NASA Contractor Report 181629

ICASE REPORT NO. 88-15

ICASE

PARALLEL PSEUDOSPECTRAL DOMAIN DECOMPOSITION TECHNIQUES

David Gottlieb

Richard S. Hirsh

(NASA-CR-181629) PARALLEL PSEUDOSPECTRAL N88-19101
DOMAIN DECOMPOSITION TECHNIQUES Final Report
(NASA) 29 p CSCL 12A
Unclas
G3/59 0129082

Contract No. NAS1-18107
February 1988

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

NASA
National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23665

PARALLEL PSEUDOSPECTRAL DOMAIN DECOMPOSITION TECHNIQUES

David Gottlieb[*] and Richard S. Hirsh[**]

ABSTRACT

The influence of interface boundary conditions on the ability to parallelize pseudospectral multidomain algorithms is investigated. Using the properties of spectral expansions, a novel parallel two domain procedure is generalized to an arbitrary number of domains each of which can be solved on a separate processor. This interface boundary condition considerably simplifies influence matrix techniques.

[*] Department of Applied Mathematics, Brown University, Providence, RI.
[**] Johns Hopkins Applied Physics Laboratory, Laurel, MD; presently at National Science Foundation, Washington, DC.

This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18107 while the authors were in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23665.

INTRODUCTION

Domain decomposition techniques have become a popular way to numerically solve partial differential equations, especially when geometric complexity needs to be addressed [1,2]. The introduction of parallel computers seems naturally suited for these techniques because each separate domain can be assigned to its own processor. These techniques are especially appropriate for use with pseudospectral methods due to both the global nature of the spectral derivative matrices and the high condition number of these matrices as the number of collocation points increases.

However, the interconnection of the individual domains, and the method of solution of the overall numerical problem have not yet taken advantage of the parallelism inherent in multi-domain techniques [3]. In the present work, we investigate interface boundary conditions and show that a judicious choice of these conditions leads naturally to an algorithm which has a high degree of concurrency, even when pseudospectral methods are applied to each of the subdomains.

Several methods will be presented for a model elliptic problem. Each of these relies upon the properties of spectral methods for its applicability to domain decomposition techniques. For a simple two domain problem, a novel concurrent forward/backward elimination procedure is used. In order to generalize this result to three or more domains, a different interface boundary condition, that makes use of the special properties of spectral methods, is introduced. These procedures are also discussed in relation to influence matrix techniques.

SECTION I

Certain features of pseudospectral methods, relevant to the issue of parallel computing, can be identified by considering the pseudospectral Legendre (PSL) discretization of the simple equation

$$u_{xx} = f(x) \qquad -1 \leq x \leq 1 \tag{1.1}$$

with the boundary conditions

$$\begin{aligned} \alpha\, u(-1) + \beta u_x(-1) &= b_1 \\ \gamma\, u(1) + \beta u_x(1) &= b_2 . \end{aligned} \tag{1.2}$$

Let $x_j \quad j = 0,\cdots,N$ be defined as follows

$x_0 = -1$, $x_N = +1$, and x_j, $j = 1,\cdots,N-1$ are the extrema of p_N, i.e.,

$$P_N'(x_j) = 0 \tag{1.2}$$

where p_N is the Nth order Legendre polynomial.

In the PSL discretization, one seeks a polynomial u_N, of degree N, such that

$$\frac{\partial^2 u_N}{\partial x^2} = f(x) \quad \text{at} \quad x = x_j \quad j = 1,\cdots,N-1 \qquad \text{(a)}$$

$$\alpha\, u_N(-1) + \beta \frac{\partial u_N}{\partial x}(-1) = b_1 \qquad \text{(b)} \qquad (1.3)$$

$$\gamma\, u_N(1) + \delta \frac{\partial u_N}{\partial x}(1) = b_2. \qquad \text{(c)}$$

In practice we express (1.3) as a linear system of equations for the grid point values $u_N(x_j)$. In fact $u_N(x)$, as an Nth order polynomial, is identical with its Nth order interpolation polynomial, thus

$$u_N(x) = \sum_{k=0}^{N} u_N(x_k) g_k(x) \tag{1.4}$$

where $g_k(x)$ are the Lagrange interpolation polynomials

$$g_k(x) = -\frac{(1-x^2)P_N'(x)}{x-x_k} \cdot \frac{1}{P_N(x_k)N(N+1)} \tag{1.5}$$

such that $g_k(x_j) = \delta_{jk}$.

Substituting (1.4) into (1.3) one gets

$$\sum_{k=0}^{N} g_k''(x_j)u_N(x_k) = f(x_j) \qquad j = 1,\cdots,N-1 \qquad \text{(a)}$$

$$\sum_{k=0}^{N} \{\alpha\delta_{k0} + \beta g_k'(x_0)\}u_N(x_k) = b_1 \qquad \text{(b)} \qquad (1.6)$$

$$\sum_{k=0}^{N} \{\gamma\delta_{Nk} + \delta g_k'(x_N)\}u_N(x_k) = b_2. \qquad \text{(c)}$$

The system of linear equations (1.6) is the PSL discretization of (1.1)(1.2).

