CONF-8906268--1

UCRL--102220

DE90 005187

STI

JAN 16 1990

A PARALLEL ITERATIVE METHOD FOR SOLVING 3-D ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS*

Anne Greenbaum
New York University
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, NY 10012

Edna Reiter
Department of Mathematics and Computer Science
California State University at Hayward
Hayward, CA 94542

Garry Rodrigue
Department of Applied Science
University of California, Davis
P. O. Box 808
Livermore, CA 94550

To appear in Proceedings of the NATO
Supercomputer Workshop
Trondheim, Norway
June 19-23, 1989

Lawrence Livermore National Laboratory

A Parallel Iterative Method for Solving 3-D Elliptic Partial Differential Equations*

Anne Greenbaum
New York University
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, NY 10012

Edna Reiter
Department of Mathematics and Computer Science
California State University at Hayward
Hayward, California,94542

Garry Rodrigue
Department of Applied Science
University of California, Davis
P.O. Box 808
Livermore, California, 94550

1 Domain Decomposition

In this paper, we will consider the parallel solution of the Dirichlet problem for a second-order uniformly elliptic equation in two and three dimensions. Specifically, we shall consider the problem

$$\begin{aligned} L(u) &= f \text{ in } \Omega \\ u &= g \text{ on } \Gamma \end{aligned} \tag{1}$$

where

$$L(u) = -\sum_{j=1}^{3} \frac{\partial}{\partial x_i}\left(a_i \frac{\partial u}{\partial x_i}\right)$$

*This work was performed under the auspices of the U.S. Department of Energy at the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48 and at the Courant Institute of Mathematical Sciences under Contract DE-AC02-76ER03077

with a_i positive, bounded, and piecewise smooth on bounded Ω with boundary Γ. For sake of exposition, we will assume the equation is 2-dimensional, however, the extension to 3-dimensions of the numerical methods to be described will be obvious.

The solution of elliptic problems such as those given by (1) arise quite often on supercomputers when they are used to numerically simulate problems arising in mechanical systems. They are often solved when the steady state of dissipative systems as in gas dynamics is desired. These equations are particularly difficult to solve on a computer because their discrete representation on a computational grid yields a linear system of equations where the number of unknowns is equal to the number of grid points on the lattice and, in 3-dimensions, the number of unknowns can be very large, (e.g. 10^6).

A general framework for developing parallel algorithms for solving the elliptic equation (1) on a multiprocessing supercomputer is provided by a techinique that has come to be known as *domain decomposition.* For time-independent problems such as (1), the technique of domain decomposition begins by subdividing the *global* domain Ω into a union of subdomains:

$$\Omega = \bigcup_{i=1}^{n} \Omega_i.$$

Again, for ease of exposition, we will assume $\Omega = [0,1] \times [0,1]$ is the unit cube and the subdomains are themselves cubes, $\Omega_i = [\alpha_i, \beta_i] \times [\gamma_i, \delta_i]$. Examples of such decompositions are given in Figure 1.

For a given decomposition, numerical methods are constructed by solving equation (1) on each of the subdomains and then piecing together the sub- solutions in some manner to provide an approximation to the global solution,[3]. This seems quite straightforward provided there is some mechanism for defining appropriate boundary conditions on each of the subdomain problems. The first attempts at providing subdomain boundary conditions were based around the classical Schwarz alternating method. Here, an initial guess is provided for the boundary conditions and then each sub-problem is solved numerically in some sequential order. Boundary conditions are updated whenever sub-solutions are computed on subdomains containing neighboring subdomain boundaries. As an example, let us consider the

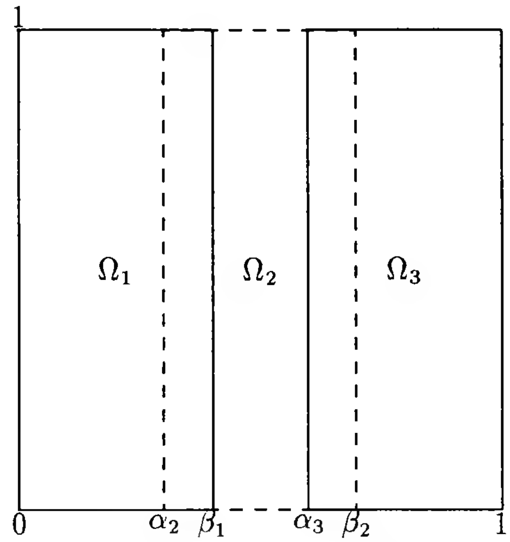

Figure 1: Three Subdomains

solution of (1) using a decomposition such as the one given in Figure 1. Let $u_i^{(0)}$ be given initial guesses for the solution u on each of the subdomains Ω_i. Then for $k = 1, 2, \ldots$, define the sequence of sub-solutions $\{u_i^{(k)}\}$ as follows:

$$Lu_i^{(k)} = f\Big|_{\Omega_1} \quad \text{on } \Omega_1,$$

$$u_i^{(k)} = g \text{ on } \Gamma \bigcap \Gamma_i,$$

$$
u_i^{(k)} = \begin{cases} u_2^{(k-1)}\Big|_{\Gamma} & \text{if } i = 1, \\ u_{i-1}^{(k-1)}\Big|_{\Gamma_{i-1}} & \text{if } 1 < i < n, \\ u_{i+1}^{(k-1)}\Big|_{\Gamma_{i+1}} & \text{if } 1 < i < n, \\ u_{n-1}^{(k-1)}\Big|_{\Gamma_n} & \text{if } i = n. \end{cases}
$$

Convergence of the above procedure to the global solution follows from the maximum principle. The numerical analog of the previous algorithm is obvious and has been shown to be equivalent to a block-Jacobi matrix iterative method, and consequentely, often results in slow convergence, [3]. Specifically, the matrix iteration takes the form

$$
\hat{x}^{(k+1)} = M^{-1}N\ \hat{x}^{(k)}\ +\ \hat{q} \tag{2}
$$

where M is a block-diagonal matrix and N is a non-symmetric matrix. Consequentely, a major effort over the past few years has been to develop iterative methods based around domain decomposition that will converge faster than the Schwarz methods.

2 Conjugate Gradient Acceleration

The easiest and, probably, the most popular way of accelerating an iteration of the form (2) is by the method of conjugate gradients (or, Krylov subspace methods),[2]. Basically, an additive splitting $A = M - N$ of a symmetric ,positive-definite, matrix is given. Then, the following iteration is used to solve the matrix equation $A\hat{x} = \hat{b}$.

Let $\hat{x}^{(0)}$ be an initial guess and $\hat{r}^{(0)} = \hat{b} - A\hat{x}^{(0)}$. Then,for $k = 0, 1, \ldots$, define

$$\begin{aligned}
i) \qquad M\hat{z}^{(k)} &= \hat{b} - A\hat{x}^{(k)}; \\
ii) \qquad \beta_k &= \begin{cases} 0, & k = 0; \\ \dfrac{\hat{z}^{(k)^t} M \hat{z}^{(k)}}{\hat{z}^{(k-1)^t} M \hat{z}^{(k-1)}}, & k \geq 1; \end{cases} \\
iii) \qquad \hat{p}^{(k)} &= \hat{z}^{(k)} + \beta_k \hat{p}^{(k-1)}; \\
iv) \qquad \alpha_k &= \frac{\hat{z}^{(k)^t} M \hat{z}^{(k)}}{\hat{p}^{(k)^t} A \hat{p}^{(k)}}; \\
v) \qquad \hat{x}^{(k+1)} &= \hat{x}^{(k)} + \alpha_k \hat{p}^{(k)}.
\end{aligned} \tag{3}$$

The convergence of this iteration in a finite number of steps is guaranteed if the matrix M is symmetric and positive-definite. In fact, it is known that

$$E(x^{(k)}) \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k E(x^{(0)}) \tag{4}$$

where

$$\begin{aligned}
\kappa &= \frac{\lambda_{max}}{\lambda_{min}}, \\
\lambda_{max} &= \text{largest absolute eigenvalue of } M^{-1}A, \\
\lambda_{min} &= \text{smallest absolute eigenvalue of } M^{-1}A, \\
E(x) &= (x^t A x)^{1/2}.
\end{aligned}$$

Consequentely, convergence will be rapid if the matrix M is *spectrally* close to A in the sense that κ should not be large. The literature on the choice of M is vast especially for the case when the matrix A represents a discrete approximation to (1). However, it is not until recently that reasonable choices of M have been provided that make efficient use of a decomposed domain. The difficulty of using domain decomposition methods in conjugate gradient lies in the origin of each of the two ideas. The conjugate gradient method has its foundation in linear algebra whereas domain decomposition is embedded in the theory of differential equations, and at the present time a unified mathematical connection between the conjugate gradient method and differential equations has not been established.

