorithm is an iterative method applied to the solution of large sparse linear systems. It is considered a non-stationary method, because each iteration inherits the information of the previous ones. This algorithm builds the solution through elementary operations, such as sum, subtraction and scalar product of vectors, multiplication of scalar and matrix by vector.

In the OSM, the matrix is locally solved using preconditioned CG (briefly, PCG). The operations requiring more processing are two matrix-vector products.

B. *Preconditioning by Neumann Polynomial*

The convergence of an iterative method applied to a matrix J depends strongly on its conditioning. Also, it is possible to prove, as in Varga [VAR 99], that if $r(J) < 1$, where r is the spectral radius of J, the inverse of $(I - J)$, can be expressed as a power series in J. In other words: $(I-J)^{-1} = \sum_{k=0}^{\infty} J^k$.

From these results, it is possible to speedup the convergence of the CG method, improving the condition number of the coefficient matrix. One possibility for doing this is to preconditionate A, improving its condition number, which can be done by several ways. One of them is to get an approximation for the inverse of A to find M^1, with a reasonable computational cost, such that $M^1 \approx A^{-1}$, because in this case M^1A has a better condition number than A. It can be done by a polynomial approximation as follow:

1. Decompose A as $A = D + L + U$, where D, L and U are, respectively, the main diagonal, the lower triangular part and the upper triangular part;
2. Write J as $J = -D^{-1}(L+U)$;
3. Do $A = D(I-J)$.

Therefore, the inverse of A can be expressed by a power series such as:

$$A^{-1} = \sum_{k=0}^{\infty} (-1)^k \left[D^{-1}(L+U)\right]^k D^{-1} \tag{2}$$

For the domain decomposition strategy, this approach allows to get preconditioners for local matrices A_i in order to obtain a global preconditioner given by (3):

$$M^{-1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i \tag{3}$$

For $k = 0$, the Jacobi preconditioner (D^{-1}) is obtained. For $k = 1$ it is obtained a polynomial that maintains the sparseness of the original matrix A, avoiding fill-in. The preconditioner built with this approach provides a convenient approximation for the inverse of A_i. For $k > 1$, fill-in begins to occur, destroying the original structure of A.

VII. THOMAS ALGORITHM FOR MASS TRANSPORT EQUATION

Since ADI method divides the time step in two halves and, for MTE, the domain was split in stripes in the

Y direction, the following procedure was used. For the first half time step, each processor applies Thomas algorithm to solve its rows doing a row-by-row sweep. In the second half time step, a column-by-column sweep is done. Since Thomas is a sequential algorithm and each column can be divided between several processors, a direct application of Thomas would produce a high idleness because each processor depends on the results of its neighbour. To avoid this trouble, in the second half time step, a pipelined version of Thomas algorithm is applied [POV 98], as will be soon described. In both cases, the ALES are solved in two stages: forward and backward.

The solution (1) in the first half time step has a high parallelism because each processor can solve its systems independently of the other processors. The only data dependency occurs on the artificial boundaries, where a 2-row overlapping is applied (figure 1) due to the 5-point stencil used in the approximation of MTE.

In the second half time step, the solution of one matrix in a processor P_j depends on the results of P_{j-1}, during the forward stage, and of P_{j+1}, during the backward stage. These dependencies require more communication than in the first half time step and cause idleness in the processors.

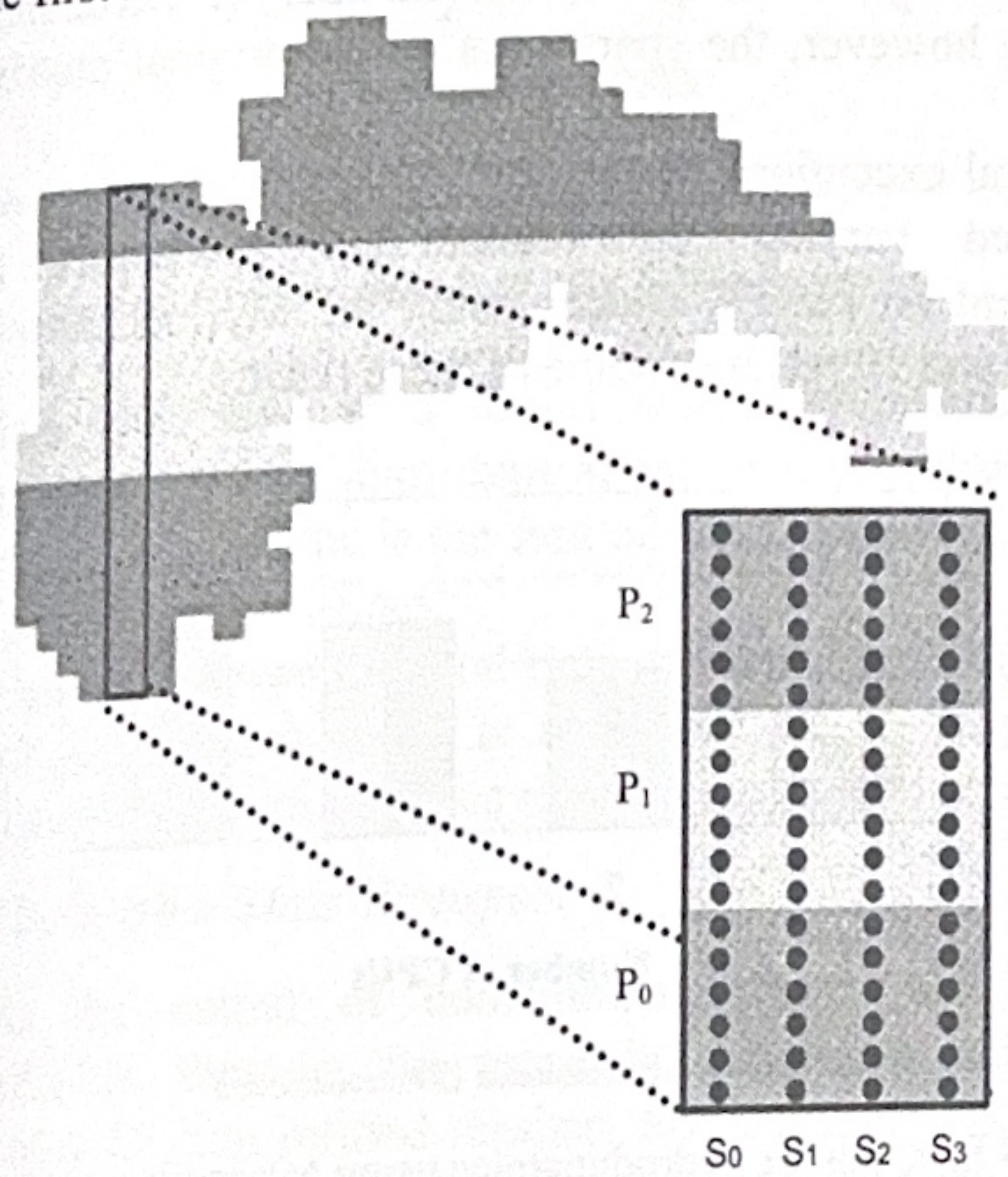


Fig. 4 A regular subdomain with four columns. Each processor solves its parts of the columns in a pipelined fashion.

For instance, figure 4 shows a regular subdomain where, for each one of the four columns, an ALES S_i must be solved. However, as each system is distributed among the three processors (P_0, P_1 and P_2), a direct application of the Thomas algorithm, as described bellow, is very inefficient.