It should be noted that all other pseudospectral methods can be expressed in the same way, the only difference being the definition of the collocation points and consequently the Lagrange interpolant $g_k(x)$. For example, in the pseudospectral Chebyshev method, x_j are the extrema of

$T_N(x) = \cos(N\cos^{-1}x)$ and $g_k(x)$ is given by

$$g_k(x) = (1 - x^2)\,\frac{T_N'(x)}{x-x_k} \cdot \frac{1}{N^2 c_k} \qquad (1.5a)$$

$c_0 = c_N = \alpha \quad c_j = 1 \quad 0 \neq j \neq N.$

For the PSL method it has been shown [4] that

$$\begin{aligned} g_h''(x_j) &= -2\,\frac{P_N(x_j)}{P_N(x_k)}\,\frac{1}{(x_j-x_k)^2} \qquad j = 1,\cdots,N-1,\ j\neq k \\ g_k''(x_k) &= -\frac{1}{3}\cdot\frac{N}{1-x_k^2}\,. \end{aligned} \qquad (1.7)$$

Thus the first important feature of pseudospectral methods follows the observation that the system in (1.6) is <u>full</u>. This is true for all spectral methods; the fact that (1.4) is a global interpolation results in full matrices approximating derivatives. This is in contrast to local methods like finite differences in which the matrices are banded.

The other important feature of pseudospectral methods is a consequence of the fact that the collocation points x_j are the nodes of the Gauss-Lobatto quadrature formula. It is known that the Gauss-Lobatto quadrature formula is **exact** for every polynomial $h(x)$ of degree at most $2N-1$. Thus there exists positive weights $w_0,\cdots,w_N$ such that

$$\int_{-1}^{1} h(x)\,dx = \sum_{j=0}^{N} h(x_j)w_j, \qquad (1.8)$$

i.e., the spectral sum is exactly the integral.

In particular if f_N is the interpolant of f at the points x_j

$$\int_{-1}^{1} f_N(x)dx = \sum_{j=0}^{N} f(x_j)w_j . \tag{1.9}$$

The significance of the above mentioned properties of spectral methods is one of the main themes of this work. The fact that pseudospectral discretizations result in full matrices creates enormous difficulties in the application of spectral methods. We will attempt to show that one can use (1.8) in order to modify this difficulty, in certain circumstances, with the use of parallel architecture.

SECTION II

Multidomain spectral techniques have been extensively used by practitioners since the beginning of this decade. In many applications these methods were proved to be superior to one domain spectral techniques allowing more flexibility in the geometry of the problem as well as easier inversion of the matrices appearing in spectral methods.

We refer the reader to the review article [1] for a detailed discussion on the merits of multidomain spectral techniques. Here we would like to analyze these methods from the point of view of parallel computing; the criterion that we suggest is whether a multidomain technique can be cast as a set of disjoint problems, such that each problem can be solved in a different processor concurrently.

We start by analyzing a variety of two domain PSL methods applied to equation (1.3). The domain [-1,1] is being divided into two domains [-1,0] and [0,1] so that we have to discretize a system of two equations

$$\frac{\partial^2 u^I}{\partial x^2} = f^I(x) \qquad -1 \leq x \leq 0 \qquad (a)$$
$$\frac{\partial^2 u^{II}}{\partial x^2} = f^{II}(x) \qquad 0 \leq x \leq 1 \qquad (b) \tag{2.1}$$

with the boundary conditions

$$\alpha u^I(-1) + \beta \frac{\partial u^I}{\partial x}(-1) = b_1 \qquad (a)$$
$$\gamma\, u^{II}(1) + \delta \frac{\partial u^{II}}{\partial x}(1) = b_2. \qquad (b) \tag{2.2}$$

The PSL method is now applied separately to (2.1a), (2.2a), (2.1b), and (2.2b). It is clear however that two more conditions should be given in order to make (2.1) and (2.2) consistent with (1.3). The first condition is the same for _all_ multidomain techniques, this is the continuity condition

$$u^I(0) = u^{II}(0). \tag{2.3}$$

It is the choice of the second condition that distinguishes between the multitude of multidomain techniques. The simplest possibility is to impose the condition for the continuity of derivatives at the point $x = 0$

$$\frac{\partial u^I}{\partial x}(0) = \frac{\partial u^{II}}{\partial x}(0). \tag{2.4}$$

It is interesting to note that for $\beta = \delta = 0$ in (2.2) one can diagonalize the system (2.1) with the conditions (2.2), (2.3), (2.4) and thus present (2.1) as a system of two equations that are completely decoupled. In fact upon defining

$$R(x) = \frac{u^{I}(x) + u^{II}(-x)}{2}$$

and

$$S(x) = \frac{u^{I}(x) - u^{II}(-x)}{2}$$

one gets the following system

$$\left.\begin{aligned} \frac{\partial^2 R}{\partial x^2} &= \frac{1}{2}\,[f^{I}(x) + f^{II}(-x)] \\ \frac{\partial^2 S}{\partial x^2} &= \frac{1}{2}\,[f^{I}(x) - f^{II}(-x)] \end{aligned}\right\} \quad -1 \leq x \leq 0 \qquad (a) \qquad (2.5)$$

$$\begin{aligned} R(-1) &= \frac{1}{2}\,[\frac{b_1}{\alpha} + \frac{b_2}{\gamma}] \qquad & \frac{\partial R}{\partial x}(0) &= 0 \\ & & & \qquad\qquad (b) \\ S(-1) &= \frac{1}{2}\,[\frac{b_1}{\alpha} - \frac{b_2}{\gamma}] \qquad & S(0) &= 0. \end{aligned}$$

