# Stability Analysis of Numerical Boundary Conditions in Domain Decomposition Algorithms

Wilson Rivera-Gallego
Electrical and Computer Engineering Department
University of Puerto Rico
Mayaguez Campus

## Abstract

This paper discusses numerical stability of a class of non-overlapping domain decomposition algorithms. Inherent shortcomings associated with different methods for proving stability are pointed out. Von Neumann analysis yields only a necessary condition for stability because it does not consider the overall effect of the boundary conditions between subdomains. Matrix analysis produces also a necessary condition for stability since the matrices of coefficients associated with the algorithms are not symmteric. The GKSO, on the other hand, produces necessary and sufficient conditions for stability. However, it is difficult to apply systematically this analysis to the algorithms.

## 1 Introduction

Time dependent partial differential equations (PDEs) are widely used in science and engineering as mathematical models for computational simulations. For example, the following parabolic equation

$$u_t = \nabla \cdot (\alpha \nabla u) + f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \tag{1}$$

with proper boundary and initial conditions, is often used to model diffusion phenomena in heat transfer, flow in porous media, and mathematical biology.

For large-scale problems, the computation of solutions may require substantial CPU time. It is therefore desirable to use multiprocessor parallel computers to calculate solutions. One method widely used for solving time dependent PDEs on parallel computers is domain decomposition [4]. It dates back to the classical Schwarz alternating algorithm with overlapping subdomains [10, 14] for solving elliptic boundary value problems. Note that the original motivation for using domain decomposition method is to deal with complex geometries, equations that exhibit different behaviors in different regions of the domain, and memory restriction for solving large scale problems.

When solving time depending PDEs with non-overlapping subdomains on parallel computers, the domain decomposition method could either be used as a pre-conditioner for Krylov type algorithm [2, 4], or as a means to decompose the original domain into subdomains and solve the PDEs defined in different subdomains concurrently [3, 6]. When it is used as a preconditioner, the relevant PDE is discretized over the entire original domain to form a large system of algebraic equations, which is then solved by Krylov type iterative algorithms. The pre-conditioning step and the inner products involved in the solution process often incur a significant amount of communication cost to gather results from all processors.

On the other hand, if the original domain is decomposed into a set of non-overlapping subdomains, it would be ideal that the PDEs defined in different subdomains could be solved on different processors concurrently. This often requires numerical boundary conditions at the boundaries between subdomains. These numerical boundary conditions are not part of the original mathematical model and the physical problem. One way to generate those numerical boundary conditions is to use the solution values from the previous time step $t_n$ to calculate the solutions at $t_{n+1}$ [1, 11]. This is often referred to as time lagging (TL). The other way to generate numerical boundary conditions is to use an explicit algorithm to calculate the solutions at the boundaries between subdomains, using the solutions from the previous time step, and then solve the PDEs defined on different subdomains concurrently using an implicit method [5, 9]. This is referred to as the explicit predictor (EP) method. It was showed in [12] that the stability and accuracy of the solution algorithm can be significantly affected by how the numerical boundary conditions are generated.This paper is motivated by the interest in numerical stability associated with methods for generating boundary conditions between subdomains. The general theory for the analysis of numerical interface or boundary conditions is well-established but can be complicated to apply in practice [7].

The objective in this paper is to bring to light and discuss the problems that arise when different methods are used for proving stability of domain decomposition algorithms. Next section will discuss the domain decomposition strategy and specifically the EP method. The different stability analyses including von Neumann, matrix analysis and GKSO theory will be discussed in Section 3. A numerical experiment will be presented in Section 4, followed by the conclusions in Section 5.