Figure 5 presents part of the scheduling for the execution of Thomas algorithm for the domain shown in figure 4. In this figure, F_i and B_i represent, respectively, the forward and backward stages being performed for a system S_i. Firstly, system S_1 is entirely solved and only after that the resolution of system S_2 is started. As can be viewed, 12 periods are necessary to solve the first two systems, and each processor stays 8 periods in an idle state. For the domain, 24 periods are necessary to solve the four ALES during this second half time step.

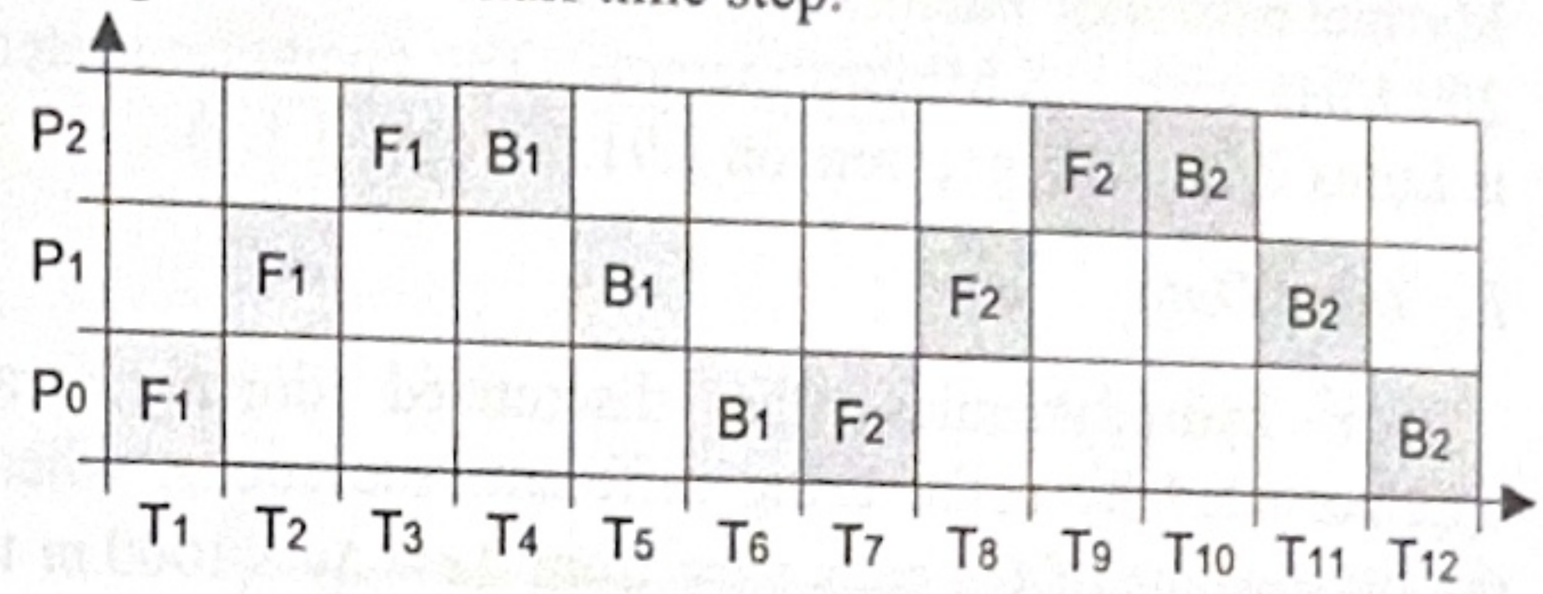


Fig. 5 Scheduling for Thomas algorithm.

To reduce such idleness, a pipelined Thomas is applied. When a processor P_j finishes a forward stage F_i for a system S_i, it sends boundary values to the next processor in the pipeline (P_{j+1}) and begins to perform the forward stage for the system S_{i+1}. When the last processor in the pipeline finishes the backward stage B_i for a system S_i, it sends boundary values to its previous processor in the pipeline and starts to perform the backward stage for the next system S_{i+1}. The scheduling of this algorithm is shown in figure 6, where the entire domain of figure 4 is solved in only 12 periods. This parallelism reduces the idleness in the processors.

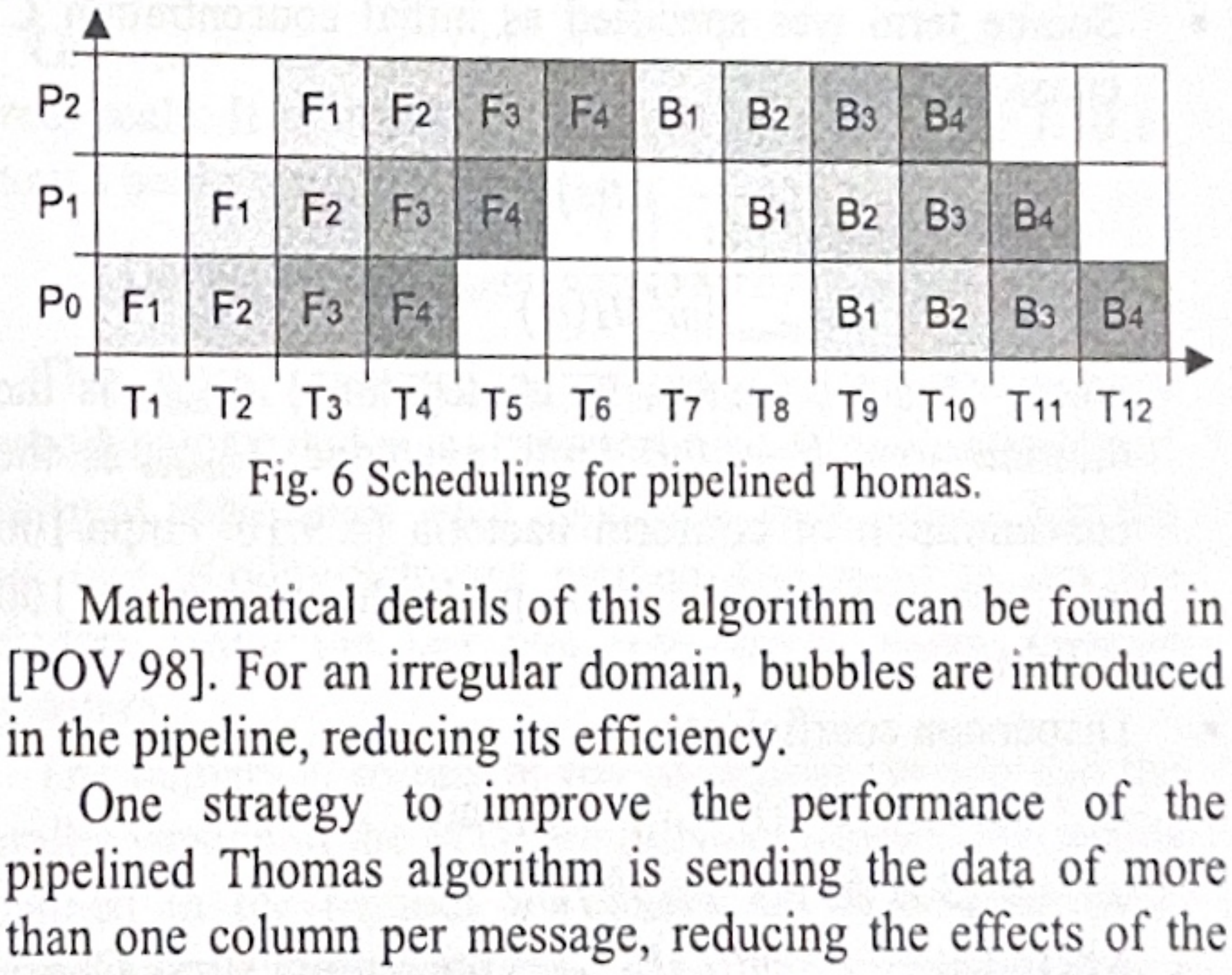


Fig. 6 Scheduling for pipelined Thomas.