This is in fact the same way that was used in [5] to show that the eigenvalues of the two-domain second derivative matrices are real negative and distinct. We believe that the ability to diagonalize the system of equations resulting from a particular multidomain technique is an indicator for parallelizing the solution of the equations resulting from the discretization of the original system. A more detailed discussion of this connection will be presented in a future paper.

We turn now to the PSL method for the discretization of (2.1)-(2.4). Let the collocation points for the first domain x_j^{I}, $j = 1,\cdots,N-1$ be defined by

$$\frac{\partial P_N}{\partial x}\,(2\,x_j^I + 1) = 0 \qquad \text{(a)}$$

note that $-1 \leq x_j^I \leq 0$, and let x_j^{II} be defined by

$$\frac{\partial p_N}{\partial x}\,(2x_j^{II} - 1) = 0 \qquad \text{(b)}$$

$x_0^I = -1$, $x_N^I = 0$, $x_0^{II} = 0$, and $x_N^{II} = 1$. We assume here for simplicity that the number of points at each domain is the same but of course this is not essential.

Denote by $(u_0^I,\cdots,u_N^I)$ the values of u_N^I at the collocation points x_j^I and denote by $(u_0^{II},\cdots,u_N^{II})$ the values of u_N^{II}. Equation (2.3) means that

$$u_N^I = u_0^{II}$$

and we denote this by w.

Equation (2.4) leads to

$$\sum g_k'(x_N^I)u_k^I = \sum g_k'(x_0^{II})u_k^{II}. \qquad (2.6)$$

Applying now the method described in (1.6) to discretize (2.1) and using (2.2) we conclude the PSL discretization for the two domain method can be described as follows

$$\begin{pmatrix} A_{11} & h_{12} & 0 \\ h_{21}^T & a & h_{23}^T \\ 0 & h_{32} & A_{22} \end{pmatrix} \begin{pmatrix} u_0^I \\ \vdots \\ u_{N-1}^I \\ w \\ u_1^{II} \\ \vdots \\ u_N^{II} \end{pmatrix} = \begin{pmatrix} b_1 \\ f_1^I \\ \vdots \\ f_{N-1}^I \\ 0 \\ f_1^{II} \\ \vdots \\ b_2^{II} \end{pmatrix} \tag{2.7}$$

where A_{11}, A_{22} are $N \times N$ matrices, h_{ij} are N-component vectors and a is a scalar [6]. For convenience we define

$$u^I = \begin{pmatrix} u_0^I \\ \vdots \\ u_{N-1}^I \end{pmatrix} \quad u^{II} = \begin{pmatrix} u^{II} \\ \vdots \\ u_N^{II} \end{pmatrix}, \quad f^I = \begin{pmatrix} b_1 \\ \vdots \\ f_{N-1} \end{pmatrix}, \quad f^{II} = \begin{pmatrix} f^{II} \\ \vdots \\ b_2 \end{pmatrix}.$$

As an example, we write down the case $N = 4$. The x denotes a nonzero element

$$\left(\begin{array}{cccc|c|cccc}
X & X & X & X & X & 0 & 0 & 0 & 0 \\
X & X & X & X & X & 0 & 0 & 0 & 0 \\
X & X & X & X & X & 0 & 0 & 0 & 0 \\
X & X & X & X & X & 0 & 0 & 0 & 0 \\
\hline
X & X & X & X & X & X & X & X & X \\
\hline
0 & 0 & 0 & 0 & X & X & X & X & X \\
0 & 0 & 0 & 0 & X & X & X & X & X \\
0 & 0 & 0 & 0 & X & X & X & X & X \\
0 & 0 & 0 & 0 & X & X & X & X & X
\end{array}\right)
\begin{bmatrix} u_0^I \\ u_1^I \\ u_2^I \\ u_3^I \\ w \\ u_1^{II} \\ u_2^{II} \\ u_3^{II} \\ u_4^{II} \end{bmatrix}
=
\begin{bmatrix} b_1 \\ f_1^I \\ f_2^I \\ f_3^I \\ 0 \\ f_1^{II} \\ f_2^{II} \\ f_3^{II} \\ b_2 \end{bmatrix} \qquad (2.8)$$

From the structure (2.7) and (2.8), it can be seen how the solution of (2.7) can be carried out in two processors in parallel. In the first processor we use the Gaussian elimination for the matrix

$$\begin{pmatrix} A_{11} & h_{12} \\ h_{21}^T & \frac{a}{2} \end{pmatrix}$$

to get to an upper triangular matrix; concurrently the matrix

$$\begin{pmatrix} \frac{a}{2} & h_{23}^T \\ h_3^2 & A_{22} \end{pmatrix}$$

is being transformed to a lower triangular form in the second processor. The system (2.8) is thus being transformed to the form

$$\begin{pmatrix} U & \tilde{h}_{12} & 0 \\ 0 & b & 0 \\ 0 & \tilde{h}_{32} & L \end{pmatrix} \begin{pmatrix} u^{I} \\ w \\ u^{II} \end{pmatrix} = \begin{pmatrix} \tilde{f}^{I} \\ c \\ \tilde{f}^{II} \end{pmatrix} . \qquad (2.9)$$

U is an upper triangular N×N matrix and L is lower triangular N×N matrix. The value of w is first determined and backward/forward substitution is then carried out, again concurrently within the two processors.