## 2 Domain Decomposition

For simplicity of the discussion, consider the following one-dimensional model with a constant diffusion coefficient $\alpha$

$$\begin{aligned} u_t &= \alpha u_{xx} + f(x,t), \quad 0 < x < 1, \quad 0 < t \leq T, \\ & u(0,t) = g(t), \quad u(1,t) = h(t), \quad t > 0, \\ & \quad u(x,0) = l(x), \quad 0 \leq x \leq 1. \end{aligned} \tag{2}$$

The analysis here can be generalized to higher dimensional cases with variable diffusion coefficients, although the details will be much more tedious. Also, we will ignore the nonhomogeneous term $f(x,t)$ in (2.1) hereafter since it does not affect the result of the analysis.

The first step to solve (2.1) is to discretize the continuous spatial and temporal domains. Assume that the spatial domain $[0,1]$ is discretized by a set of grid points $x_i, i = 0, \cdots, L$, uniformly distributed with $\Delta x = x_i - x_{i-1} = \frac{1}{L}$, and the temporal domain $[0,T]$ is discretized by a set of discrete time steps $t_n, n = 0, \cdots, N$, with $\Delta t = t_n - t_{n-1} = \frac{T}{N}$. The numerical solution $u(x_i, t_n)$ is denoted by $u_i^n$.

Also, assume that the original spatial domain $\Omega$ is decomposed into $M$ subdomains $\Omega_k, k = 1, \cdots, M$. The two end points of subdomain $\Omega_k$ are denoted as $r_{k-1}$ and $r_k$, respectively.

Since only two physical boundary conditions are available at the points $r_0$ and $r_M$, numerical boundary conditions are needed at points $r_k$, $k = 1, \cdots, M-1$, if the PDEs defined in different subdomains are to be solved concurrently using an implicit algorithm.
For the EP method, the solution values of the boundaries between subdomains at the new time level $t_{n+1}$ are generated by an explicit method. For example, the following forward time central space (FTCS) algorithm

$$\bar{u}^{n+1}_{r_k} = u^n_{r_k} + \gamma(u^n_{r_k-1} - 2u^n_{r_k} + u^n_{r_k+1}), \qquad k = 1, \ldots, M-1, \tag{3}$$

can be used to provide numerical boundary conditions at points $r_k, k = 1, \cdots, M-1$. With these boundary conditions, (2.1) can be discretized in subdomain $\Omega_k$ using an implicit algorithm, such as the backward time central space (BTCS) algorithm,

$$\begin{aligned}
u^{n+1}_{r_{k-1}+1} &= u^n_{r_{k-1}+1} + \gamma(\bar{u}^{n+1}_{r_{k-1}} - 2u^{n+1}_{r_{k-1}+1} + u^{n+1}_{r_{k-1}+2}), \\
u^{n+1}_j &= u^n_j + \gamma(u^{n+1}_{j-1} - 2u^{n+1}_j + u^{n+1}_{j+1}), \qquad j = r_{k-1}+2, \ldots, r_k - 2, \\
u^{n+1}_{r_k-1} &= u^n_{r_k-1} + \gamma(u^{n+1}_{r_k-2} - 2u^{n+1}_{r_k-1} + \bar{u}^{n+1}_{r_k}),
\end{aligned} \tag{4}$$

which is equivalent to solving a linear system of equations in each subdomain:

$$\begin{bmatrix}
1+2\gamma & -\gamma & 0 & \cdots & 0 \\
-\gamma & 1+2\gamma & -\gamma & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & -\gamma \\
0 & \cdots & 0 & -\gamma & 1+2\gamma
\end{bmatrix}
\begin{bmatrix}
u^{n+1}_{r_{k-1}+1} \\
u^{n+1}_{r_{k-1}+2} \\
\vdots \\
\vdots \\
u^{n+1}_{r_k-1}
\end{bmatrix}
=
\begin{bmatrix}
u^n_{r_{k-1}+1} + \gamma\bar{u}^{n+1}_{r_{k-1}} \\
u^n_{r_{k-1}+2} \\
\vdots \\
\vdots \\
u^n_{r_k-1} + \gamma\bar{u}^{n+1}_{r_k}
\end{bmatrix}. \tag{5}$$

A new approach based on explicit predictors and implicit correctors (EPIC) for the solution of convection-diffusion equations was proposed in a previous paper [13]. The results demonstrated stability of the scheme and a significant improvement in accuracy when calculating transient solutions.
Since our goal in this paper is to illustrate the difficulties in stability analysis of these algorithms, we can choose just one method for the discussion. We have selected the EP method because it is only conditionally stable and one of the problems precisely is to determine the stability condition correctly.

