Domain Decomposition for Two-Point Boundary Value Problems

Herbert B. Keller

CRPC-TR90026
August 1990

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

Domain Decomposition for Two-Point Boundary Value Problems*

ABSTRACT

The rapid development of concurrent multicomputers calls for the corresponding development of numerical methods that can most effectively be implemented on these new machines. Domain decomposition is the general term used to describe methods for solving boundary value problems which allow concurrent computations to occur on different subdomains. The major emphasis in such methods has been on iterative procedures by means of which the correct boundary data on the subdomains can be determined. We show how to devise effective domain decomposition methods that are not iterative, for two-point boundary value problems. As could be anticipated, these methods are related to the original parallel shooting procedures introduced years ago.

1. Introduction

Our motivation for studying domain decomposition methods is the rapid development of concurrent multicomputers (*i.e.* parallel processing). Indeed, this work is an extension of earlier studies in this area [1,2] in which we introduced parallel shooting (also termed multiple shooting). We stress in this paper the analytical aspects of more general domain decomposition procedures and we do not employ iterative methods for solving either the decomposed problem or its various numerical approximations. One reason for this is that direct methods are more suitable for a variety of auxiliary considerations (*i.e.* stability, continuation, bifurcation, etc.) that may be relevant to the basic problems of interest.

The problems we consider are linear two-point boundary value problems for first order systems of ordinary differential equations. Such systems arise naturally in many

* This work was supported in part by the NSF under Cooperative Agreement No. CCR-8809615.

applications. Also, the method of lines reduces the approximate solution of various P.D.E.-problems to these systems. Finally, we recall that Newton's method applied to solve nonlinear problems yields linear systems for each iteration. Thus, in general, our studies can be relevant to the solution of a variety of nonlinear partial differential equations problems.

The methods we shall employ rely only on the basic existence and uniqueness theory for linear ordinary differential equations. As it turns out, essentially the same existence and uniqueness theory is valid for systems of difference equations. Thus, our reduction procedures for systems of ordinary differential equations automatically yield corresponding decompositions for various systems of linear algebraic equations. In particular, these analogous treatments are clear for linear algebraic systems whose coefficient matrices are in block bi-diagonal or block tri-diagonal form.

2. First Order Systems

We consider the general linear two-point boundary value problem:

$$\text{(1)} \qquad \mathbf{L}\mathbf{y}(t) \equiv \frac{d\mathbf{y}}{dt} + A(t)\mathbf{y} = \mathbf{f}(t)\,,\ a < t < b\,;$$

$$\text{(2)} \qquad B\mathbf{y}(t) \equiv L\mathbf{y}(a) + R\mathbf{y}(b) = \boldsymbol{\beta}\,.$$

Here $\mathbf{y}(t), \mathbf{f}(t), \boldsymbol{\beta} \in \mathbb{R}^n$ and $A(t), L, R \in \mathbb{R}^{n\times n}$. The special case of separated boundary conditions which frequently occurs is included by taking

$$L \equiv \begin{pmatrix} \hat{L} \\ 0 \end{pmatrix} \begin{matrix} \}\, p \\ \}\, q \end{matrix}\,,\ R \equiv \begin{pmatrix} 0 \\ \hat{R} \end{pmatrix} \begin{matrix} \}\, p \\ \}\, q \end{matrix}\,,\ \boldsymbol{\beta} \equiv \begin{pmatrix} \boldsymbol{\beta}_a \\ \boldsymbol{\beta}_b \end{pmatrix}\,,\ p + q = n\,.$$

Then (2) is simply:

$$\text{(3)} \qquad \hat{L}\mathbf{y}(a) = \boldsymbol{\beta}_a \quad , \quad \hat{R}\mathbf{y}(b) = \boldsymbol{\beta}_b\,.$$

The existence theory for (1),(2) is well known [2]. In terms of the fundamental solution matrix $Y(t,s)$, defined as the solution of:

$$\mathbf{L}Y(t,s) = 0\ ,\ a < t < b\ ;\ Y(s,s) = I\ ; \tag{4}$$

problem (1),(2) has a unique solution for each $\mathbf{f}(t)\varepsilon C[a,b]$ and $\boldsymbol{\beta}$ iff the $n \times n$ matrix

$$Q \equiv LY(a,s) + RY(b,s) \tag{5}$$

is NONSINGULAR. When Q is nonsingular, the solution of (1),(2) is given by:

$$\begin{aligned} \mathbf{y}(t) &= Y(t,s)\boldsymbol{\xi} + \int_s^t Y(t,s)\mathbf{f}(s)ds \\ &\equiv Y(t,s)\boldsymbol{\xi} + \mathbf{y}_p(t)\ . \end{aligned} \tag{6a}$$