The procedure described above is strongly related to the fact that the system (2.1)-(2.4) can be diagonalized, however it is more general since even if (2.1)-(2.4) cannot be diagonalized the parallel inversion of (2.8) can still be carried out. For example if β or δ are different from zero in (2.2) or if (2.1) is of the form $(a(x)u_x)_x = f$, diagonalization is impossible but the form (2.7) is still the same and the parallel solution of (2.8) is the same.

A different approach to the solution of (2.7) is to rewrite the system explicitly

$$A_{11}u^{I} + h_{12}w = f^{I} \qquad \text{(a)}$$

$$A_{22}u^{II} + h_{32}w = f^{II} \qquad \text{(b)} \qquad (2.10)$$

$$h_{21}^{T} \cdot u^{I} + aw + h_{23}^{T}u^{II} = 0. \qquad \text{(c)}$$

Thus

$$u^{I} = A_{11}^{-1}f^{I} - w\, A_{11}^{-1}h_{12} \qquad \text{(a)}$$

$$u^{II} = A_{22}^{-1} f^{II} - w\, A_{22}^{-1} h_{32} \qquad (b) \qquad (2.11)$$

$$w = - \frac{h_{21}^T A_{11}^{-1} f^I + h_{23}^T A_{22}^{-1} f^{II}}{a - h_{21}^T A_{11}^{-1} h_{12} - h_{23}^T A_{22}^{-1} h_{32}} \,. \qquad (c)$$

The system (2.11) seems to be, at first glance, more complicated than it really is. In fact, let p_1, p_2, p_3, p_4 be N-component vectors such that

$$A_{11} p_1 = f^I \qquad (a)$$

$$A_{11} p_2 = h_{12} \qquad (b)$$

$$A_{22} p_3 = f^{II} \qquad (c) \qquad (2.12)$$

$$A_{22} p_4 = h_{32}. \qquad (d)$$

Then equation (2.11) can be written as

$$w = - \frac{h_{21}^T p_1 + h_{23}^T T_3}{a - h_{21}^T p_2 - h_{21}^T p_4}$$

$$u^I = p_1 - w p_2 \qquad (2.13)$$

$$u^{II} = p_3 - w p_4.$$

It is self evident now that a part of the solution procedure can be carried out in parallel. The vectors $p_1, \cdots, p_4$ can be computed in parallel in four (or two) different processors. The inner products $h_{21}^T p_1$, $h_{23}^T p_3$, $h_{21}^T p_2$, and $h_{21}^T p_3$ can be again done in parallel. Equation (2.13) is basically gathering information from the processors to the shared memory. It seems that the procedure outlined in (2.9) is more efficient than (2.12)-(2.13); however (2.12)-(2.13) can be easily generalized to more than two domains.

The procedure outlined in (2.12)-(2.13) has a very interesting interpretation on the differential equation level. In fact equation (2.12a) is a discretized version of the equation

$$\frac{\partial^2 q_1}{\partial x^2} = f \qquad -1 \leq x \leq 0 \tag{2.14}$$

with

$$\alpha\, q_1(-1) + \beta \frac{\partial q_1}{\partial x}(-1) = b_1 \qquad q_1(0) = 0$$

whereas (2.12b) is the PSL discretization of

$$\begin{gathered}\frac{\partial^2 q_2}{\partial x^2} = 0 \qquad -1 \leq x \leq 0 \\ \alpha\, q_2(-1) + \beta \frac{\partial q_2}{\partial x}(-1) = 0 \qquad q_2(0) = -1.\end{gathered} \tag{2.15}$$

Similarily (2.12c) and (2.12d) are the discretization of

$$\begin{gathered}\frac{\partial^2 q_3}{\partial x^2} = f \\ q_3(0) = 0 \qquad \gamma q_3(1) + \delta \frac{\partial q_3}{\partial x}(1) = b_2\end{gathered} \tag{2.16}$$

and

$$\frac{\partial^2 q_4}{\partial x^2} = 0 \tag{2.17}$$

$$q_4(0) = -1 \qquad \gamma q_4(1) + \delta \frac{\partial q_4}{\partial x}(1) = 0.$$

It is well-known that the general solution to (2.1a) and (2.2a) is

$$u^I(x) = q_1 - wq_2. \tag{2.18}$$

The continuity condition

$$u^I(0) = u^{II}(0)$$

implies that

$$u^{II}(x) = q_3 - wq_4 \tag{2.19}$$

and (2.13a) is the manifestation of the condition (2.4).