Mathematical details of this algorithm can be found in [POV 98]. For an irregular domain, bubbles are introduced in the pipeline, reducing its efficiency.

One strategy to improve the performance of the pipelined Thomas algorithm is sending the data of more than one column per message, reducing the effects of the communication latency. Povitsky [POV 98] comments that it is possible to calculate an optimal number as a function of the computation and communication times. In this work, the number of columns per message was set based on empirical experiments.

VIII. IMPLEMENTATIONS AND RESULTS

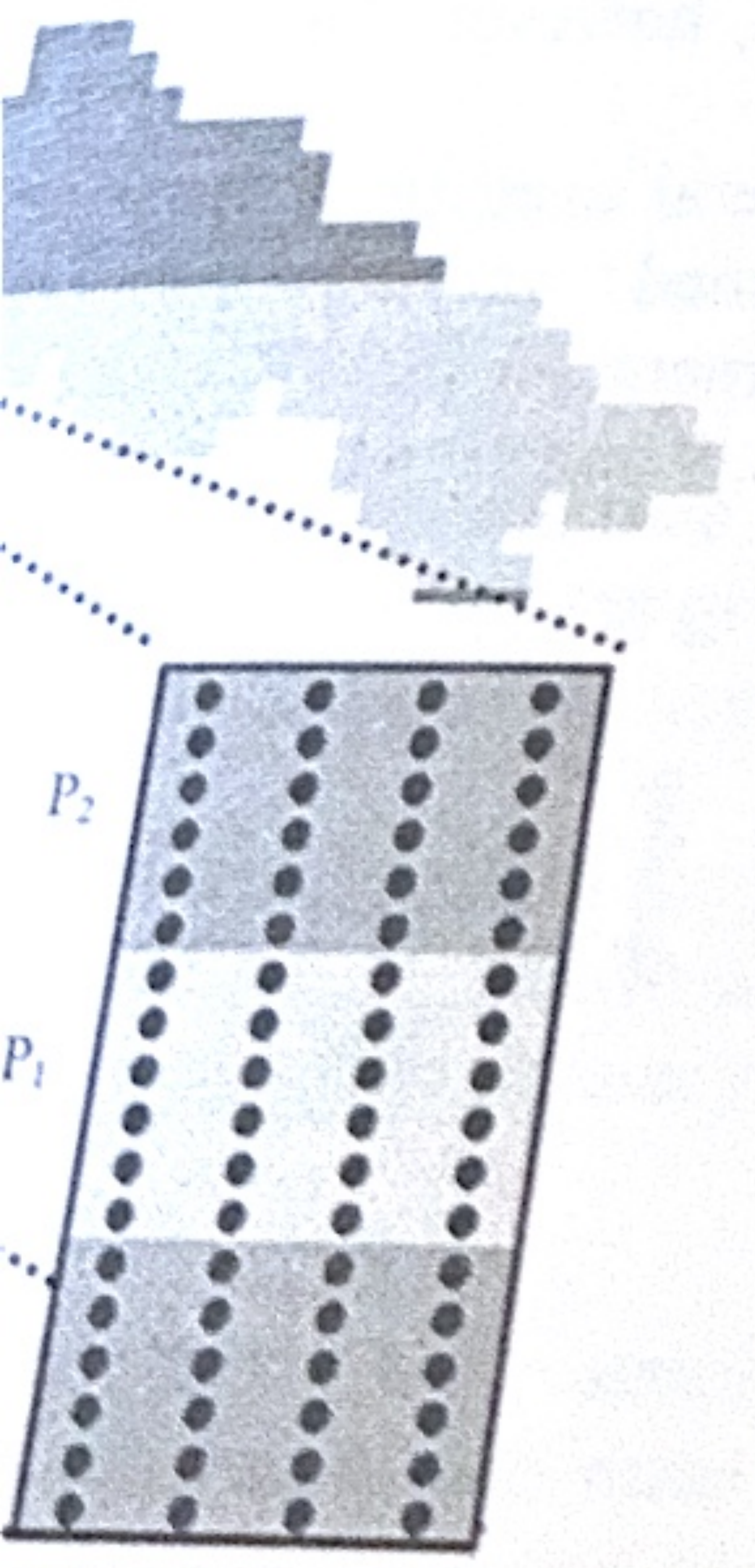
This section presents some results for the implementation above described.

A. *Hardware and Software Platform*

...tion, the following procedure was used. For the first
...e step, each processor applies Thomas algorithm to
... rows doing a row-by-row sweep. In the second
... step, a column-by-column sweep is done. Since
... a sequential algorithm and each column can be
...ween several processors, a direct application of
...uld produce a high idleness because each
...ends on the results of its neighbour. To avoid
... in the second half time step, a pipelined
...mas algorithm is applied [POV 98], as will
...ed. In both cases, the ALES are solved in
...ard and backward.

... (1) in the first half time step has a high
...se each processor can solve its systems
... the other processors. The only data
... on the artificial boundaries, where a 2-
... applied (figure 1) due to the 5-point
...proximation of MTE.

... time step, the solution of one matrix
...ends on the results of P_{j-1}, during the
...f P_{j+1}, during the backward stage.
...quire more communication than in
...d cause idleness in the processors.



... columns. Each processor
... a pipelined fashion.

...ular subdomain where,
...an ALES S_i must be
...listributed among the
...ect application of the
... is very inefficient.
...scheduling for the
... domain shown in
...t, respectively, the
...rmed for a system
...d only after that

the resolution of system S_2 is started. As can be viewed, 12 periods are necessary to solve the first two systems, and each processor stays 8 periods in an idle state. For the domain, 24 periods are necessary to solve the four ALES during this second half time step.