Here $\boldsymbol{\xi}$ is the unique solution of

$$Q\boldsymbol{\xi} = \boldsymbol{\beta} - B\mathbf{y}_p(t)\ . \tag{6b}$$

3. Domain Decomposition

On the interval $I_{ab} \equiv [a,b]$ we introduce the subintervals (or subdomains) I_j as:

$$I_j \equiv [t_{j-1}, t_j]\ ,\ j = 1, 2, \cdots, J\ , \tag{7a}$$

where the endpoints satisfy

$$a = t_o < t_1 < \cdots < t_J = b\ . \tag{7b}$$

On each I_j we consider the SUBDOMAIN PROBLEMS for $\mathbf{v}_j(t)\varepsilon \mathbb{R}^n$:

$$\mathbf{L}\mathbf{v}_j(t) = \mathbf{f}(t)\ ,\ t\varepsilon I_j\ , \tag{8a}$$

$$L_j \mathbf{v}_j(t_{j-1}) + R_j \mathbf{v}(t_j) = \mathbf{0} \; ; \tag{8b}$$

and for $V_j(t) \varepsilon \mathbb{R}^{n\times n}$:

$$\mathbf{L}V_j(t) = \mathbf{0} \; , \; t\varepsilon I_j \; , \tag{9a}$$

$$L_j V_j(t_{j-1}) + R_j V_j(t_j) = M_j \; . \tag{9b}$$

Here $M_j \varepsilon \mathbb{R}^{n\times n}$ are to be NONSINGULAR and the matrices L_j and R_j are assumed to be such that

$$Q_j \equiv L_j Y(t_{j-1}, s) + R_j Y(t_j, s) \; , \; j = 1, 2, \cdots, J \tag{10}$$

are NONSINGULAR. Then each of (8) and (9) have unique solutions. Further, we have the fact that **each $V_j(t)$ is NONSINGULAR.** To see this, we use the fundamental solution $Y(t,s)$ and the fact that $V_j(t)$ satisfies (9a), perhaps on an extended interval to contain s, and note that

$$V_j(t) = Y(t,s)V_j(s) \; . \tag{11}$$

Using this representation in (9b), we have that

$$\begin{aligned} M_j &= [L_j Y(t_{j-1}, s) + R_j Y(t_j, s)]V_j(s) \\ &= Q_j V_j(s) \; . \end{aligned}$$

Since M_j and Q_j are nonsingular, the same is true of $V_j(s)$ and thus also for $V_j(t)$ for all t.

Now we seek the solution of (1),(2) in the form:

$$\mathbf{y}(t) = \mathbf{v}_j(t) + V_j(t)\eta_j \; , \; t\varepsilon I_j \; , \; j = 1, 2, \cdots, J \; . \tag{12}$$

Clearly, we need only insure the continuity of $\mathbf{y}(t)$ on I_{ab} and that it satisfies the boundary conditions (2). Continuity, $\mathbf{y}(t_j+0)=\mathbf{y}(t_j-0)$, requires of (12) that

$$V_{j+1}(t_j)\eta_{j+1} - V_j(t_j)\eta_j = \mathbf{v}_j(t_j) - \mathbf{v}_{j+1}(t_j)\ ,\ j=1,2,\cdots,J-1\ .$$

The boundary conditions require

$$LV_1(a)\eta_1 + RV_J(b)\eta_J = \beta - L\mathbf{v}_1(a) - R\mathbf{v}_J(b) \equiv \hat{\beta}\ .$$

We can write the above conditions as a linear system of nJ equations

$$\mathcal{A}\eta = \mathbf{B}\ , \tag{13a}$$

where

$$\eta \equiv \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_J \end{pmatrix}\ ,\quad \mathbf{B} \equiv \begin{pmatrix} \hat{\beta} \\ \mathbf{v}_1(t_1) - \mathbf{v}_2(t_1) \\ \vdots \\ \mathbf{v}_{J-1}(t_{J-1}) - \mathbf{v}_J(t_{J-1}) \end{pmatrix}\ , \tag{13b}$$

and

$$\mathcal{A} \equiv \begin{pmatrix} LV_1(a) & & & & RV_J(b) \\ -V_1(t_1) & V_2(t_1) & & & \\ & -V_2(t_2) & V_3(t_2) & & \\ & \ddots & & \ddots & \\ & & -V_{J-1}(t_{J-1}) & V_J(t_{J-1}) & \end{pmatrix}\ . \tag{14}$$