# 3 Stability Analysis

## 3.1 von Neumann Analysis

We have that the numerical boundary conditions between subdomains at points $r_k$, $k = 1, \cdots, M-1$, are given by

$$\tilde{u}^{n+1}_{r_k} = u^n_{r_k} + \gamma\left(u^n_{r_k-1} - 2u^n_{r_k} + u^n_{r_k+1}\right). \tag{6}$$

This provides boundary data for the equation at the point $r_k - 1$

$$u^{n+1}_{r_k-1} = u^n_{r_k-1} + \gamma(u^{n+1}_{r_k-2} - 2u^{n+1}_{r_k-1} + \tilde{u}^{n+1}_{r_k}), \tag{7}$$

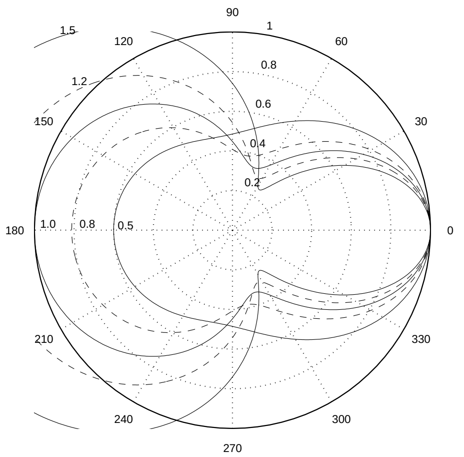

Figure 1: Amplification factor $\xi_{r_k}(\theta)$ for the EP algorithm: $\gamma = 0.5, 0.8, 1.0, 1.2, 1.5$.

which leads to

$$\xi_{r_k} = \frac{u^{n+1}_{r_k-1}}{u^n_{r_k-1}} = \frac{u^n_{r_k-1} + \gamma(u^{n+1}_{r_k-2} - 2u^{n+1}_{r_k-1} + \tilde{u}^{n+1}_{r_k})}{u^n_{r_k-1}}. \tag{8}$$

Solving for $\xi_{r_k}$, we have

$$\xi_{r_k} = \frac{\gamma^2 e^{i2\theta} + \left(\gamma - 2\gamma^2\right) e^{i\theta} + \gamma^2 + 1}{1 + 2\gamma - \gamma e^{-i\theta}}. \tag{9}$$

Figure 3.1 shows the curves for the modulus of the amplification factor $|\xi_{r_k}|$ for different values of $\gamma$ with $0 \leq \theta < 2\pi$. It can be observed that $|\xi_{r_k}| \leq 1$ for $\gamma \leq 1$. When $\gamma > 1$, the curve of $\xi_{r_k}(\theta)$ goes outside of the unit circle. This would lead to the conclusion that the EP scheme is only conditionally stable when $\gamma \leq 1$.

However, it is well-known that von Neumann analysis gives us only a necessary condition for stability. The only class of problems for which von Neumann analysis provides also sufficient conditions is the class of initial-value problems with periodic boundary conditions. The reason is that the analysis is based on finite Fourier modes which do not represent boundary conditions in general. If we use the appropriate basis functions then we should obtain the necessary and sufficient conditions for stability. Unfortunately, the problem of determining generalized basis functions for a specific scheme may be very difficult in most cases. In addition, engineering applications are increasingly using finite volume and finite element methods based on unstructured grids, for which Fourier stability analysis is not applicable. Thus, one must instead consider the full discrete matrix that arise from the combined spatial and temporal discretization of the PDEs and associated boundary conditions.

