We present a new domain decomposed fast Poisson solver on a rectangle divided into parallel strips or boxes. The method first performs uncoupled fast solves on each subdomain, and then the interface variables are computed exactly by fast Fourier transform, without computing or inverting the capacitance matrix explicitly. Finally, the solution on the interior of the subdomains can be computed by one more fast solve on each subdomain.

This method, as opposed to others, does not involve any iteration in the solution of the system for the interface variables. It is especially suited for parallel implementation, since the independent problems in the subdomains can be solved in parallel, and the communication involves the interface variables only.

A Domain-Decomposed Fast Poisson Solver on a Rectangle

Tony F. Chan and Diana C. Resasco †

Research Report YALEU/DCS/RR-409
August 1985

The authors were supported in part by the Department of Energy under contract DE-AC02-81ER10996 and by a BID-CONICET fellowship from Argentina.

Keywords: *domain decomposition, substructuring, capacitance matrix, elliptic partial differential equations, preconditioning, fast Poisson solvers, parallel computation.*

†Dept. of Computer Science, Research Center for Scientific Computing , Yale Univ., Box 2158, Yale Station, New Haven, Conn. 06520.

1. Introduction

We consider the solution of the Poisson equation in a rectangular region partitioned into strips. By the method of Domain Decomposition, the solution of the algebraic equations resulting from the discretization on a regular grid is reduced to the solution of problems in the subdomains and a linear system for the interface unknowns, given by the *capacitance matrix.* This is an important tool in the solution of elliptic partial differential equations. There are several reasons why these techniques might be attractive:

- The method is suited for the solution of very large problems on machines with limited storage.
- Special solution techniques might exist to solve the problems on the subdomains that cannot be applied efficiently to the entire domain. This is often the case, for example, when the domain has irregular geometry, but it can be broken up into regular subdomains, like rectangles.
- The equations in the different subdomains might have different parameters or even be of different nature, in which case the idea of substructuring comes very naturally.
- The idea is attractive for parallel processing, since the problem can be decoupled in independent subproblems and the communication needed will be only for the interface values.

The capacitance matrix is expensive to compute or invert explicitly, while it is relatively simple to compute the product of such matrix with an arbitrary vector. For that reason the interface system is usually solved by conjugate gradient methods instead of a direct method. In order to improve the convergence of the method, several preconditioning techniques have been given in the literature [6, 7, 1, 2].

For the problem considered in this paper, we present a fast direct method for the solution to the capacitance matrix that does not require the computation of the elements of the matrix. As oppossed to the other methods given so far, there is no iteration involved.

In the case of uniform sized strips, we derive the eigenvectors and eigenvalues of the capacitance matrix. This gives not only a direct method to solve the interface system, but also a framework to analyse other preconditioners.

In section 2, we apply the method of Domain Decomposition to the solution of the Poisson equation in a rectangular domain subdivided into strips, and derive a system for the interface unknowns and the capacitance matrix. In section 3, we analyze the capacitance matrix, giving its eigenvectors and eigenvalues. Based on this, we derive a fast direct method for the solution of the interface system. The eigen-decomposition of the capacitance matrix also provides a clear way of analyzing some other preconditioners. In section 4, we analyze the preconditioners given by Golub and Mayers, Dryja and Proskurowski, and Bjorstad and Widlund. Finally, in sections 5 and 6, we discuss extensions of the method to rectangular domains subdivided into boxes, and more general variable coefficient operators and summaryze the results, giving some concluding remarks.

2. Domain Decomposition

We consider the Poisson equation

$$\Delta u = f \qquad \text{on} \quad \Omega \tag{2.1}$$

with boundary conditions

$$u = u_b \qquad \text{on} \quad \partial\Omega$$

where the domain Ω is as illustrated in Fig. 1.

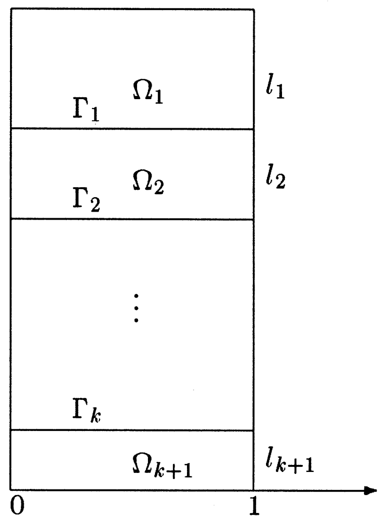

Figure 1: The domain Ω and its partition

We partition Ω into $k+1$ strips $\Omega_i, i = 1, \ldots, k+1$ of sizes 1 by l_i and denote the interfaces between Ω_i and Ω_{i+1}, by $\Gamma_i, i = 1, \ldots, k$. We use a uniform mesh with grid size h on Ω with n internal grid points in the x–direction, i.e.