For separated boundary conditions, as in (3), we write the boundary conditions at $t_0=a$ first and those at $t_J=b$ last, in which case $\mathcal{A}$ becomes

$$\mathcal{A} \equiv \begin{pmatrix} \hat{L}V_1(a) & & & \\ -V_1(t_1) & V_2(t_1) & & \\ \ddots & \ddots & & \\ & V_{J-1}(t_{J-1}) & V_J(t_{J-1}) & \\ & & \hat{R}V_J(b) & \end{pmatrix} \begin{matrix} \}\ p \\ \}\ n=q+p \\ \vdots \\ \}\ n=q+p \\ \}\ q \end{matrix}\ . \tag{15a}$$

This is a special block tri-diagonal form

$$\mathcal{a} \equiv [A_j, B_j, C_j] \ , \tag{15b}$$

with

$$A_j \equiv \begin{pmatrix} \hat{A}_j \\ 0 \end{pmatrix} \begin{matrix} \} \ p \\ \} \ q \end{matrix} \quad , \quad C_j \equiv \begin{pmatrix} 0 \\ \hat{C}_j \end{pmatrix} \begin{matrix} \} \ p \\ \} \ q \end{matrix} \quad . \tag{15c}$$

To show that $\mathcal{a}$ is nonsingular when (1),(2) has a unique solution, we factor $\mathcal{a}$ in (14) as:

$$\mathcal{a} = \begin{pmatrix} P_1 & P_2 & \cdots & P_J \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix} \begin{pmatrix} V_1(a) & & & \\ -V_1(t_1) & V_2(t_1) & & \\ & \ddots & \ddots & \\ & & V_{J-1}(t_{J-1}) & V_J(t_{J-1}) \end{pmatrix} ,$$

where

$$\begin{aligned} P_J &\equiv R V_J(b) V_J^{-1}(t_{J-1}) \ ; \\ P_j &\equiv P_{j+1} V_j(t_j) V_j^{-1}(t_{j-1}) \ , \ j = J-1, \cdots, 2 \ ; \\ P_1 &\equiv L + P_2 V_1(t_1) V_1^{-1}(a) \ . \end{aligned}$$

Since all $V_j(t)$ are nonsingular, this factorization is valid. Also, the right-most matrix factor is nonsingular, as it is block lower triangular with $V_j(t_{j-1})$ as the diagonal blocks. Thus, $\mathcal{a}$ will be nonsingular iff P_1 is nonsingular. By recursion of the P_j, we find that

$$P_1 = L + R[V_J(t_J) V_J^{-1}(t_{J-1})] \cdots [V_1(t_1) V_1^{-1}(t_0)] \ .$$

However, recalling (11), it follows that

$$V_j(t_j) V_j^{-1}(t_{j-1}) = Y(t_j, s) Y^{-1}(t_{j-1}, s) \ , \ j = 1, 2, \cdots, J \ ;$$

and so P_1 becomes

$$\begin{aligned} P_1 &= L + RY(t_J, s)Y^{-1}(t_0, s) , \\ &= [LY(a,s) + RY(b,s)]Y^{-1}(a,s) , \\ &= QY^{-1}(a,s) . \end{aligned}$$

Thus, P_1 and hence $\mathcal{A}$ are nonsingular. We do **not** advocate using the above factorization to solve (13).

4. Parallel Numerical Implementation: General Case

To employ the indicated domain decomposition procedure to solve (1),(2) or (1),(3), we must first solve the subdomain problems (8),(9) for $j = 1, 2, \cdots, J$. As these problems are independent of each other, they can be solved concurrently if J processors are available. Of course, some discrete procedure will be used and so the $\mathbf{v}_j(t)$ and $V_j(t)$ are actually approximated either on some grid over I_j for finite difference methods or in some appropriate subspace for finite element, spectral or collocation procedures. We do not discuss these aspects here and indeed all of the above indicated procedures could be employed at the same time on different subdomains if MIMD machines are used. Different nonuniform grids can be employed on different subintervals, etc. However these calculations are done, they will probably form the major part of the computing effort in both time and arithmetic operations. They will supply the approximations to the elements of $\mathcal{A}$ and $\mathbf{B}$ that we use to solve (13a) for η. Then we can use this solution in (12) to approximate $\mathbf{y}(t)$ over each subdomain I_j, and again this can be done concurrently. Thus we must examine the solution of (13a), and we seek to do this in a parallel manner.