## 3.2 Matrix Analysis

For a general domain $\Omega = \{x_i, i = 0, \cdots, L\}$ and $M$ subdomains with equal number of grid points, assuming $m = \frac{L}{M}$ is an integer, the matrix representation of the EP algorithm can be written as

$$\begin{bmatrix} A & v & & & & & & & \\ & 1 & & & & & & & \\ & w & A & v & & & & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & w & A & v & \\ & & & & & & & 1 & \\ & & & & & & & w & A \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ u_{r_1} \\ \mathbf{u}_2 \\ u_{r_2} \\ \vdots \\ \mathbf{u}_{M-1} \\ u_{r_{M-1}} \\ \mathbf{u}_M \end{bmatrix}^{n+1} = \begin{bmatrix} I & & & & & & & \\ y & \sigma & z & & & & & \\ & & I & & & & & \\ & & y & \sigma & z & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & & I & & \\ & & & & & y & \sigma & z \\ & & & & & & & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ u_{r_1} \\ \mathbf{u}_2 \\ u_{r_2} \\ \vdots \\ \mathbf{u}_{M-1} \\ u_{r_{M-1}} \\ \mathbf{u}_M \end{bmatrix}^{n}, \tag{10}$$

where $\sigma = 1 - 2\gamma$, $A$ is a matrix of order $m-2$ with similar structure as that in (2.4), $I$ is an identity matrix of order $m-2$, $\mathbf{u}_k,\ k = 1, \cdots, M$, represent the solution vector in the interior of the $k$-th subdomain $\Omega_k$ (not including the two end points $u_{r_{k-1}}$ and $u_{r_k}$), and the vectors $v, w, y, z$ are defined as

$$v = \{0, \cdots, 0, -\gamma\}^T, \qquad w = \{-\gamma, 0, \cdots, 0\}^T, \qquad y = -v^T, \qquad z = -w^T.$$

The above equation can be written in a compact form as

$$C\mathbf{u}^{n+1} = F\mathbf{u}^n.$$

It is well-known that the necessary condition for stability is $\rho(C^{-1}F) = max|\lambda_j| \leq 1, j = 1, \ldots, L-1$, where $\lambda_j$'s are the eigenvalues of matrix $C^{-1}F$ [8, 15]. Since it is usually difficult to obtain an analytic expression of the eigenvalues for this matrix, the software package MATLAB is used to calculate the magnitude of the largest eigenvalue.

Figure 3.2 shows the curve of $\rho(C^{-1}F)$ vs. the value of $\gamma$ with $L = 200$ and $M = 10$. It can be observed that $\rho(C^{-1}F)$ stays close to but below 1 for the value of $\gamma$ between 0 and 1.5. When $\gamma$ increases beyond 1.5, $\rho(C^{-1}F)$ quickly goes above 1. Figure 3.3 shows the curve of $\rho(C^{-1}F)$ vs. the number of subdomains $M$ with $L = 200$, $\gamma = 1.2$ and $\gamma = 1.8$, respectively. It can be observed that the value of $\rho(C^{-1}F)$ does not change significantly as the number of subdomains varies, with $\rho(C^{-1}F)$ remaining below 1 for $\gamma = 1.2$ and greater than 1 for $\gamma = 1.8$.

This demonstrates that the EP method is only conditionally stable, with a necessary condition of $\gamma \leq 1.5$ for stability.

Note that even if the implicit BTCS algorithm, which is unconditionally stable, is used to calculate solutions at all interior points of each subdomain, the use of an explicit predictor for the numerical boundary conditions at the points $r_k,\ k = 1, \cdots, M-1$, between the subdomains makes the final algorithm only conditionally stable. This is true even if the explicit predictor is used for only one point with two subdomains.