$$h = \frac{1}{n+1}$$

and m_i internal grid points in Ω_i in the y–direction, i.e. for $i = 1, \ldots, k+1$,

$$l_i = (m_i + 1)h.$$

Let us consider a standard 5–point centered difference approximation to (2.1). If we order the unknowns for the internal points of the subdomains first and then those in the interfaces Γ_i, then the discrete solution vector $u = (u_\Omega, u_\Gamma)$ satisfies a linear system that can be expressed in block form as:

$$\begin{pmatrix} Q_\Omega & P \\ P^T & Q_\Gamma \end{pmatrix} \begin{pmatrix} u_\Omega \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_\Omega \\ f_\Gamma \end{pmatrix}, \tag{2.2}$$

where the right hand side depends on f and u_b and

$$Q_\Omega = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_{k+1} \end{pmatrix}, \tag{2.3}$$

$$Q_\Gamma = \begin{pmatrix} L_{k+2} & & \\ & \ddots & \\ & & L_{2k+1} \end{pmatrix} \tag{2.4}$$

where L_i corresponds to the discrete Laplacian on Ω_i for $i \leq k+1$, and L_i for $i > k+1$ is the tridiagonal matrix *tridiag*$(1, -4, 1)$. The matrix P has the following block bi-diagonal form:

$$P = \begin{pmatrix} P_{1,1} & & & \\ P_{2,1} & P_{2,2} & & \\ & & \ddots & \\ & & P_{k,k-1} & P_{k,k} \\ & & & P_{k+1,k} \end{pmatrix}, \tag{2.5}$$

where P_{ij} corresponds to the coupling between the unknowns in the subdomain Ω_i with those in the interface Γ_j.

By applying block Gaussian Elimination to (2.2), we obtain the following system for the interface unknowns:

$$C u_\Gamma = g \tag{2.6}$$

where

$$g \equiv f_\Gamma - P^T Q_\Omega^{-1} f_\Omega \tag{2.7}$$

and

$$C \equiv Q_\Gamma - P^T Q_\Omega^{-1} P \quad . \tag{2.8}$$

C is sometimes called the *capacitance matrix.* Since Q_Ω is block diagonal, the right hand side of (2.6) given by (2.7) can be evaluated by solving a problem like (2.1) on each subdomain Ω_i, with zero Dirichlet boundary conditions on the interfaces Γ_{i-1} and Γ_i , $i = 1, \ldots, k+1$. Once (2.6) is solved, the problem is decoupled and the solution u_{Ω_i} at the subdomains can be computed by solving $k+1$ independent subproblems:

$$L_i u_{\Omega_i} = f_{\Omega_i} - P_{i,i-1} u_{\Gamma_{i-1}} - P_{i,i} u_{\Gamma_i}$$

for $i = 1, \ldots, k+1$. This is nothing more than solving for u_{Ω_i} on each subdomain Ω_i with the computed $u_{\Gamma_{i-1}}$ and u_{Γ_i} as boundary conditons.

This technique of reducing the problem on Ω to the solution of decoupled problems on the subdomains and a smaller system for the interface is usually called *domain decomposition* or *substructuring.*

3. The capacitance system

By substituting (2.3), (2.4) and (2.5) in (2.8), we can see that the matrix C has the following block tridiagonal form:

$$C = \begin{pmatrix} C_1 & B_2 & & \\ B_2 & C_2 & \ddots & \\ & \ddots & \ddots & B_k \\ & & B_k & C_k \end{pmatrix}, \tag{3.1}$$

where C_i is the capacitance matrix corresponding to the inteface Γ_i, i.e.

$$C_i = L_{k+1+i} - P_{i,i}^T L_i^{-1} P_{i,i} - P_{i+1,i}^T L_{i+1}^{-1} P_{i+1,i} \tag{3.2}$$

and

$$B_i = -P_{i,i-1}^T L_i^{-1} P_{i,i}. \tag{3.3}$$

The matrix C is expensive to compute explicitly, since the computation of the blocks C_i and B_i requires the solution of $2n$ subproblems on each Ω_i. Also, the resulting blocks are dense n by n matrices, and therefore, the solution to (2.6) by Gaussian elimination would require $\mathcal{O}(n^3)$ operations, which may overcome the $\mathcal{O}(n^2 \log(n))$ complexity of the fast solvers for the subdomains, when $m \approx n$.