We rewrite (14) in the block notation:

$$(16) \qquad \mathcal{A} \equiv \begin{pmatrix} B_1 & & & A_1 \\ A_2 & B_2 & & \\ & \ddots & \ddots & \\ & & A_J & B_J \end{pmatrix}$$

where each A_j and B_j is $n \times n$ for all $j = 1, 2, \cdots, J$. Now the system (13a) is decomposed in the BORDERED FORM

(17) $$a \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{pmatrix} \quad , \quad \mathbf{x}, \mathbf{g} \varepsilon \mathbb{R}^{n \cdot (J-1)} \quad ; \quad \boldsymbol{\xi}, \boldsymbol{\gamma} \varepsilon \mathbb{R}^{n} .$$

Thus, a is decomposed as

(18a) $$a \equiv \begin{pmatrix} a_0 & B_0 \\ C_0^T & B_J \end{pmatrix}$$

with

(18b) $$a_0 \equiv \begin{pmatrix} B_1 & & & \\ A_2 & B_2 & & \\ & \ddots & \ddots & \\ & & A_{J-1} & B_{J-1} \end{pmatrix} , \; B_0 \equiv \begin{pmatrix} A_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} , \; C_0^T \equiv (0 \cdots 0 A_J) ,$$

and (17) becomes:

(19a) $$a_0 \mathbf{x} + B_0 \boldsymbol{\xi} = \mathbf{g} ,$$

(19b) $$C_0^T \mathbf{x} + B_J \boldsymbol{\xi} = \boldsymbol{\gamma} .$$

To solve (19), we first find $\mathbf{u}$ and U from

(20) $$a_0 \mathbf{u} = \mathbf{g} \quad , \quad a_0 U = B_0 .$$

Then $\mathbf{x}$ in the form

(21) $$\mathbf{x} = \mathbf{u} - U \boldsymbol{\xi}$$

will satisfy (19a). To find $\boldsymbol{\xi}$, we use this $\mathbf{x}$ in (19b) to get

$$(22) \qquad [B_J - C_0^T U]\boldsymbol{\xi} = \boldsymbol{\gamma} - C_0^T \mathbf{u} \; .$$

To carry out this procedure, we introduce $\mathbf{w}$ and W where:

$$\mathbf{u} \equiv \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{J-1} \end{pmatrix} , \; U \equiv \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{J-1} \end{pmatrix} , \; \mathbf{w} \equiv \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_{J-1} \end{pmatrix} , \; W \equiv \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_{J-1} \end{pmatrix} .$$

Now (20) can be solved by first solving

$$(23) \qquad B_j \mathbf{w}_j = \mathbf{g}_j \quad , \quad B_j W_j = A_j \quad , \quad j = 1, 2, \cdots, J-1 \quad ;$$

this can be carried out **concurrently** for all $j \leq J-1$. Then we form, sequentially:

$$(24) \qquad \left.\begin{array}{lll} \mathbf{u}_1 = \mathbf{w}_1 & , & \mathbf{u}_j \mathbf{w}_j - W_j \mathbf{u}_{j-1} \\ U_1 = W_1 & , & U_j = -W_j U_{j-1} \end{array}\right\} j = 2, 3, \cdots, J-1 \quad .$$

Next, we must solve the n^{th} order system (22), which becomes:

$$(25) \qquad [B_J - A_J U_{J-1}]\boldsymbol{\xi} = \boldsymbol{\eta} - A_J \mathbf{u}_{J-1} \; .$$

Using (24) and (25), we can evaluate (21) **concurrently** by forming:

$$(26) \qquad \mathbf{x}_j = \mathbf{u}_j - U_j \boldsymbol{\xi} \quad , \quad j = 1, 2, \cdots, J-1 \; .$$

Note that in the above algorithm only (24) and (25) are not concurrent procedures. The main computations, on the other hand, occur in (23). Analyses of bordering algorithms, when $\mathcal{A}_0$ is singular or near singular, are contained in [3,4].

5. Special Block Tridiagonal Case

To solve (13) when $\mathcal{a}$ has the general block tridiagonal form (15b), we can proceed in either of two obvious ways. First, say, factor as:

$$(27a)\qquad \mathcal{a} = DR \equiv \begin{pmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_J \end{pmatrix} \begin{pmatrix} I & X_1 & & & \\ Y_2 & I & X_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & X_{J-1} \\ & & & Y_J & I \end{pmatrix};$$

or second, factor as:

$$(27b)\qquad \mathcal{a} = LD \equiv \begin{pmatrix} I & X_1 & & & \\ Y_2 & I & X_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & X_{J-1} \\ & & & Y_J & I \end{pmatrix} \begin{pmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_J \end{pmatrix};$$

Of course, the $\{X_j\}$ and $\{Y_i\}$ differ in each case. However, in (27a), zero **columns** in the blocks A_i and C_j are preserved in the corresponding Y_i and X_j. In (27b), zero **rows** in the A_i and C_j are preserved in the corresponding Y_i and X_j.