Since the solutions at the points $r_k$ are calculated using an explicit FTCS algorithm, one might think that the stability of the final algorithm is governed by the stability condition of the FTCS algorithm. This, however, would lead to a more restrictive condition of $\gamma \leq \frac{1}{2}$ [5], as opposed to the condition $\gamma \leq \frac{3}{2}$ obtained using matrix analysis.

A numerical scheme $u^{n+1} = Au^n$ with $A$ symmetric is stable with respect to the Euclidean norm if and only if $\rho(A) \leq 1$. It follows from the fact that for symmetric matrices $\rho(A) = \|A\|_2$. Indeed, it can be proved that for any matrix norm $\rho(A) \leq \|A\|$. Since the matrix $C^{-1}F$ is nonsymmetric, we can not guarantee that the condition $\gamma \leq \frac{3}{2}$ is also sufficient condition. However, extensive numerical experiments have demonstrated that it is also sufficient to maintain stability [13].

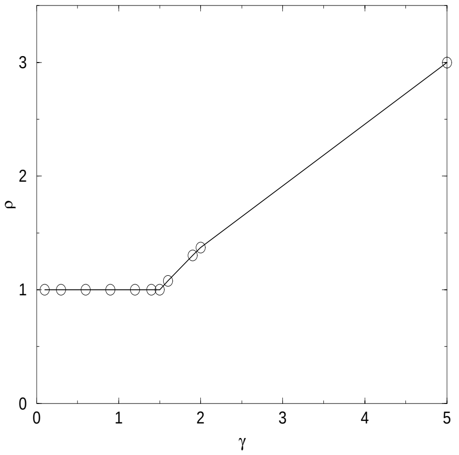

Figure 2: Spectral radius $\rho$ vs. $\gamma = \frac{\Delta t}{\Delta x^2}$: EP method, $L = 200$, $M = 10$.

### 3.3 GKSO Theory

While the standard Fourier analysis is based on a particular value of $t$, the GKSO analysis consists of summing over all $n$ values. Thus, stability is proved for all $t > 0$ as well as the given spatial region. GKSO analysis investigates the existence of separable normal modes of the form

$$u_j^n = z^n \xi^j. \tag{11}$$

The form of the solution is very similar to the assumed Fourier modes, except that it is compatible with the discrete Laplace transform. The discretization is unstable if the difference equation admits such solution for which $|z| > 1$ allowing exponential growth in time.
The difference equation at the boundary and adjacent points are given by

$$\begin{aligned} u_{r_k-1}^{n+1} &= u_{r_k-1}^{n} + \gamma\{u_{r_k-2}^{n+1} - 2u_{r_k-1}^{n+1} + u_{r_k}^{n+1}\}, \\ u_{r_k}^{n+1} &= u_{r_k}^{n} + \gamma\{u_{r_k-1}^{n} - 2u_{r_k}^{n} + u_{r_k+1}^{n}\}, \\ u_{r_k+1}^{n+1} &= u_{r_k+1}^{n} + \gamma\{u_{r_k}^{n+1} - 2u_{r_k+1}^{n+1} + u_{r_k+2}^{n+1}\}. \end{aligned} \tag{12}$$

Applying discrete Laplace transform, we have

$$\begin{aligned} & z\{-\gamma\xi_-^{-1} + (1+2\gamma) - \gamma\xi_-\} = 1, \\ & z = 1 + \gamma\{\xi_-^{-1} - 2 + \xi_+\}, \\ & z\{-\gamma\xi_+^{-1} + (1+2\gamma) - \gamma\xi_+\} = 1. \end{aligned} \tag{13}$$

As a consequence,

$$\xi_- = \frac{-1 + z + 2z\gamma - (1 - 2z - 4z\gamma + z^2 + 4z^2\gamma)^{1/2}}{2z\gamma}, \tag{14}$$

$$\xi_+ = \frac{-1 + z + 2z\gamma + (1 - 2z - 4z\gamma + z^2 + 4z^2\gamma)^{1/2}}{2z\gamma}. \tag{15}$$