Instead of using a direct method to solve the system (2.6), preconditioned conjugate gradient (PCG) methods can be applied, where only matrix–vector products of the form Cw for a given vector w are needed. From (3.2), we can see that the evaluation of $C_i w$ for each i requires the solution of two subdomain problems. For example, the product $-P_{2,2}^T L_2^{-1} P_{2,2} w$ can be computed by solving the discretized version of (2.1) on Ω_2 with homogeneous right hand side (i.e., the Laplace equation) and boundary conditions $u = w$ on Γ_2, $u = 0$ on the rest of the boundary, and then taking the solution on the first row of grid points above Γ_2. Similarly, $B_2 w$ corresponds to taking the same solution on the first row below Γ_1.

Each iteration of the PCG method requires the solution of at least one problem at each subdomain. Therefore, it is important to keep the number of iterations low. A number of preconditioners for the matrix C have been given in the literature in order to improve the convergence [6, 7, 1, 2].

It turns out that for the problem we are considering, an exact decomposition for the capacitance matrix is possible. The system can then be solved directly by fast Fourier transforms, and therefore no preconditioned conjugate gradient iteration is necessary. The two strips case is analyzed by Chan in [4], where the eigenvectors and eigenvalues of the capacitance matrix C are given exactly.

Here we extend that result to the multistrip and multi-box cases. We will show that the matrices C_i and B_i in (3.1) have the same eigenvectors. The system (2.6) can thus be solved by fast Fourier transforms and cyclic reduction [3, 8].

Let us define the orthogonal matrix W as the matrix whose columns are

$$w_j = \sqrt{\frac{2}{n+1}} (\sin j\pi h, \sin 2j\pi h, \cdots, \sin nj\pi h)^T \tag{3.4}$$

for $j = 1, \ldots, n$,

$$\sigma_j = 4 \sin \frac{j\pi h}{2}^2 \quad , \tag{3.5}$$

and

$$\gamma_j = \left(1 + \frac{\sigma_j}{2} + \sqrt{\sigma_j + \frac{\sigma_j^2}{4}} \right)^2 \tag{3.6}$$

Then, we have the following

Lemma 3.1. *The matrices C_i and B_i have the same eigenvectors (3.4). For $i = 1, \ldots, k$, we have*

$$W^T C_i W = \Lambda_i = \ diag(\lambda_{i1}, \ldots, \lambda_{in}) \tag{3.7}$$

and for $i = 2, \ldots, k$, we have

$$W^T B_i W = D_i = \ diag(\delta_{i1}, \ldots, \delta_{in}) \tag{3.8}$$

where

$$\lambda_{ij} = - \left(\frac{1 + \gamma_j^{m_i+1}}{1 - \gamma_j^{m_i+1}} + \frac{1 + \gamma_j^{m_{i+1}+1}}{1 - \gamma_j^{m_{i+1}+1}} \right) \sqrt{\sigma_j + \frac{\sigma_j^2}{4}}, \tag{3.9}$$

and

$$\delta_{ij} = \sqrt{\gamma_j^{m_i}}\left(\frac{1-\gamma_j}{1-\gamma_j^{m_i+1}}\right). \tag{3.10}$$

Proof. We will compute the three terms of $C_i w_j$ in (3.2) separately. As we stated before, the product $-P_{i,i}^T L_i^{-1} P_{i,i} w_j$ can be computed by solving the problem

$$\begin{aligned} \Delta_h v &= 0 && \text{in} \quad \Omega_i \\ v &= 0 && \text{on} \quad \partial\Omega_i \cap \partial\Omega \\ v &= 0 && \text{on} \quad \Gamma_{i-1} \\ v &= w_j && \text{on} \quad \Gamma_i \end{aligned} \tag{3.11}$$

and taking the values of the solution at the first row of grid points above Γ_i. By using separation of variables, it can be shown [4] that the solution to (3.11) at each grid point (s,t) is given by:

$$v(sh, th) = (c_1 r_+^t + c_2 r_-^t)\sin sj\pi h, \tag{3.12}$$

where r_+ and r_- are the roots of the characteristic polynomial corresponding to substituting (3.12) in (3.11), namely,

$$\begin{aligned} r_+ &= 1 + \frac{\sigma_j}{2} + \sqrt{\sigma_j + \frac{\sigma_j^2}{4}} \\ r_- &= 1 + \frac{\sigma_j}{2} - \sqrt{\sigma_j + \frac{\sigma_j^2}{4}} \end{aligned} \tag{3.13}$$

with σ_j given by (3.5). The constants c_1 and c_2 are determined by the boundary conditions and they are given by

$$\begin{aligned} c_1 &= -\frac{r_-^{m_1+1}}{r_+^{m_1+1} - r_-^{m_1+1}} \\ c_2 &= \frac{r_+^{m_1+1}}{r_+^{m_1+1} - r_-^{m_1+1}}. \end{aligned}$$

