Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

26. Parallel Implementation of Collocation Methods

R. Yates[1], I. Herrera[2]

1. Introduction. Domain decomposition methods (DDM) have received much attention in recent years[3] in that they offer very effective means for parallelizing computational models of continuous systems. Combining collocation procedures with domain decomposition methods, however, presents complications which must be overcome in order to profit from the advantages of parallel computing. Such methods can in fact be derived using a variety of approaches. One possibility involves the use of Steklov-Poincaré operators [7] while another is to apply an indirect formulation [3],[4],[6],[1]. In this paper, a method is derived based upon the application of collocation together with an indirect formulation which is suitable for parallel computation. As a first approach, we shall consider the case of a symmetric, elliptic differential operator which will allow for the utilization of the conjugate gradient method in a novel manner - where successive iterations involve the (parallel-computed) solutions of local problems.

2. Formulation. We shall use the indirect or Trefftz-Herrera formulation [5] for a boundary-value problem with prescribed jumps (BVPJ) for the case of a symmetric, elliptic operator $\mathcal{L}$ as follows:

Let Ω be a domain in $\mathbb{R}^n$ with external boundary $\partial\Omega$ together with a partition $\Pi = \{\Omega_1, ..., \Omega_E\}$ and internal boundary Σ (Fig. 2.1). Let

$$\mathcal{L}u = -\nabla \cdot \left(\underline{\underline{a}} \cdot \nabla u\right) + cu \tag{2.1}$$

be a symmetric, elliptic operator with $c \geq 0$ and

$$\mathcal{L}u = f \quad on \quad \Omega_i \quad for \quad i = 1, ..., E \tag{2.2}$$

$$u(x) = g(x) \quad on \quad \partial\Omega \tag{2.3}$$

$$[u] = j^0 \quad and \quad [\underline{a}_n \cdot \nabla u] = j^1 \quad on \quad \Sigma \tag{2.4}$$

Then u is said to be a solution of the BVPJ. Here, as in the general theory, the notation $[u] = u_+ - u_-$ and $\dot{u} = \frac{1}{2}(u_+ + u_-)$ is used for the jump of a function and average value across a (possibly discontinuous) internal boundary Σ.

The Green-Herrera formula [2] for this problem is given as

$$P - B - J = Q^* - C^* - K^*; \tag{2.5}$$

where

$$\langle Pu, w\rangle = \int_\Omega w\mathcal{L}udx, \quad \langle Q^*u, w\rangle = \int_\Omega u\mathcal{L}^*wdx, \tag{2.6}$$

$$\langle Bu, w\rangle = \int_{\partial\Omega} u\underline{a}_n \cdot \nabla wds, \quad \langle C^*u, w\rangle = \int_{\partial\Omega} w\underline{a}_n \cdot \nabla uds, \tag{2.7}$$

$$\langle Ju, w\rangle = \big\langle (J^0 + J^1)u, w\big\rangle = -\int_\Sigma j^0\dot{\overline{\underline{a}_n \cdot \nabla w}}ds + \int_\Sigma j^1 wds, \tag{2.8}$$

[1] Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , yates@altcomp.com.mx

[2] Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , iherrera@servidor.unam.mx

[3] See: International Scientific Committee for Domain Decomposition "Proceedings of 13 conferences on Domain Decomposition Methods", www.ddm.org, 1988-2001

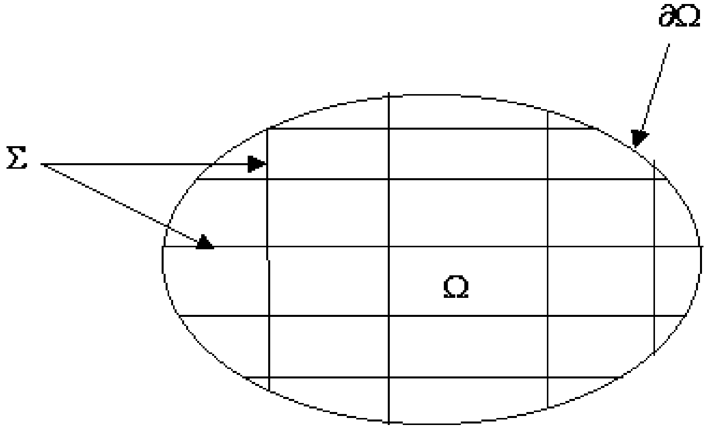

Figure 2.1: Partition of the domain Ω

$$\langle K^* u, w\rangle = \langle (R^* - S^*)u, w\rangle = -\int_\Sigma [w]\dot{\overline{\underline{a}_n \cdot \nabla u}} ds + \int_\Sigma \dot{u}[\underline{a}_n \cdot \nabla w] ds \tag{2.9}$$