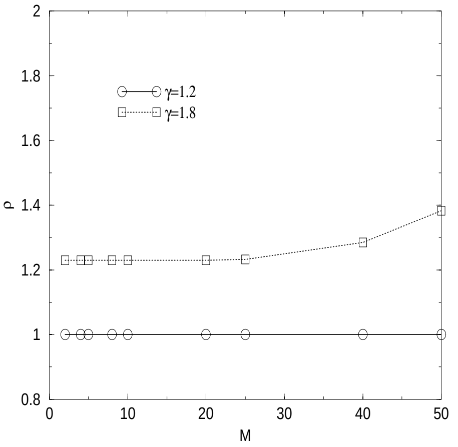

Figure 3: Spectral radius $\rho$ vs. number of subdomains $M$: EP method, $L = 200$, $\gamma = 1.2$ and 1.8.

Substituting into the second equation of (3.8) we obtain

$$z = \frac{-1 + z + 2\gamma + (2\gamma - 1)(1 - 2z - 4z\gamma + z^2 - 4z^2\gamma)^{1/2}}{-1 + z + 2z\gamma - (1 - 2z - 4z\gamma + z^2 + 4z^2\gamma)^{1/2}}, \tag{16}$$

and solving for $z$ we have

$$z = \frac{3}{2} - \frac{1}{2}(1 + 16\gamma)^{1/2}. \tag{17}$$

Hence, if $\gamma > \frac{3}{2}$ then $|z| > 1$, and consequently the scheme is unstable.

# 4 Numerical Experiments

Consider the following equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - 12x^2, \quad x \in \Omega = (0,1), \quad t \in [0,1], \tag{18}$$

with the initial and boundary conditions

$$u^*(x,0) = \sin \pi x + x^4, \quad u(0,t) = 0, \quad u(1,t) = 1,$$

and the exact solution $u^*(x,t) = e^{-\pi^2 t} \sin \pi x + x^4$. The FTCS scheme is used as predictor and the BTCS scheme is used to solve the subdomains.

Table 4.1 contains results from the EP method for solving (4.1) with different values for $\gamma = \frac{\Delta t}{\Delta x^2}$. Based on the spectral analysis in Section 3, it is known that the necessary condition for maintaining stability of the EP method is $\gamma \leq 1.5$. The numerical results demonstrate that the condition is also sufficient for stability. For the cases of $\gamma = 1.0$ and $\gamma = 1.5$, accurate numerical results were obtained. But for the case of $\gamma = 1.6$, the errors grow explosively as the grid is being refined.

Table 1: Maximum errors: EP method, $M = 2$

| $\Delta x$ | $\gamma = 1.0$, $T = 0.1$ | $\gamma = 1.5$, $T = 0.15$ | $\gamma = 1.6$, $T = 0.16$ |
|---|---|---|---|
| 0.1000 | 0.1389E-01 | 0.2555E-01 | 0.3684E-01 |
| 0.0500 | 0.4545E-02 | 0.6433E-02 | 0.1829E-01 |
| 0.0250 | 0.1277E-02 | 0.1699E-02 | 0.6959E+01 |
| 0.0100 | 0.2180E-03 | 0.2853E-03 | 0.8121E+27 |
| 0.0050 | 0.5565E-04 | 0.7264E-04 | 0.3017E+125 |
| 0.0025 | 0.1406E-04 | 0.1833E-04 | $\infty$ |

## 5 Conclusions

We have discussed relevant issues regarding numerical stability of a class of non-overlapping domain decomposition algorithms. The issues presented here should be taken account for proving stability of discretization schemes in general. In the particular case of the EP algorithm, von Neumann analysis yields only a necessary condition for stability because it does not consider the overall effect of the boundary conditions between subdomains. Moreover, von Neumann analysis gives us a strong stability condition of $\gamma \leq 1$. In fact, the stability condition of the EP algorithm is more flexible as shown in matrix analysis where the stability condition is $\gamma \leq 1.5$. Since the matrix of coefficients associated with the EP algorithm is nonsymmetric, the stability condition obtained using matrix analysis is also a necessary condition. However, numerical experiments have shown that this condition is also sufficient for stability. In addition, it is not possible to express the eigenvalues of the matrix of coefficients in a close form so a rigorous analytical proof of stability can not be carried out. As a consequence, MATLAB has been used to investigate the behavior of the spectral radius.