The advantage of the formulation (2.14)-(2.19) is that it lends itself easily to generalization without getting into the details of the discretization.

We conclude by summarizing the main results in this section:

1. We showed that for Dirichlet boundary conditions, the system (2.1) and (2.2) can be diagonalized and argued that each one of the equations can be solved in a different processor.

2. We presented a general technique to solve concurrently the system (2.7) by assigning each separate domain to a separate processor [e.g., (2.9)].

3. We presented a different procedure (related to the influence matrix method) and gave an interpretation on the level of the differential equations.

In Section IV, we will discuss methods 2 and 3 in the general case of more than two domains. In the next section, we generalize method 1 to a two dimensional equation.

SECTION III

The extension of the techniques mentioned in Section II to the two dimensional Poisson equation seems to be obvious. However, there are some differences in implementation. The two dimensional extension of (1.1) is the the equation

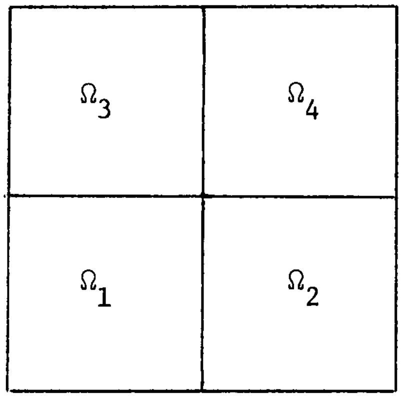
$$u_{xx} + u_{yy} = f(x,y) \qquad \begin{array}{c} -1 \leq x \leq 1 \\ -1 \leq y \leq 1 \end{array} \quad . \tag{3.1}$$

Here we will discuss only Dirichlet boundary conditions of the form

$$\begin{array}{ll} u(x,-1) = b_1(x) & u(-1,y) = b_3(y) \\ u(x,1) = b_2(x) & u(1,y) = b_4(y). \end{array} \tag{3.2}$$

The original domain is divided now into four domains

$$\begin{array}{ll} \Omega_1 = \{-1 \leq x \leq 0,\ -1 \leq y \leq 0\} & \Omega_2 = \{0 \leq x \leq 1,\ -1 \leq y \leq 0\} \\ \Omega_3 = \{-1 \leq x \leq 0,\ 0 \leq y \leq 1\} & \Omega_4 = \{0 \leq x \leq 1,\ 0 \leq y \leq 1\}. \end{array} \tag{3.3}$$



It is readily verified that (3.1)(3.2) can be factored into <u>four</u> independent problems if one imposes continuity of the function and normal derivative at interfaces.

In fact denote by u^I, u^{II}, u^{III}, u^{IV} the solution $u(x,y)$ in $\Omega_1, \Omega_2, \Omega_3, \Omega_4$. The interface conditions amount to

$$\begin{aligned} &u^I(0,y) = u^{II}(0,y) \qquad -1 \le y \le 0; \quad u^{III}(0,y) = u^{IV}(0,y) \qquad 0 \le y \le 1 \\ &\frac{\partial u^I}{\partial y}(0,y) = \frac{\partial u^{II}}{\partial y}(0,y) \qquad\qquad \frac{\partial u^{IV}}{\partial y}(0,y) = \frac{\partial u^{IV}}{\partial y}(0,y) \end{aligned} \tag{3.5}$$

$$\left.\begin{aligned} &u^I(x,0) = u^{III}(x,0) \\ &\frac{\partial u^I(x,0)}{\partial x} = \frac{\partial u^{III}(x,0)}{\partial x} \end{aligned}\right. \quad -1 \le x \le 0 \qquad \left.\begin{aligned} &u^{II}(x,0) = u^{IV}(x,0) \\ &\frac{\partial u^{II}(x,0)}{\partial x} = \frac{\partial u^{IV}(x,0)}{\partial x} \end{aligned}\right. \quad 0 \le x \le 1 \quad . \tag{3.6}$$

We define now

$$R_1(x,y) = u^I(x,y) + u^{II}(-x,y) + u^{III}(x,-y) + u^{IV}(-x,-y)$$

$$R_2(x,y) = u^I(x,y) - u^{II}(-x,y) + u^{III}(x,-y) - u^{IV}(-x,y)$$

$$R_3(x,y) = u^I(x,y) + u^{II}(-x,y) - u^{III}(x,-y) - u^{IV}(-x,-y) \tag{3.7}$$

$$R_4(x,y) = u^I(x,y) - u^{II}(-x,y) - u^{III}(x,-y) + u^{IV}(-x,-y).$$

The functions R_1, R_2, R_3, R_4 satisfy trivially the Equation (3.1) in Ω_1 defined 'in (3.4). Moreover

$$R_1(-1,y) = b_3(y) + b_4(y) + b_3(-y) + b_4(-y)$$

$$R_2(-1,y) = b_3(y) - b_4(y) + b_3(-y) - b_4(-y)$$

$$R_3(-1,y) = b_3(y) + b_4(y) - b_3(-y) - b_4(-y) \tag{3.8}$$

$$R_4(-1,y) = b_3(y) - b_4(y) - b_3(-y) + b_4(-y).$$

In the same way we can express each $R_i(x,-1)$ $i = 1,\cdots,4$ as combinations of $b_1(\pm x)$ and $b_2(\pm x)$. Moreover, e.g., (3.5) yields

$$R_2(0,y) = R_4(0,y) = 0$$

$$\frac{\partial R_1}{\partial y}(0,y) = \frac{\partial R_3}{\partial y}(0,y) = 0 \tag{3.9}$$

and, e.g., (3.6) yields

$$\begin{gathered} R_3(x,0) = R_4(x,0) = 0 \\ \frac{\partial R_2}{\partial x}(x,0) = \frac{\partial R_1}{\partial x}(x,0) = 0. \end{gathered} \tag{3.10}$$