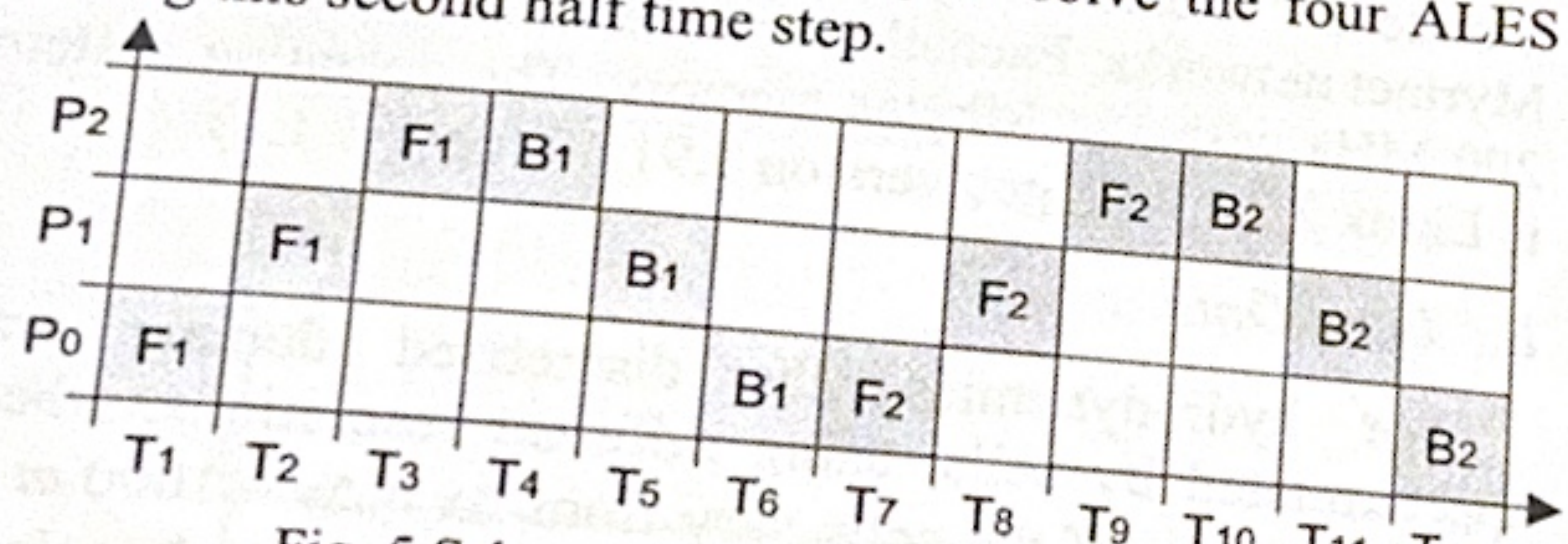


Fig. 5 Scheduling for Thomas algorithm.

To reduce such idleness, a pipelined Thomas is applied. When a processor P_j finishes a forward stage F_i for a system S_i, it sends boundary values to the next processor in the pipeline (P_{j+1}) and begins to perform the forward stage for the system S_{i+1}. When the last processor in the pipeline finishes the backward stage B_i for a system S_i, it sends boundary values to its previous processor in the pipeline and starts to perform the backward stage for the next system S_{i+1}. The scheduling of this algorithm is shown in figure 6, where the entire domain of figure 4 is solved in only 12 periods. This parallelism reduces the idleness in the processors.

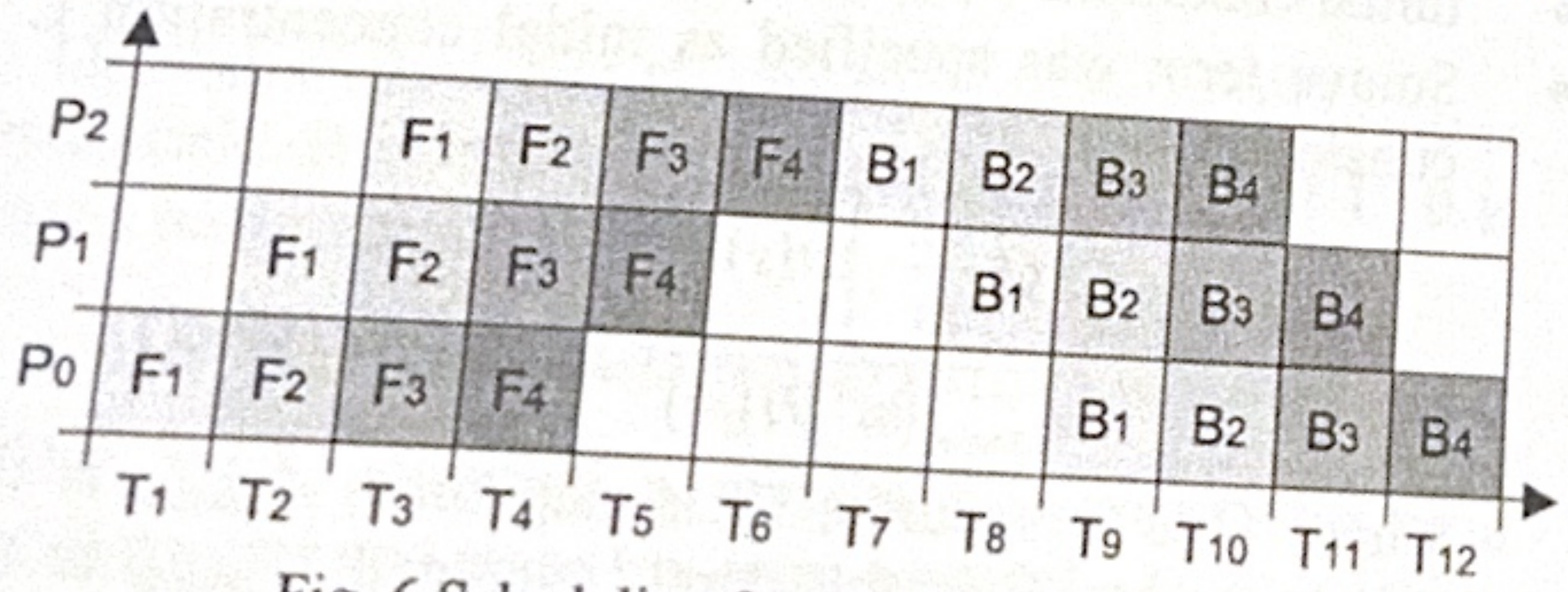


Fig. 6 Scheduling for pipelined Thomas.

Mathematical details of this algorithm can be found in [POV 98]. For an irregular domain, bubbles are introduced in the pipeline, reducing its efficiency.

One strategy to improve the performance of the pipelined Thomas algorithm is sending the data of more than one column per message, reducing the effects of the communication latency. Povitsky [POV 98] comments that it is possible to calculate an optimal number as a function of the computation and communication times. In this work, the number of columns per message was set based on empirical experiments.

VIII. IMPLEMENTATIONS AND RESULTS

This section presents some results for the implementation above described.

A. Hardware and Software Platform

This computational model was implemented in C language using MPI message passing library (MPICH 1.1.2 from the MCS Laboratory [GRO 96]). It was run on a 4-node PC cluster connected by both Fast Ethernet and Myrinet networks. Each cluster node is Dual Pentium Pro at 200 MHz with 128 Mbytes memory. The operating system is Linux 2.2.1 with gcc version 2.91.60 (egcs-1.1.1).

B. Input Data

For hydrodynamics, the discretized domain was approximated by cells, using different resolutions, where the dimensions of the cells vary from $\Delta x = \Delta y = 1000\ m$ to $\Delta x = \Delta y = 50\ m$. With this last choice, the entire domain is composed by about 600.000 cells. In other words, at each time step, CG must solve a 50×30 matrix for the first resolution and a 1000×600 matrix for the last one. For each different resolution, the time step Δt is automatically calculated based on the Courant number.