In the indirect formulation, the test functions w are chosen so that $w \in \tilde{N} \equiv N_Q \cap N_C \cap N_R$, or equivalently:

$$\mathcal{L}^* w = 0, \quad w = 0 \quad on \quad \partial\Omega \quad and \quad [w] = 0 \quad on \quad \Sigma \tag{2.10}$$

In this case, the resulting formula reduces to

$$\langle S^* u, w\rangle = \langle (P - B - J)u, w\rangle \tag{2.11}$$

which is a variational formulation of the problem. A straightforward calculation shows that

$$\langle S^* u, w\rangle = -\int_\Sigma \dot{u}\,[\underline{a}_n \cdot \nabla w] = \int_\Omega (\nabla u \cdot \underline{\underline{a}} \cdot \nabla w + cuw) \quad \forall u, w \in \tilde{N} \tag{2.12}$$

so that S^* is symmetric, positive-definite when $\mathcal{L}$ is.

In order to obtain the formulation suitable for parallelization, the notion of a *particular solution* must be introduced.

Definition 2.1.- *A function u_p is said to be a* particular solution *of the BVPJ provided*

$$(P - B - J^0)u_p = f - g - j^0 \tag{2.13}$$

or equivalently, if

$$Pu_p = \mathcal{L}u_p = f \quad in\ each \quad \Omega_i \tag{2.14}$$

$$u_p = g \quad on \quad \partial\Omega \tag{2.15}$$

and

$$[u_p] = j^0 \quad on \quad \Sigma \tag{2.16}$$

A particular solution is therefore a function which satisfies the differential operator locally, the external boundary conditions and the jump conditions of the function values on the internal boundary. Nothing is specified regarding the jump conditions of the normal derivative (or flux) of a particular solution. As will be shown later, particular solutions can be obtained readily from solutions of the local problems only. Clearly, a particular solution u_p of the BVPJ is a solution of the BVJP if and only if $J^1 u_p = j^1$.

From this last remark, the following result is easily derived:

Theorem 2.1 *A function u is a solution of the BVPJ if and only if, for any particular solution u_p we have*

$$\langle S^* v, w\rangle = \left\langle J^1 u_p - j^1, w\right\rangle ; \quad \forall w \in \tilde{N} \quad \textit{where} \quad v = u - u_p \tag{2.17}$$

3. The Numerical Algorithm.

To derive a numerical procedure for the BVPJ in a parallel processing environment, we first obtain a matrix equation from the above variational principle and then develop an iterative solution process using the conjugate gradient method in which each iteration involves the solution of local problems in the subregions Ω_i; these local problems can be effectively solved in parallel.

We first assume that we have a particular solution u_p of the BVPJ and, as above, let $v = u - u_p$ where u is the desired solution. Since the differential operator $\mathcal{L}$ is symmetric and $[v] = v|_{\partial\Omega} = 0$, then we have $v \in \tilde{N}$. From the above result Eq. (2.17), we have:

$$\langle S^* v, w\rangle = \left\langle J^1 u_p - j^1, w\right\rangle ; \quad \forall w \in \tilde{N} \tag{3.1}$$

A more explicit form of the above equation is

$$-\int_\Sigma v[\underline{a}_n \cdot \nabla w] ds = \int_\Sigma w([\underline{a}_n \cdot \nabla u_p] - j^1) ds \tag{3.2}$$