Thus, $R_1(x,y)$, $R_2(x,y)$, $R_3(x,y)$, $R_4(x,y)$ can be solved concurrently.

To get back $u^I,\cdots,u^{II}$ one concludes from (3.7)

$$u^I(x,y) = \frac{1}{4}\{R_1(x,y) + R_2(x,y) + R_3(x,y) + R_4(x,y)\} \qquad \begin{matrix} -1 \leq x \leq 0 \\ -1 \leq y \leq 0 \end{matrix}$$

$$u^{II}(x,y) = \frac{1}{4}\{R_1(-x,y) - R_2(-x,y) + R_3(-x,y) - R_4(-x,y)\} \qquad \begin{matrix} 0 \leq x \leq 1 \\ -1 \leq y \leq 0 \end{matrix}$$

$$u^{III}(x,y) = \frac{1}{4}\{R_1(x,-y) + R_2(x,-y) - R_3(x,-y) - R_4(x,-y)\} \qquad \begin{matrix} -1 \leq x \leq 0 \\ 0 \leq y \leq 1 \end{matrix} \tag{3.11}$$

$$u^{IV}(x,y) = \frac{1}{4}\{R_1(-x,-y) - R_2(-x,-y) - R_3(-x,-y) + R_4(-x,-y)\} \qquad \begin{matrix} 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \end{matrix} .$$

Equation (3.7) shows a way to fully parallelize the solution of (3.1) using four different processors. This is an extension of (2.5) which we adopted as an indicator for possible parallelism. Investigations into the choice of interface conditions leading to multidomain techniques that can easily be parallelized will be reported in a future paper.

SECTION IV

We return to Equation (1.1) to discuss multidomain techniques with more than two domains. We will discuss also a modification of (1.1) in the form

$$(h(u,u_x))_x = f(x) \qquad -1 \leq x \leq 1 \tag{4.1}$$

$$h(u,u_x) = b_1 \quad \text{at} \quad x = -1$$

$$u = b_2 \qquad \text{at} \quad x = 1.$$

Actually this is the form that appears most often in applications.

Nothing essential is lost if we restrict ourselves to three domains and for simplicity we will discuss this case only. Generalization to more domains are trivial.

We start by showing that our indicator for parallelism shows that imposing continuity of u and u_x as an interface condition for (1.1) (or continuity of u and $h(u,u_x)$ in the case of (4.1)) leads to a method that cannot be factorized into disjoint problems. Let u^I, u^{II}, u^{III} be the solutions of

$$u^I_{xx} = f^I \qquad -1 \leq x \leq \xi_1$$

$$u^{II}_{xx} = f^{II} \qquad \xi_1 \leq x \leq \xi_2 \tag{4.2}$$

$$u^{III}_{xx} = f^{III} \qquad \xi_2 \leq x \leq 1$$

with $u^I(-1) = b_1$, $u^{III}(1) = b_2$ and the interface condition

$$\begin{aligned} u^{I}(\xi_1) &= u^{II}(\xi_1) & u^{II}(\xi_2) &= u^{III}(\xi_2) \\ \frac{\partial u^{I}}{\partial x}(\xi_1) &= \frac{\partial u^{II}}{\partial x}(\xi_1) & \frac{\partial u^{II}}{\partial x}(\xi_2) &= \frac{\partial u^{III}}{\partial x}(\xi_2). \end{aligned} \tag{4.3}$$

As in (2.5) we map

$$[-1,\xi_1] \rightarrow [\xi_2,\xi_1] \quad \text{and} \quad [\xi_2,1] \rightarrow [\xi_2,\xi_1]$$

and denote the new dependent variables $\tilde{u}^{I}$, $\tilde{u}^{II}$, $\tilde{u}^{III}$. We have now

$$\begin{aligned} \tilde{u}^{I}_{xx} &= \tilde{f}^{I} \\ \tilde{u}^{II}_{xx} &= \tilde{f}^{II} \qquad\qquad x \in [\xi_1,\xi_2] \\ \tilde{u}^{III}_{xx} &= \tilde{f}^{III} \end{aligned} \tag{4.4}$$

with

$$\tilde{u}^{I}(\xi_2) = b_1 \qquad \tilde{u}^{III}(\xi_1) = b_2 \tag{4.5}$$

$$\tilde{u}^{I}(\xi_1) = \tilde{u}^{II}(\xi_1) \qquad \tilde{u}^{II}(\xi_2) = u^{III}(\xi_2)$$

$$\frac{\partial \tilde{u}^{I}(\xi_1)}{\partial x} = - \frac{\partial \tilde{u}^{II}(\xi_1)}{\partial x} \qquad \frac{\partial u^{II}(\xi_2)}{\partial x} = - \frac{\partial u^{III}(\xi_2)}{\partial x} .$$