For mass transport, in this work, all the cells were set with $\Delta x = \Delta y = 25\ m$. This high resolution is necessary to capture the details of the coliform bacteria dispersion and to minimize the generation of negative concentrations, a numerical effect due to the discretization method used. Some of the input parameters for MTE are:

- Initial conditions $C(x,y,0) = 0$ in domain;
- Source term was specified as initial concentration C^n considering that:

$$C^n = C^{n-1} + \frac{Q\left(\frac{m^3}{s}\right)\Delta t(s)}{A_{dilution}(m^2)H(m)} C_{source}(mpn/100ml)$$

  where Q is the source; Δt is step time; $A_{dilution}$ is the dilution area; H is the depth averaged; C_{source} is the concentration of coliform bacteria ($\cong 9.10^6$ mpn/100 ml), where mpn is the mean probable number per 100 milliliter;
- Dispersion coefficients:

  $a_L = 5.93\, H\, (U^2+V^2)^{1/2}\, g^{1/2} / C_h$ and

  $a_T = 0.23\, H\, (U^2+V^2)^{1/2}\, g^{1/2} / C_h$,

  where $C_h = 7{,}83\ ln(0{,}3H/Z_0)$ is the bottom stress Chezy coefficient, and Z_0 is the bottom roughness;
- T_{90}, the time for the removal of 90 percent of the organisms, was considered 4 hours.
- Boundary conditions: in open boundaries, two types of boundary conditions were used; $C(x,y,t) = 0$ in inflow and $\nabla C(x,y) = 0$ in outflow. In closed boundaries it was used $C(x,y,t)=0$.

C. Results for Hydrodynamics

The boundary conditions were set as velocity type (Dirichlet condition) for hydrodynamics and, on the correspondent cells, |U| was set constant and equal to 0.3 m/s at outflow (at left of figure 2) and 0.5 m/s at inflow (at right). The velocity field is shown in figure 7. For mass transport the boundary conditions are Dirichlet condition at inflow and Neumann condition at outflow.

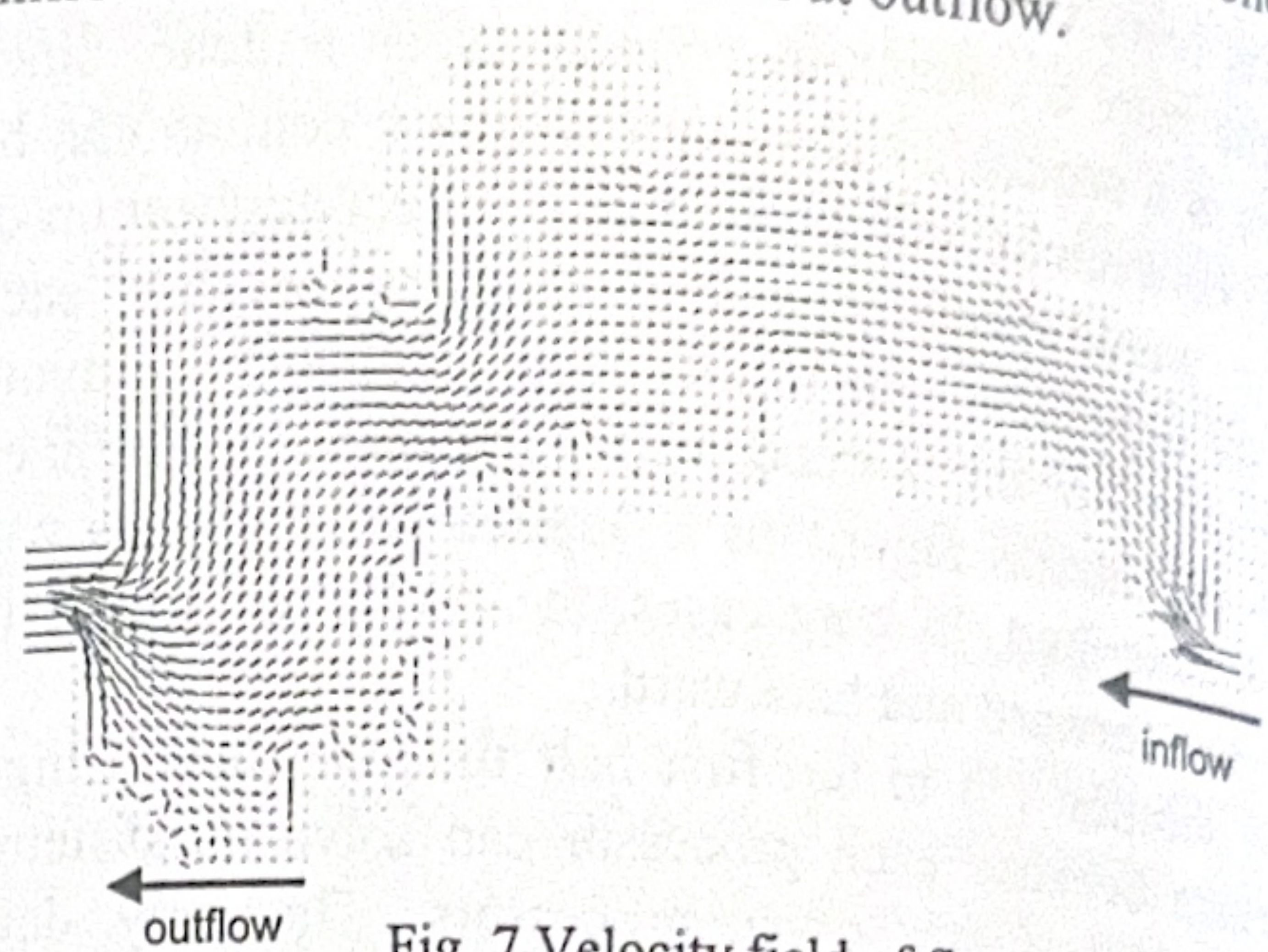


Fig. 7 Velocity field of figure 2.

The implementation of the non-preconditioned CG solved the ALES in 11 iterations. With the polynomial preconditioner, with k=1, the systems are solved in 6 iterations. Nevertheless, the total time (including preconditioning, computation and communication) remains almost the same due the cost to preconditionate the matrices. If k is increased, the number of iterations is reduced, however, the sparseness of the original matrix is lost.

Several executions of this parallel implementation were performed varying the number of processors and refinement on the PC cluster using the Myrinet network. Fig. 8, 9 and 10 resumes some of these results.