Finally, the GKSO analysis has become the standard for proving stability of initial-boundary-value problems. For the EP method, the GKSO analysis has also provided a stability condition of $\gamma \leq 1.5$. However, we have found difficulties in applying systematically GKSO analysis to our algorithms. For example, for the TL method $z = 1$ and no conclusions about stability can be deduced from the analysis, and for the EPIC method, it is not possible to apply GKSO analysis reasonably.

## References

[1] S. Barnard, S. Saini, R. Van der Wijngaart, M. Yarrow, L. Zechtzer, I. Foster, and O. Larsson, "Large-scale distributed computational fluid dynamics on the information power grid using Globus," in Proceedings of the 7th Symposium on the Frontiers of Massively Parallel Computation, 1999.

[2] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods," SIAM: Philadelphia, 1994.

[3] E. Brakkee, A. Segal, and C. G. M. Kassels, "A parallel domain decomposition algorithm for the incompressible Naiver-Stokes equations," *Simulation Practice and Theory,* 3 (1995), pp. 185-205.

[4] T. F. Chan and T. P. Mathew, "Domain decomposition algorithms," *Acta Numerica,* 3 (1994), pp. 61-143.

[5] C. N. Dawson, Q. Du, and T. F. Dupont, "A finite difference domain decomposition algorithm for numerical solution of the heat equations," *Mathematics of Computation*, **57** (1991), pp. 63 - 71.

[6] D.Drikakis and E. Schreck, "Development of parallel implicit Navier-Stokes solvers on MIMD multi-processor systems," *AIAA* 93-0062.

[7] B. Gustafsson, H. Kreiss and J. Oliger,"Time Dependent Problems and Difference Methods," John Wiley and Sons: New York, 1995.

[8] C. Hirsch, "Numerical Computation of Internal and External Flows," Volume I, John Wiley and Sons: New York, 1988.

[9] Y. A. Kuznetsov, "New algorithms for approximate realization of implicit difference schemes," *Sovietic Journal of Numerical Analysis and Mathematical Modeling,* 3 (1988), pp. 99-114.

[10] P.L. Lions, "On the Schwartz alternating method I," *First International Symposium on Domain Decomposition Methods for Partial Differential Equations,* R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., SIAM, Philadelphia, 1988, pp. 1-42.

[11] R. Pankajakshan and W. R. Briley, "Parallel solution of viscous incompressible flow on multi-block structured grids using MPI," *Parallel Computational Fluid Dynamics: Implementation and Results Using Parallel Computers,* A. Ecer, J. Periaux, N. Satofuka, and S. Taylor, ed., Elsevier Science, Amsterdam, 1996, pp. 601-608.

[12] W. Rivera and J. Zhu, "A scalable parallel domain decomposition algorithm for solving time dependent partial differential equations," *Proceedings of the 1999 International Conference on Parallel and Distributed Processing Technology and Applications*, H. R. Arabnia Ed., CSREA Press, Athens, GA 1999, pp. 240-246.

[13] W. Rivera, J. Zhu and D. Huddleston, "An efficient parallel algorithm with application to computational fluid dynamics," *To appear in Computers & Mathematics with Applications.*

[14] H.W. Schwartz, "Gesammelete Mathematische Abhandlungen," 2 (1890), pp. 133-143.

[15] J. W. Thomas, "Numerical Partial Differential Equations: Finite Difference Methods," Springer-Verlag: New York, 1995.