We seek now a combination of the form

$$R = A\tilde{u}^{I} + B\tilde{u}^{III} + C\tilde{u}^{III} \tag{4.6}$$

that will satisfy the differential equation at $\xi_1 \leq x \leq \xi_2$ with boundary conditions that are independent of $\tilde{u}^{I}, \tilde{u}^{II}, \tilde{u}^{III}$. In fact we need three different values of A, B, C to decouple the problem completely. However, it is easily verified that there is only one quantity that is decoupled. Namely

$$R = -\tilde{u}^{I} + \tilde{u}^{II} - \tilde{u}^{III} \tag{4.7}$$

is the only quantity that can be computed separately from the other two.

This conclusion becomes more apparent when we observe the structure of the PSL method for three domains. In analogy to (2.7) we get the system

$$\begin{bmatrix} A_{11} & h_{12} & & & \\ h_{21}^{T} & a_1 & h_{23}^{T} & a_4 & \\ & h_{31} & A_{22} & h_{32} & \\ & a_3 & h_{41}^{T} & a_2 & h_{42}^{T} \\ & & & h_{51} & A_{33} \end{bmatrix} \begin{bmatrix} u^{I} \\ w_1 \\ u^{II} \\ w_2 \\ u^{III} \end{bmatrix} = \begin{bmatrix} f^{I} \\ 0 \\ f^{II} \\ 0 \\ f^{III} \end{bmatrix} \tag{4.8}$$

where A_{ii} are $N\times N$ matrices h_{ij} are N component vectors and a_i are scalars. The unknowns are

$$u^{I} = (u_0^{I},\cdots,u_{N-1}^{I}),\ u^{II} = u_1^{II},\cdots,u_N^{II}),\ u^{III} = (u_1^{III},\cdots,u_N^{III})$$

w_1 stands for $u_N^{I} = u_0^{II}$ and w_2 stands for $u_{N+1}^{II} = u_0^{III}$.

We write down explicitly the case $N = 3$, the x denotes nonzero elements

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & X & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & X & X & X \end{bmatrix} \begin{bmatrix} u_0^I \\ u_1^I \\ u_2^I \\ w_1 \\ u_1^{II} \\ u_2^{II} \\ u_3^{II} \\ w_2 \\ u_1^{III} \\ u_2^{III} \\ u_3^{III} \end{bmatrix} = \begin{bmatrix} b_1 \\ f_1^I \\ f_2^I \\ 0 \\ f_1^{II} \\ f_2^{II} \\ f_3^{II} \\ 0 \\ f_1^{III} \\ f_2^{III} \\ b_2 \end{bmatrix} \tag{4.9}$$

It is clear from (4.5) that we can carry out Gaussian elimination in parallel only for two blocks, the first is $(2N+2)\times(2N+2)$ and the second $(N+1)\times(N+1)$, so at this point nothing has been gained from using three domains. This corresponds to the observation that there is only one quantity that is independent of the others. It seems that the method outlined in (2.9) does not carry over to more than two domains. We will show later that a judicious choice of the interface boundary condition can lead to a separable set of equations.

The method outlined in (2.10)-(2.12) is still valid. We first define p^I, p^{II}, p^{III}, p_{21}, p_{22}, p_{32}, p_{31} to be the solution vectors of

$$\begin{array}{llll} A_{11}P^{I} = f^{I} & A_{11}P_{12} = h_{12} & & \text{(a)} \\ A_{22}P^{II} = f^{II} & A_{22}P_{21} = h_{31} & A_{22}P_{22} = h_{32} & \text{(b)} \\ A_{33}P^{III} = f^{III} & A_{33}P_{31} = h_{51}. & & \text{(c)} \end{array} \qquad (4.10)$$

It is readily verified that if w_1, w_2 are the solution of the system

$$\begin{aligned} w_1[a_1 - h_{21}^T P_{12} - h_{23}^T P_{21}] + w_2[a_4 - h_{23}^T P_{22}] &= -h_{21}^T P^{I} - h_{23}^T P^{II} \\ w_1[a_3 - h_{41}^T P_{21}] + w_2[a_2 - h_{41}^T P_{22} - h_{42}^T P_{31}] &= -h_{41}^T P^{II} - h_{41}^T P^{III} \end{aligned} \qquad (4.11)$$

then

$$\begin{aligned} u^{I} &= P^{I} - w_1 P_{12} \\ u^{II} &= P^{II} - w_1 P_{21} - w_2 P_{22} \\ u^{III} &= P^{III} - w_2 P_{31}. \end{aligned} \qquad (4.12)$$

The procedure (4.10)-(4.12) shows that one has to solve three equations per inner domain, c.f. (4.10b), and two equations per boundary domain; (4.10a,c) this can be done in parallel. Equation (4.11) has then to be solved. The dimension of this system is proportional to the number of domains.