To obtain the matrix equation, we will use a system of weighting functions $\{w_1, ..., w_N\}$ of $\tilde{N}$ whose restrictions to the internal boundary Σ form a suitable subspace of $L^2(\Sigma)$. These restrictions, $w_1|_\Sigma, ..., w_N|_\Sigma$ will be used as basis functions to represent v:

$$v = \sum_{j=1}^{N} c_j w_j \tag{3.3}$$

In this case we have:

$$-\sum_{j=1}^{N} c_j \int_\Sigma [\underline{a}_n \cdot \nabla w_i] w_j ds = \int_\Sigma w_i([\underline{a}_n \cdot u_p] - j^1) ds \tag{3.4}$$

which can be rewritten as:

$$\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{c}} = \underline{\mathbf{b}} \tag{3.5}$$

where $A_{ij} = -\int_\Sigma [\underline{a}_n \cdot \nabla w_i] w_j ds$ and $b_i = \int_\Sigma ([\underline{a}_n \cdot \nabla u_p] - j^1) w_i ds$. It should be noted that the matrix $\underline{\underline{\mathbf{A}}}$ is both symmetric and positive definite from Eq. (2.12) as $w_i, w_j \in \tilde{N}$.

The solution of this matrix equation will provide the solution $u = u_p + v$ to the BVPJ on the interior boundary Σ. The solution on Ω can then be obtained by obtaining local solutions in each subregion Ω_i with boundary values supplied by u.

However, a direct computation of the matrix $\underline{\underline{\mathbf{A}}}$ and the obtention of the solution vector $\underline{\mathbf{c}}$ is expensive since the subdomains can be quite large. An alternative approach involves the use of the conjugate gradient method, included in Appendix A, which requires the calculation of the matrix product $\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{c}}$ once for each iteration. In fact, this method does not require the calculation or storage of the components of the matrix $\underline{\underline{\mathbf{A}}}$. Rather, the product is derived from the local solutions of homogeneous BVPs in the subregions Ω_i. To this end, we make use of the following theorem:

Theorem 3.1 *Let $\underline{\mathbf{p}} = (p_1, ..., p_n)$ be a vector and let φ be the solution to the homogeneous boundary value problem in Ω_i defined by:*

$$\mathcal{L}^*\varphi = 0 \quad on \quad \Omega_{\mathrm{i}} \quad for\ each \quad i = 1, ..., E \tag{3.6}$$

$$\varphi = 0 \quad on \quad \partial\Omega \cap \partial\Omega_i \tag{3.7}$$

$$\varphi = \sum_{j=1}^{N} p_j w_j \quad on \quad \Sigma \tag{3.8}$$

then

$$(\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{p}})_i = \int_{\Sigma} w_i [\underline{a}_n \cdot \nabla\varphi] ds \tag{3.9}$$

Proof. Since φ is continuous and vanishes on the exterior boundary $\partial\Omega$, then $\varphi \in \tilde{N}$. Then we must have $\varphi \equiv \sum\limits_{j=1}^{N} p_j w_j$ in each Ω_i so that $\sum\limits_{j=1}^{N} p_j [\underline{a}_n \cdot \nabla w_j] = [\underline{a}_n \cdot \nabla\varphi]$ which is essentially the statement of the theorem. ■

4. The Numerical Procedure. Implementation of the above algorithm requires both the construction of a particular solution u_p and the test function basis $\{w_1, ..., w_N\}$. One way to obtain a particular solution is by setting $\dot{u}_p = 0$ on Σ ie. by solving local problems u_p^i on each Ω_i such that:

$$\mathcal{L}u_p^i = f \quad in\ each \quad \Omega_i \tag{4.1}$$

$$u_p^i = g \quad on \quad \partial\Omega \cap \partial\Omega_{\mathrm{i}} \tag{4.2}$$

$$u_p^i = \pm\tfrac{1}{2} j^0 \quad on \quad \Sigma \cap \partial\Omega_{\mathrm{i}} \tag{4.3}$$

where the sign is chosen according the outward normal.

The resulting u_p will satisfy the differential equation, the external boundary conditions and the internal jump operator J^0. An alternative way of deriving a u_p is to solve first the global BVPJ on the coarse grid and then solve the problem locally on each of the local domains Ω_i using boundary values supplied by the coarse solution. This latter method would tend to give an approximate solution "closer" to the desired solution.