Therefore, we can compute the product $-P_{i,i}^T L_i^{-1} P_{i,i} w_j$ by taking $t = 1$ in (3.12), i.e.

$$-P_{i,i}^T L_i^{-1} P_{i,i} w_j = \left(\frac{r_- - r_+\gamma^{m_i+1}}{1-\gamma^{m_i+1}}\right) w_j \quad , \tag{3.14}$$

where

$$\gamma_j = \frac{r_-}{r_+} \quad . \tag{3.15}$$

Since $r_+ r_- = 1$, $\gamma_j = r_-^2$ and therefore (3.6) and (3.15) are equivalent. By a similar computation we can prove that

$$-P_{i+1,i}^T L_{i+1}^{-1} P_{i+1,i} w_j = \left(\frac{r_- - r_+\gamma_j^{m_{i+1}+1}}{1-\gamma_j^{m_{i+1}+1}}\right) w_j \quad . \tag{3.16}$$

Finally, it can also be verified that

$$L_{k+1+i} w_j = (-2 - \sigma_j) w_j. \tag{3.17}$$

Combining (3.14),(3.16) and (3.17) we have

$$C_i w_j = \lambda_{ij} w_j$$

where

$$\lambda_{ij} = -2 - \sigma_j + \left(\frac{r_- - r_+ \gamma_j^{m_1+1}}{1 - \gamma_j^{m_1+1}} \right) + \left(\frac{r_- - r_+ \gamma_j^{m_2+1}}{1 - \gamma_j^{m_2+1}} \right) \quad ,$$

which, after some simplifications, leads to the expression (3.9).

By similar arguments, $B_i w_j$ requires the computation of the solution to (3.11) at the m_i-th row away from Γ_i and we can prove that:

$$B_i w_j = v(\cdot, m_i h) = -\delta_{ij} w_j$$

where

$$\begin{aligned} \delta_{ij} &= c_1 r_+^{m_i} + c_2 r_-^{m_i} \\ &= \sqrt{\gamma_j^{m_i}} \left(\frac{1 - \gamma_j}{1 - \gamma_j^{m_i+1}} \right) \quad . \end{aligned}$$

∎

Let us partition the vectors u_Γ and g in (2.6) as:

$$u_\Gamma = \begin{pmatrix} u_{\Gamma_1} \\ u_{\Gamma_2} \\ \vdots \\ u_{\Gamma_k} \end{pmatrix} \qquad g = \begin{pmatrix} g_{\Gamma_1} \\ g_{\Gamma_2} \\ \vdots \\ g_{\Gamma_k} \end{pmatrix}$$

where, for $i = 1, \ldots, k$,

$$u_{\Gamma_i} = \begin{pmatrix} u_{1i} \\ u_{2i} \\ \vdots \\ u_{ni} \end{pmatrix} \qquad g_{\Gamma_i} = \begin{pmatrix} g_{1i} \\ g_{2i} \\ \vdots \\ g_{ni} \end{pmatrix} .$$

The system (2.6) may be written

$$\begin{aligned} &C_1 u_{\Gamma_1} + B_2 u_{\Gamma_2} = g_{\Gamma_1} \\ &B_i u_{\Gamma_{i-1}} + C_i u_{\Gamma_i} + B_{i+1} u_{\Gamma_{i+1}} = g_{\Gamma_i}, \qquad i = 2, \ldots, k-1 \\ &B_k u_{\Gamma_{k-1}} + C_k u_{\Gamma_k} = g_{\Gamma_k} \end{aligned} \tag{3.18}$$