The method described in (4.10)-(4.12) has, again, an interpretation on the differential equation level. The vectors P^{I}, P^{II}, P^{III} are the approximations to the Equation (1.1) (or (4.17) with homogeneous boundary conditions in each domain). The vectors P_{ij} are the approximations to the solutions of the homogeneous problem with homogeneous boundary conditions at one side and value -1 at the other side. Equation (4.11) expresses the interface conditions.

We claim now that by a better choice of the interface condition one can greatly simplify the solution procedure. In fact integrating (4.1) from -1 to ξ_1 one gets

$$h(u,u_x)(x = \xi_1) - h(u,u_x)(x = -1) = \int_{-1}^{\xi_1} f dx. \tag{4.13}$$

Therefore,

$$h(u,u_x)(x = \xi_1) = b_1 + \int_{-1}^{\xi_1} f dx. \tag{4.14}$$

We note the exact result, that within the context of the PSL method

$$\int_{1}^{\xi_1} (P_N f) dx = \sum_{j=0}^{N} f(x_j) w_j$$

and therefore the integral is known explicitly. We therefore replace (4.2) and (4.3) by

$$\begin{aligned} h(u^{I},u_x^{I})_x &= f^{I} & -1 \leq x \leq \xi_1 \\ h(u^{II},u_x^{II})_x &= f^{II} & \xi_1 \leq x \leq \xi_2 \\ h(u^{III},u_x^{III})_x &= f^{III} & \xi_2 \leq x \leq 1 \end{aligned} \tag{4.15}$$

with interface conditions given as

$$u^{I}(\xi_1) = u^{II}(\xi_1) \qquad\qquad u^{II}(\xi_2) = u^{III}(\xi_2)$$

$$h(u^{II},u_x^{II})(x = \xi_1) = b_1 + \sum f^{I}(x_j)w_j \qquad h(u^{III},u_x^{III})\times(x = \xi_2) = b_1 + \sum f^{I}(x_j)w_j + \sum f^{II}(x_j)w_j. \tag{4.16}$$

When (4.15) and (4.16) are used, we get basically the system described in (4.8) with $h_{21}^{T} = 0$, $a_3 = 0$, and $h_{41}^{T} = 0$.

With this notation we note first that the system (4.11) is bidiagonal. Alternatively, one can transform A_{11} and A_{22} to an upper diagonal form and the lower corner structure in (4.8), i.e.,

$$\begin{bmatrix} a_2 & h_{42}^{T} \\ h_{51} & A_{33} \end{bmatrix}$$

to a lower diagonal form and then solve for w_2 and w_1. Backward substitution will then be done concurrently.

We conclude this section by emphasizing that interface boundary conditions have a great impact on the ability to carry out the solution procedure in parallel.

ACKNOWLEDGEMENT

The authors are indebted to Professor Saul Abarbanel for his help in proving the result in Section III.

REFERENCES

1. A. Quarteroni: "Domain decomposition algorithms using spectral methods," to appear in Calcolo.

2. A. T. Patera: "A spectral element method for fluid dynamics," J. Comp. Physics, Vol. 54, 1984, pp. 468-488.

3. R. C. Y. Chin: "Asymptotics and domain decomposition in parallel computing," presented at the Third SIAM Conf. on Parallel Processing, Los Angeles, December 1987.

4. D. Gottlieb, M. Y. Hussaini, and S. A. Orszag: "Theory and application of spectral methods," in Spectral Methods for Partial Differential Equations, SIAM, 1984.

5. D. Gottlieb and L. Lustman: "The spectrum of the Chebyshev collocation operator for the heat equation," SIAM J. Num. Anal., Vol. 20, 1983, pp. 909-921.

6. H. C. Ku, R. S. Hirsh, and T. D. Taylor: "A C(0) pseudospectral element method for the solution of the Navier-Stokes equations in primitive variables," presented at the First ICIAM Conference, Paris, June 1987.

National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-181629 ICASE Report No. 88-15	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

PARALLEL PSEUDOSPECTRAL DOMAIN DECOMPOSITION TECHNIQUES

5. Report Date

February 1988

6. Performing Organization Code

7. Author(s)

David Gottlieb and Richard S. Hirsh

8. Performing Organization Report No.

88-15

10. Work Unit No.

505-90-21-01

9. Performing Organization Name and Address

Institute for Computer Applications in Science
and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23665-5225

11. Contract or Grant No.

NAS1-18107

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

13. Type of Report and Period Covered

Contractor Report

14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Submitted to SISC

Final Report

16. Abstract

The influence of interface boundary conditions on the ability to parallelize pseudospectral multidomain algorithms is investigated. Using the properties of spectral expansions, a novel parallel two domain procedure is generalized to an arbitrary number of domains each of which can be solved on a separate processor. This interface boundary condition considerably simplifies influence matrix techniques.

17. Key Words (Suggested by Author(s))

interface boundary conditions, parallel computing, pseudospectral methods

18. Distribution Statement

59 - Mathematical and Computer Sciences (General)
64 - Numerical Analysis

Unclassified - unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 28	22. Price A03

NASA-Langley, 1988