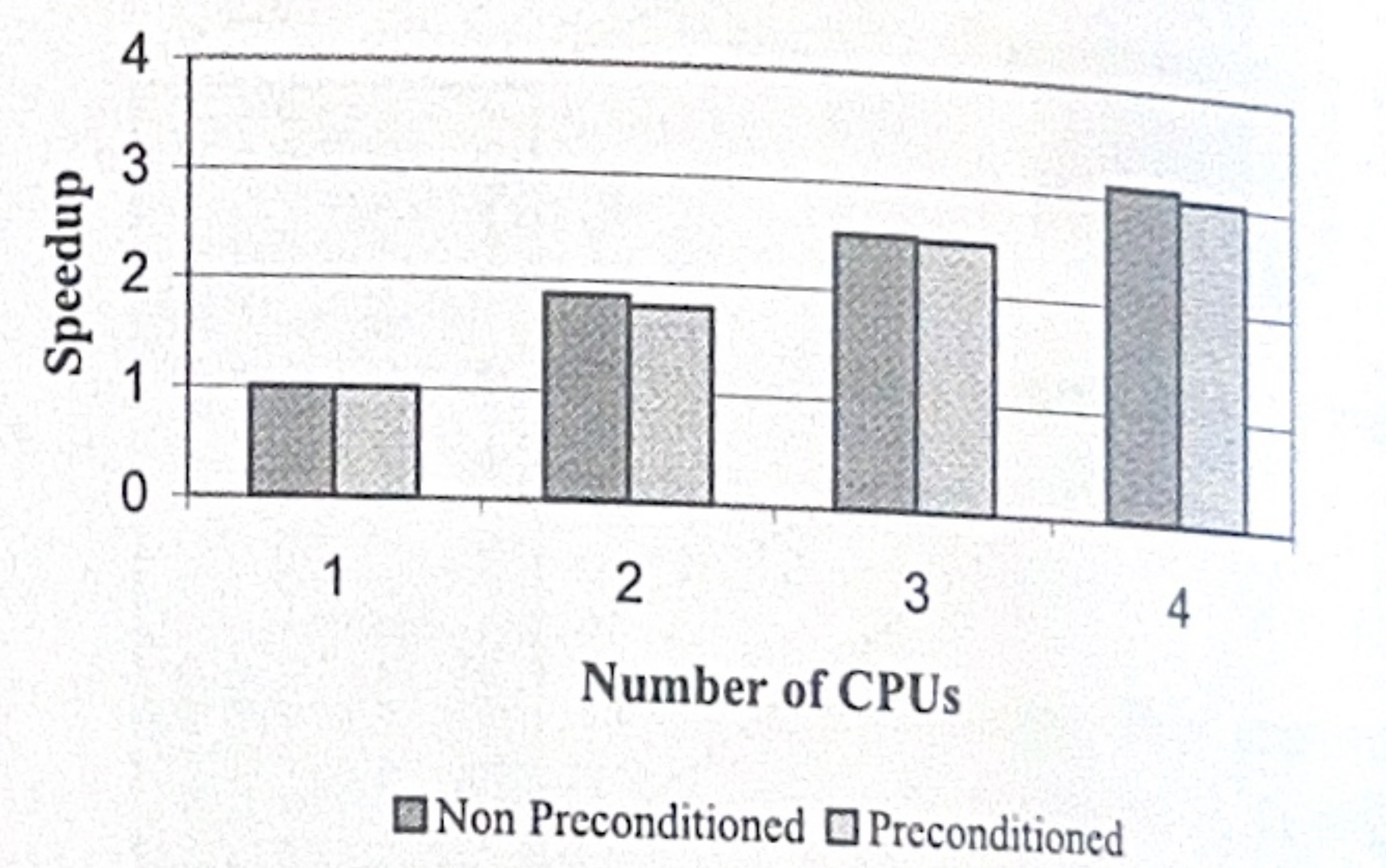


Fig. 8 Speedup for hydrodynamics using Myrinet network.

In figures 8 and 9, each group of two bars represent the speedup and efficiency, respectively obtained with 2, 3 and 4 processors, for a 20×20 refinement.

Speedup is the ratio between the time of sequential execution and the time of parallel execution. Efficiency is the ratio between the speedup and the number of processors used in a given execution. Figures 8 and 9 show, respectively, the speedup increasing and the efficiency decreasing when the number of processors increases. The speedup of PCG algorithm is smaller than the speedup of CG because the first one spends more time in communication than the last one.

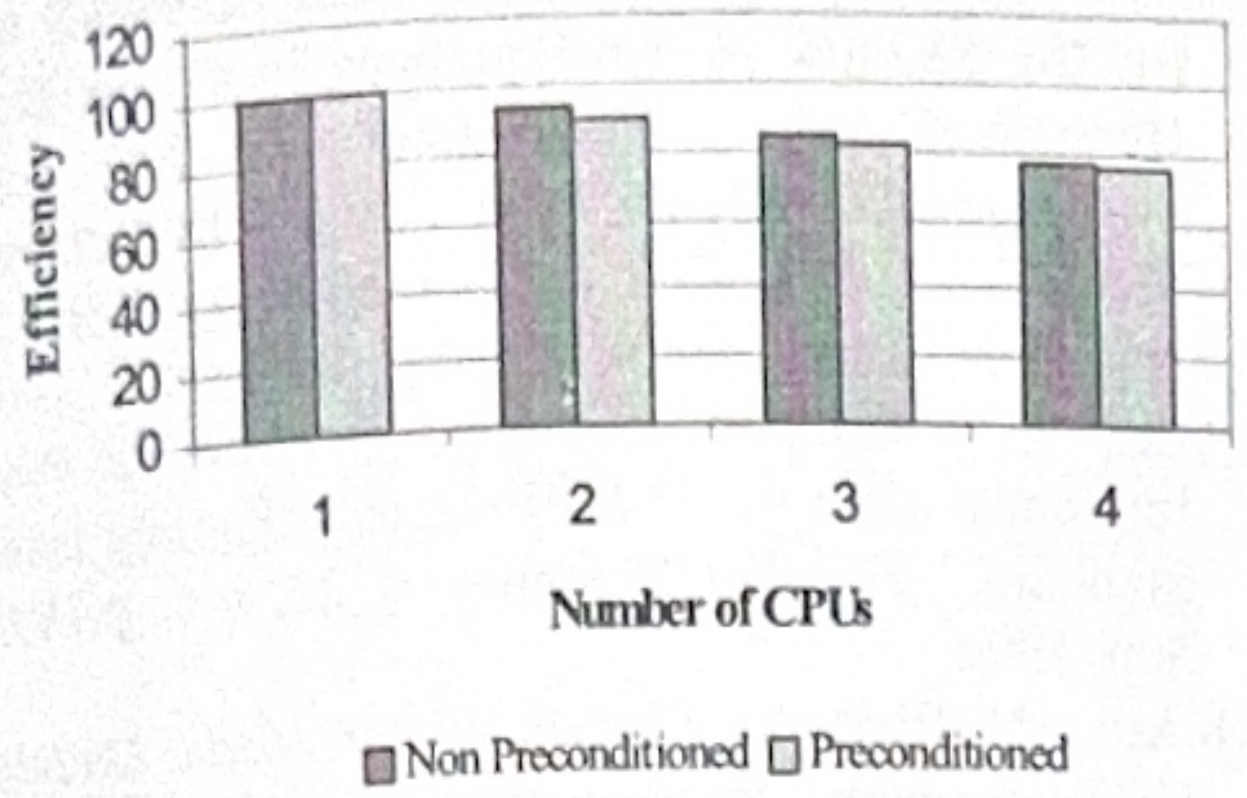


Fig. 9 Efficiency for hydrodynamics using Myrinet network.

For each refinement, efficiency decreases with more processors, as a consequence of the communication/ computation rate.