3 Additive Schwarz Method

Because the convergence rate of the conjugate gradient method depends on the relationship between the matrix A that approximates equation (1) and the preconditioning matrix M, it may be possible to build a matrix M from a domain decomposed differential equation that is related to equation (1). We do so by first splitting the global solution of (1) into the sum of a *homogeneous* solution and a *particular* solution. That is, $u = u^{(H)} + u^{(P)}$ where

$$\begin{aligned} L(u^{(H)}) &= 0 \text{ on } \Omega, \\ u^{(H)} &= g \text{ on } \Gamma, \end{aligned} \tag{5}$$

and

$$\begin{aligned} L(u^{(P)}) &= f \text{ on } \Omega, \\ u^{(P)} &= 0 \text{ on } \Gamma. \end{aligned} \tag{6}$$

Now, let

$$\Omega = \bigcup_{i=1}^{n} \Omega_i \tag{7}$$

be a decomposition of Ω and define the subproblems

$$\begin{aligned} L(u_i) &= f\Big|_{\Omega_i} \quad \text{on } \Omega_i, \\ u_i &= \begin{cases} g & \text{on} \quad \Gamma \bigcap \Gamma_i, \\ u & \text{on} \quad \Gamma_i - \Gamma. \end{cases} \end{aligned} \tag{8}$$

Clearly, $u = u_i$ on Ω_i. Again, $u_i = u_i^{(H)} + u_i^{(P)}$ where

$$\begin{aligned} L(u_i^{(H)}) &= 0 \text{ on } \Omega_i, \\ u_i^{(H)} &= \begin{cases} g & \text{on} \quad \Gamma_i \bigcap \Gamma, \\ u & \text{on} \quad \Gamma_i - \Gamma, \end{cases} \end{aligned} \tag{9}$$

and

$$\begin{aligned} L(u_i^{(P)}) &= f\Big|_{\Omega_i} \quad \text{on } \Omega_i, \\ u_i^{(P)} &= 0 \text{ on } \Gamma_i. \end{aligned} \tag{10}$$

Clearly, each of the "particular" problems defined by (10) can be solved in parallel. Thus, we take as an approximation to $u^{(P)}$ the function $\nu^{(P)}$ where

$$\nu^{(P)} = \sum_{i=1}^{n} \nu_i^{(P)} \tag{11}$$

and

$$\nu_i^{(P)} = \begin{cases} u_i^{(P)} & \text{in} \quad \Omega_i, \\ 0 & \text{in} \quad \Omega - \Omega_i. \end{cases} \tag{12}$$

Then,

$$u \simeq u^{(H)} + \nu^{(P)} \tag{13}$$

and

$$\mathcal{L}(u) \simeq L(u) \tag{14}$$

where

$$\mathcal{L}(u) = L(u^{(H)}) + \left(\sum_{i=1}^{n} L(\nu_i^{(P)}) \right). \tag{15}$$

We now carry out the numerical equivalent of the above construction. To do so, we overlay the global domain Ω by a grid G such that the boundaries Γ_i of the subdomains are contained in G. Now, suppose the grid $G^{(H)}$ also overlays the domain Ω with $G^{(H)} \subset G$. Let

$$G_i = G \bigcap \overline{\Omega}_i$$

where $\overline{\Omega}_i$ is the closure of Ω_i, $\hat{u}$ the discrete representation of the function u on grid G and $\hat{\nu}_i$ the restriction of $\hat{u}$ on G_i. If A_i is the matrix representation of the equation (10) on G_i, then define the matrix $\mathcal{A}_i$ on G by

$$\mathcal{A}_i\hat{u} = \begin{cases} A_i\hat{\nu}_i, & \text{on} \quad G_i \\ \\ 0, & \text{otherwise.} \end{cases}$$

Now let P be an operator that maps grid functions on $G^{(H)}$ into grid functions defined on G by bilinear interpolation. If $A^{(H)}$ is the matrix approximation to equation (5) on $G^{(H)}$, then define

$$\mathcal{A}^{(H)}\hat{u} = A^{(H)}P\hat{u}, \text{ on } G^{(H)}.$$

Now, if M is the preconditioning matrix to be used in the conjugate gradient method, then we carry out the computation of step $(3i)$ by

$$\begin{aligned} z^{(k)} &= M^{-1}r^{(k)} \\ &= \mathcal{A}^{(H)^*}r^{(k)} + \left(\sum_{i=1}^{n} \mathcal{A}_i^{\dagger} r^{(k)}\right) \end{aligned}$$

where the matrices $\mathcal{A}_i^{\dagger}$ are the pseudo-inverses of $\mathcal{A}_i$,i.e.,

$$\mathcal{A}_i^{\dagger} r^{(k)} = \begin{cases} A_i^{-1}r^{(k)} & \text{on} \quad G_i, \\ \\ 0 & \text{otherwise} \quad . \end{cases}$$

and

$$\mathcal{A}^{(H)^*}r^{(k)} = PA^{(H)^{-1}}P^t r^{(k)}.$$

Again, we see that M is symmetric and positive-definite. The above method of constructing M is called the additive Schwarz method,[1],[4],[5].