Now, to treat (15a,b,c), we observe that zero rows occur, and hence we use the factorization (27b). To carry this out, we need only solve, concurrently:

$$(28a)\qquad X_j B_{j+1} = C_j \quad , \quad j = 1, 2, \cdots, J-1 \, ;$$

$$(28b)\qquad Y_i B_{i-1} = A_i \quad , \quad i = 2, 3, \cdots, J \, .$$

Note from (15c) that the X_j and Y_i have the form

$$(28c)\qquad X_j = \begin{pmatrix} 0 \\ \hat{X}_j \end{pmatrix} \begin{matrix} \} \; p \\ \} \; q \end{matrix} \quad , \quad Y_i = \begin{pmatrix} \hat{Y}_i \\ 0 \end{pmatrix} \begin{matrix} \} \; p \\ \} \; q \end{matrix} \, .$$

Thus, we need only insert the caret, "^", atop each of X_j, C_j, Y_i and A_i in (28) to indicate the actual computations to be done. To solve (13a) using (15) and the above, we use:

$$D\eta = \xi \tag{29a}$$

$$L\xi = \mathbf{B}\ . \tag{29b}$$

Clearly (29a) can be solved concurrently as

$$B_j \eta_j = \xi_j \quad , \quad j = 1, 2, \cdots, J\ . \tag{30}$$

Of course, to do this, we need the $\{\xi_j\}$ which, from (29b), satisfy:

$$\begin{aligned} &\xi_1 + X_1 \xi_2 = \mathbf{B}_1 \\ &Y_i \xi_{i+1} + \xi_i + X_i \xi_{i-1} = \mathbf{B}_i\ ,\ i = 2, 3, \cdots, J-1 \\ &Y_J \xi_{J-1} + \xi_J = \mathbf{B}_J \end{aligned} \quad . \tag{31}$$

From (28c), in the first and last of (31), we see that the first p components of ξ_1 and the last q components of ξ_J are given by the corresponding components in $\mathbf{B}_1$ and $\mathbf{B}_J$, respectively. Unfortunately, this trivial solution aspect ceases and the remaining $(n \cdot J - n)$ unknowns are sparsely coupled in a block tridiagonal form. The coupling is indicated by the block matrix $\hat{a}$ of order $n(J-1)$ of the schematic form

$$\hat{a} \equiv \begin{pmatrix} \begin{pmatrix} I^q & X^p \\ Y^q & I^p \end{pmatrix} & \begin{pmatrix} X^q & 0 \\ 0 & 0 \end{pmatrix} & & & \\ \begin{pmatrix} 0 & 0 \\ 0 & Y^p \end{pmatrix} & \begin{pmatrix} I^q & X^p \\ Y^q & I^p \end{pmatrix} & \begin{pmatrix} X^q & 0 \\ 0 & 0 \end{pmatrix} & & \\ \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \\ & & & & \begin{pmatrix} X^q & 0 \\ 0 & 0 \end{pmatrix} \\ & & & \begin{pmatrix} 0 & 0 \\ 0 & Y^p \end{pmatrix} & \begin{pmatrix} I^q & X^p \\ Y^q & I^p \end{pmatrix} \end{pmatrix} \ . \tag{32}$$

The orders of these submatrices are:

$$(33) \qquad I^q\,,\; X^q \varepsilon \mathbb{R}^{q\times q}\;;\; I^p\,,\; Y^p \varepsilon \mathbb{R}^{p\times p}\;;\; X^p \varepsilon \mathbb{R}^{q\times p}\;;\; Y^q \varepsilon \mathbb{R}^{p\times q}\,.$$

Efficient concurrent algorithms for solving such systems are being developed.

References

[1] H. B. Keller, 1968, Numerical Methods for Two-Point Boundary-Value Problems, Ginn-Blaisdell, Waltham, MA (184pp).

[2] H. B. Keller, 1976, Numerical Solution of Two Point Boundary Value Problems, *SIAM*, Vol. 24, Regional Conf. Ser. in Appl. Math. (61pp).

[3] H. B. Keller, 1982, Practical Procedures in Path Following Near Limit Points, in *Computing Methods in Applied Sciences & Engineering VI*, eds. R. Glowinski and J. L. Lions, North-Holland, New York (pp 177-183).

[4] H. B. Keller, 1983, The Bordering Algorithm and Path Following Near Singular Points of Higher Nullity, *SIAM J. Sci. and Stat. Comp.*, Vol. 4, No. 4 (pp 573-582).