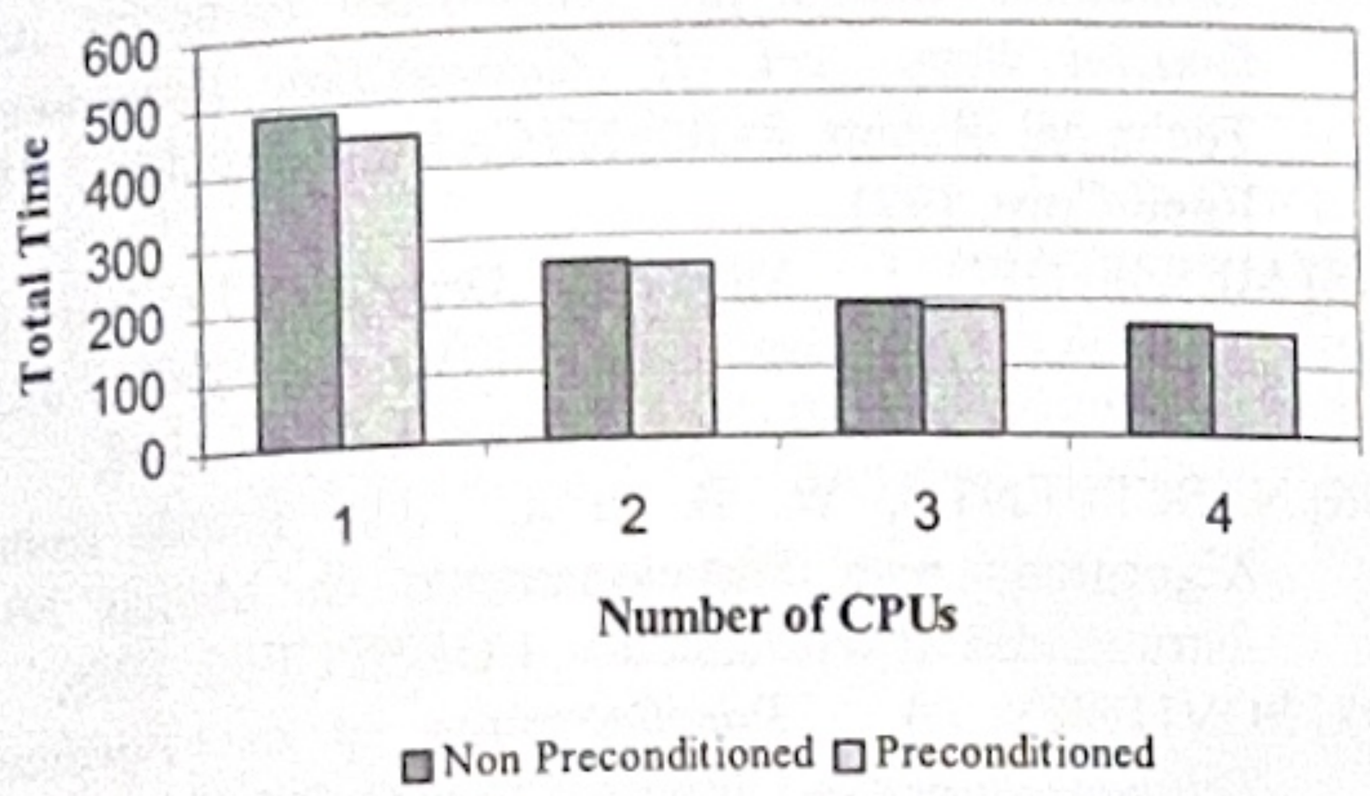


Fig. 10 Total time for CG x PCG algorithms.

The total time is a useful metric to compare the performance of two algorithms acting over the same group of data. The total time is the sum of preconditioning, cycles and communication times.

Figure 10 shows the total time of each algorithm for 30 cycles, where each cycle is composed by the maximum number of iterations required to reach the desired accuracy.

D. *Results for Mass Transport*

As the output of contaminant emitters has small dimensions, there is no sense in simulating the mass transport for non refined meshes. Furthermore, the use of centered finite differences to discretize the advective term in MTE causes numerical oscillations generating strongly negative concentrations. As these oscillations are related to Peclet number, they can be controlled if the mesh is sufficiently refined.

For this reason, in this work, the studies of the mass transport were restricted to cells whose dimensions vary from $\Delta x = \Delta y = 50$ to $\Delta x = \Delta y = 25$ meters.

Figure 11 shows a distribution of concentrations of bacteria coliformes obtained with the computational model for the discharge of 25 l/s, with dilution factor of 25x25x4 and a coliformes concentration of 9.10^6 mpn/100ml (mean probable number per 100 milliliter).

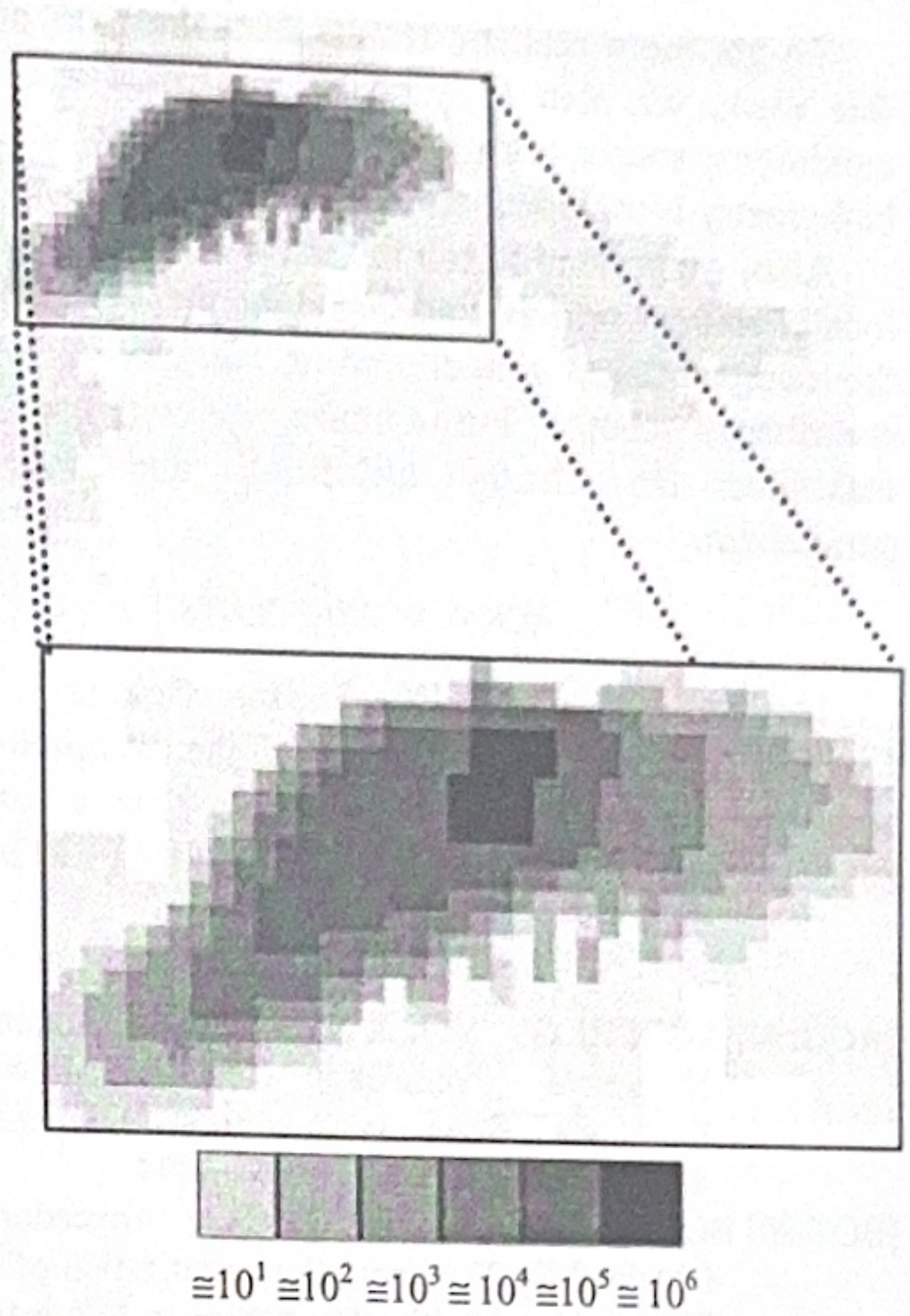


Fig. 11 A subdomain where occurs transport.