To obtain the test functions, the discretization of the local subdomains gives rise to a discretization of the internal boundary Σ. For each such node point $n_i \in \Sigma$ a test function w_i can be constructed s.t.

$$\mathcal{L}^* w_i = 0 \tag{4.4}$$

$$w_i = 0 \quad on \quad \partial\Omega \cap \partial\Omega_j \quad for\ all \quad j \tag{4.5}$$

and

$$w_i(n_j) = \delta_{ij} \tag{4.6}$$

so that a function $\phi(x)$ on Σ can be approximated as $\phi(x) \simeq \sum\limits_{j=1}^{N} \phi(n_j) w_j(x)$.

Techniques for such constructions using both linear and cubic polynomials on can be found in [5]. It should be stressed that in the computations of the required integrals of the form

$$\int_{\Sigma} w_i[\underline{a}_n \cdot \nabla u]ds = \int_{\Sigma} w_i \bar{a}\frac{\partial u}{\partial n}ds \quad where \quad \bar{a} = \underline{n} \cdot \underline{\underline{a}} \cdot \underline{n} \tag{4.7}$$

that care must be taken to evaluate $\frac{\partial u}{\partial n}$ with consistent precision.

5. Conclusions. The method presented above defines a procedure for parallelizing the numeric solution for second order symmetric elliptic equations. As the solution matrix obtained is symmetric and positive definite, direct application of the conjugate gradient metod is utilized to insure adequate convergence; no preconditioning techniques are required. Moveover, the method is applicable to problems with prescribed jumps (BVPJ) as well as to the case of discontinuous coefficients with no additional complications to the numerical procedure. Finally, it should be stressed that in the solution of the local problems, any numerical procedure can be successfully employed.

Appendix A: The Conjugate Gradient Algorithm

To solve $\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{v}} = \underline{\mathbf{b}}$ where $\underline{\underline{\mathbf{A}}}$ is an $N \times N$ symmetric, positive definite matrix.

```
$\underline{\mathbf{v}}^0 = 0$
$\underline{\mathbf{r}}^0 = \underline{\mathbf{b}}$
$\underline{\mathbf{p}}^0 = \underline{\mathbf{r}}^0$
$k = 0$

while $\left\|\underline{\mathbf{r}}^k\right\|_\infty \geq \varepsilon$
    $\underline{\mathbf{u}} = \underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{p}}^k$[Singlematrixmultiplication/iteration]
    $\alpha^{k+1} = \underline{\mathbf{r}}^k \cdot \underline{\mathbf{r}}^k / (\underline{\mathbf{p}}^k \cdot \underline{\mathbf{u}})$
    $\underline{\mathbf{v}}^{k+1} = \underline{\mathbf{v}}^k + \alpha^{k+1}\underline{\mathbf{p}}^k$
    $\underline{\mathbf{r}}^{k+1} = \underline{\mathbf{r}}^k - \alpha^{k+1}\underline{\mathbf{u}}$
    $\beta^{k+1} = \underline{\mathbf{r}}^{k+1} \cdot \underline{\mathbf{u}} / (\underline{\mathbf{p}}^k \cdot \underline{\mathbf{u}})$
    $\underline{\mathbf{p}}^{k+1} = \underline{\mathbf{r}}^{k+1} - \beta^{k+1}\underline{\mathbf{p}}^k$
    $k = k + 1$
```

REFERENCES

[1] M. Diaz, I. Herrera, and R. Yates. Indirect Method of Collocation: Second Order Equations. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[2] I. Herrera. Unified Approach to Numerical Methods. Part 1. Green's Formulas for Operators in Discontinuous Fields. *Numerical Methods for Partial Differential Equations*, 1(1):12–37, 1985.

[3] I. Herrera. Trefftz-Herrera Domain Decomposition. *Advances in Engineering Software*, 24:43–56, 1995.

[4] I. Herrera. Trefftz Method: A General Theory. *Numerical Methods for Partial Differential Equations*, 16(6):561–580, 2000.

[5] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[6] I. Herrera, R. Yates, and M. Diaz. The Indirect Approach to Domain Decomposition. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[7] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 1999.