4 Computational Example

We apply the above the previously described to compute a numerical solution to the 3-dimensional Poisson equation

$$L(u) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f \qquad (16)$$

where

$$
\begin{aligned}
f(x,y,z) &= -2((y-y^2)(z-z^2)+(x-x^2)(z-z^2)+(x-x^2)(y-y^2));\\
(x,y,z) &\in \Omega=[0,1]\times[0,1]\times[0,1];\\
u &= 0 \text{ on } \Gamma.
\end{aligned}
$$

On a uniform grid with meshsize Δ that is overlayed on Ω, the operator L in (16) is central differenced to yield a matrix approximation A where

$$
A=\begin{bmatrix}
E & F & G & & & & \\
F & E & F & G & & & \\
G & F & E & F & G & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & G & F & E & F & G \\
 & & & G & F & E & F \\
 & & & & G & F & E
\end{bmatrix}
$$

where

$$
E=\begin{bmatrix}
6 & -1 & & & \\
-1 & 6 & -1 & & \\
 & \ddots & \ddots & \ddots & \\
 & & -1 & 6 & -1 \\
 & & & -1 & 6
\end{bmatrix}
$$

and

$$
G=F=-\begin{bmatrix}
1 & & & \\
 & \ddots & & \\
 & & \ddots & \\
 & & & 1
\end{bmatrix}.
$$

Let Δ_c be the mesh-size of a grid $G^{(H)}$ and $A^{(H)}$ the matrix approximation of the homogeneous equation (5). The grid lines of $G^{(H)}$ define boundaries of a decomposition

Δ_c	Δ_f	t_s	t_p
1/3	1/16	.243	.199
1/3	1/24	.76	.51
1/3	1/32	2.33	1.25

Table 1: Timings on NMFECC Cray-II

of Ω. Let this decomposition be given by

$$\Omega = \bigcup_{i=1}^{n} R_i.$$

On the closure of each subdomain R_i define a grid G_f of mesh-size Δ_f. Extend the internal boundaries of the regions R_i by Δ_f in each coordinate direction to obtain a larger sub-domain Ω_i. Clearly, $\Omega = \bigcup_{i=1}^{n} \Omega_i$. We use the conjugate gradient method given in (3) where the preconditioner M is defined by the additive Schwarz method. The calculations were executed on the Cray-II located at the NMFECC in the Lawrence Livermore National Laboratory. t_s refers to time in minutes without multiprocessing and t_p refers to the time in minutes with multiprocessing. The banded version of the LU-factorization algorithm was used to solve the linear systems involving $\mathcal{A}_i$ and $\mathcal{A}^{(H)}$ in step ($3i$) of the conjugate gradient method.

5 Summary

We have described an iterative algorithm for solving linear systems arising from the discretization of elliptic partial differential equation. The algorithm combines domain decomposition techiniques and the conjugate gradient method in order to achieve rapid convergence and parallelism. Subdividing the domain allows us to solve 3-dimensional problems in an efficient manner. Table 1 indicates the timings that were achieved on the Cray-II located at the NMFECC. Although the maximum speedup we achieved was 1.9, the parallel performance of the algorithm was disappointing. The idealized timing which assume we receive all four processors simultaneously should produce speedups in the range of 3.5-3.8. We are continuing to research the cause of this poor performance and will be the subject of a future paper.

References

[1] P.E.Bjorstad, Multiplicative and Additive Schwarz Methods: Convergence in the Two-Domain Case, Proc. of the 2nd Int. Symp. on Domain Decomposition Methods, Los Angeles, Calif., Jan. 14-16,1988, SIAM Pulications, Philadelphia, Pa., pp.147-159.

[2] G.Golub and C.Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, Maryland, 1983.

[3] G. Rodrigue, Inner/Outer iterative methods and numerical Schwarz methods, Journal of Parallel Computing, 2, 1985, pp. 205-218.

[4] O. Widlund, Optimal iterative refinement methods, Proc. of the 2nd Int. Symp. on Domain Decomposition Methods, Los Angeles, Calif., Jan. 14-16, 1988, SIAM Publications, Philadelphia, Pa., pp. 114 - 126.

[5] A. Greenbaum, C. Li, and H.Z. Chao, Parallelizing Preconditioned Conjugate Gradient Algorithms, Computer Physics Communications, 53, 1989, pp. 295-309