One can observe that Guaíba River in figure 11 is not in a real scale. It is in a scale reduced by a factor of 1/10 to allow a better visualization.

X. CONCLUSION AND FUTURE WORKS

This work presented a parallel implementation of a hydrodynamics and mass transport model for Guaíba River. Different approaches were applied to each model. For the first one, Krylov-Schwarz method was used to get the solution, while the last one was solved using Thomas algorithm.

The numerical results of the sequential version and the parallel version of the MTE are fully coincident. The results obtained in the parallel SWEs are not fully coincident, although the differences shown are small. The speedup and efficiency obtained were good, indicating that our code, once adequately calibrated, can be used, in an operational environment.

For future works we intend to improve the mass transport model, developing a semi-implicit numerical scheme for 5-diagonal systems, which will be solved using GMRES algorithm.

Since it is desirable to have a convective scheme that can maintain accuracy, positivity and conservation in the results (to avoid the appearance, for example, of negative concentrations), more appropriate numerical schemes, as semi-Lagrangian (Eulerian-Lagrangian) [CHE 84] or Bott's advection scheme [BOT 88], will be developed later.

To get more realistic results than the ones obtained in this work, we plan to improve the initial and boundary conditions, source term, another constituents and when the bathymetry is included, an interpolation will be performed.

Also, we pretend to run the mass transport model over a locally refined grid and adding dynamic load balancing to the computational model and to develop a generalized coordinates scheme. Furthermore, we will use threads to maximize the cluster utilization and improve the parallelism.

ACKNOWLEDGMENTS

The authors thank Dr. Susana Beatriz Vinzon of COPPE/UFRJ for friendly help in the comprehension of some topics related to the MTE. This research was supported in part by CNPq, CAPES and FAPERGS.

REFERENCES

[AGO 94] AGOSHKOV, V. I. et al. Recent Developments in the Numerical Simulation of Shallow Water Equations I: Boundary Conditions. *Applied Numerical Mathematics*, v.15, p.175-200, 1994.

[BOT 90] BOTT, A. A Positive Definite Advection Scheme Obtained By Nonlinear Renormalization of Advective Fluxes. *Monthly Weather Review*. v.117. p.1006-1015, 1988.

[CAS 85] CASALAS, Alejandro B. *Modelo Matemático de Correntologia do Estuário do Rio Guaíba*. Technical Report 12, Porto Alegre : Instituto de Pesquisas Hidráulicas da UFRGS, 1985.

[CAS 96] CASALAS, Alejandro B. *IPH-A Aplicativo para Modelação de Estuários e Lagoas : Manual de utilização do Sistema*. Porto Alegre : Instituto de Pesquisas Hidráulicas da UFRGS, 1996.

[CAS 90] CASULLI, Vincenzo. Semi-Implicit Finite Difference Methods for the Two-Dimensional Shallow Water Equations. *Journal of Computational Physics*, v.86. p.56-74, 1990.

[CHA 94] CHAN, Tony .F.; MATHEW, T. P. Domain Decomposition Algorithms. *Acta Numerica*, p.61-143. 1994.

[CHE 84] CHENG, R. Eulerian-Lagrangian Solution of the Convection-Dispersion Equation in Natural Coordinates. *Water Resources Research*, v.20. p.944-952. July 1984.

[CHE 98] CHEFFER, J. G. et al. Domain Decomposition for the Shallow Water Equations. Proceendings .of the Ninth International Conference on Domain Decomposition Methods in Scientific and Engineering Computing, 1998. Available at http://www.ddm.org/DD9.

[CIA 94] CIARLET Jr, P. et al. *On the Influence of Partitioning Schemes on the Efficiency of Overlapping Domain Decomposition Methods*. 1994.

[DIE 97] DIEKMANN, Ralf et al. Load Balancing Strategies for Distributed Memory Machines. In *Multi-Scale Phenomena and Their Simulation*. World Scientific. 1997.

[GOO 93] GOOSSENS, S. et al. An Efficient FGMRES Solver for the Shallow Water Equations based on Domain Decomposition. Proceeding of the Seventh International Conference on Domain Decomposition Methods in Scientific and Engineering Computing. Available at http://www.ddm.org. 1993 (Nov. 1999).

[GRO 96] GROOP, William et al. A High Performance, Portable Implementation of the MPI Message Passing Interface Standard. *Parallel Computer*, v.22, n.6, p.789-828, Sep. 1996.

[KAP 98] KAPLAN, Elias A. *Shallow Water Model Distributed Using Domain Decomposition*. Sweden: The Royal Institute of Technology, 1998 (Master Thesis).

[LEE 70] LEENDERTSE, J. J. *A Water-Quality Simulation Model for Well-Mixed Estuaries and Coastal Seas: vol. I, Principles of Computation*. Technical Report RM-6230-RC, Santa Monica : The Rand Corp., 1970.

[LEE 71] LEENDERTSE, J. J; GRITTON, E. C. *A Water-Quality Simulation Model for Well-Mixed Estuaries and Coastal Seas: vol. II, Computation Procedures*. Technical Report R-708-NYC, Santa Monica : The Rand Corp, 1971.

[MES 98] MESSINGER, F. *Numerical Methods: The Arakawa Approach, Horizontal Grid. Global and Limited-Area Modeling*. Camp Springs: Academic Press. 1998.

[NAS 96] NASCIMENTO, W. D. et al. Um Estudo Sobre Algoritmos para Particionamento de Malhas Não Estruturadas. Rio de Janeiro : COPPE/UFRJ. 1996.

[POV 98] POVITSKY, A. *Parallelization of the Pipelined Thomas Algorithm*. NASA/CR-1998-208736 ICASE Report nº 98-48. Institute for Computer Aplication in Science and Engineering. Hampton : NASA Langley Res. Center, 1998.

[RIZ 00] RIZZI, Rogério L., et al. Fluvial Flow of the Guaíba River - A Parallel Solution for the Shallow Water Equations Model. Proceedings of the Fourth Vecpar. 2000. Part III, June 23, p. 895-896.

[SHE 94] SHEWCHUCK, Jonathan. R. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. School of Computer Science*. Carnegie Mellon University, 1994. Available at ftp://WARP.CS.CMU.EDU. (Oct. 1999).

[SIL 97] SILVA, Renato S. et al. *Iterative Local Solvers for Distributed Krylov-Schwarz Method Applied to Convection-Diffusion Problems*. Preprint submitted to Elsevier Science. 1997.

[SMI 96] SMITH, Barry. et al. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University. 1996.

[VAR 99] VARGA, Richard S. *Matrix Iterative Analysis*. 2 ed. rev. and expanded. Springer Series in Computational Mathematics. 1999.

[VOL 97] VOOLEBREGT, Edwin A. H. *Parallel Software Development Techiques for Shallow Water Models*. Delft: Technische Universiteit Delf, 1997 (Ph.D. Thesis).

[ZHE 95] ZHENG, Chunmiao; BENNETT, Gordon D. *Applied Contaminant Transport Modeling*. Van Nostrand Reinhold. 1995.