Using (3.7) and (3.8), the system (3.18) becomes

$$\begin{pmatrix} \Lambda_1 & D_2 & & \\ D_2 & \ddots & \ddots & \\ & \ddots & \ddots & D_k \\ & & D_k & \Lambda_k \end{pmatrix} \begin{pmatrix} \hat{u}_{\Gamma_1} \\ \hat{u}_{\Gamma_2} \\ \vdots \\ \hat{u}_{\Gamma_k} \end{pmatrix} = \begin{pmatrix} \hat{g}_{\Gamma_1} \\ \hat{g}_{\Gamma_2} \\ \vdots \\ \hat{g}_{\Gamma_k} \end{pmatrix} \tag{3.19}$$

where, for $i \le k$,

$$\hat{g}_{\Gamma_i} = W^T g_{\Gamma_i} \tag{3.20}$$

and

$$u_{\Gamma_i} = W \hat{u}_{\Gamma_i}. \tag{3.21}$$

By reordering the equations in (3.19), the system can be reduced to the solution of n tridiagonal systems of dimension k and u_Γ can then be computed by by the expression (3.21). Note that Fast Fourier Transforms can be used in computing $\hat{g}$ and u_Γ. An outline of the algorithm follows:

ALGORITHM DDFPS (Domain-Decomposed Fast Poisson Solver)

Step 1: Compute right hand side g by (2.7). This requires the solution of $k+1$ decoupled problems of the form (2.1) on $\Omega_i, i = 1, \ldots, k+1$, namely

$$\begin{aligned} \Delta_h v_i &= f \quad &&\text{in} \quad \Omega_i \\ v_i &= u_{b_i} \quad &&\text{on} \quad \partial\Omega_i \cap \partial\Omega \\ v_i &= 0 \quad &&\text{on} \quad \Gamma_{i-1} \\ v_i &= 0 \quad &&\text{on} \quad \Gamma_i \end{aligned}$$

and then compute

$$g_{\Gamma_i} = f_{\Gamma_i} - v_i(.,m_i) - v_{i+1}(.,1)$$

Step 2: Obtain new right hand side $\hat{g}$ by (3.20) using Fast Fourier Transforms.

Step 3: For $j = 1, \ldots, n$, solve

$$\begin{pmatrix} \lambda_{1j} & \delta_{2j} & & \\ \delta_{2j} & \ddots & \ddots & \\ & \ddots & \ddots & \delta_{kj} \\ & & \delta_{kj} & \lambda_{kj} \end{pmatrix} \begin{pmatrix} \hat{u}_{j1} \\ \hat{u}_{j2} \\ \vdots \\ \hat{u}_{jk} \end{pmatrix} = \begin{pmatrix} \hat{g}_{j1} \\ \hat{g}_{j2} \\ \vdots \\ \hat{g}_{jk} \end{pmatrix} \tag{3.22}$$

Step 4: Fast Fourier Transform the resulting u_{Γ_i}'s to obtain u_{Γ_i} by (3.21).

Step 5: Solve the following problem in each subdomain

$$\Delta_h u_i = f \qquad \text{on} \quad \Omega_i$$

with boundary conditions

$$\begin{aligned} u_i &= u_{b_i} \quad &&\text{on} \quad \partial\Omega_i \cap \partial\Omega \\ u_i &= u_{\Gamma_{i-1}} \quad &&\text{on} \quad \Gamma_{i-1} \\ u_i &= u_{\Gamma_i} \quad &&\text{on} \quad \Gamma_i \end{aligned}$$

3.1. Uniform size strips

When $m_i = m$ for all i, i.e. all strips have identical dimensions, all the blocks C_i in (3.1) are identical, as well as the blocks B_i. In this case, the eigenvalues and eigenvectors of C can be explicitly computed.

Lemma 3.2. *If $m_i \equiv m, 1 \le i \le k+1$, then the capacitance matrix has the form*

$$C = \begin{pmatrix} C_1 & B & & \\ B & C_1 & \ddots & \\ & \ddots & \ddots & B \\ & & B & C_1 \end{pmatrix} ,$$

and the eigenvalues of C are given by

$$\alpha_{ij} = \lambda_j + \delta_j cos \frac{i\pi}{k+1}$$

for $i = 1, \dots k$ and $j = 1, \dots n$, where λ_j are the eigenvalues of C_1 and δ_j are the eigenvalues of B. The eigenvectors v_{ij} are the direct product of the vectors w_j defined in (3.4) and

$$z_i = (\sin \frac{\pi i}{k+1}, \sin 2\frac{\pi i}{k+1}, \cdots, \sin k\frac{\pi i}{k+1})^T ,$$

i.e.

$$v_{ij} = z_i \otimes w_j = \begin{pmatrix} (\ sin \frac{\pi i}{k+1}) w_j \\ (\ sin \frac{2\pi i}{k+1}) w_j \\ \vdots \\ (\ sin \frac{k\pi i}{k+1}) w_j \end{pmatrix} .$$

Proof. It can be verified that

$$C v_{ij} = \alpha_{ij} v_{ij}$$

by simple substitution and using the fact that z_i and α_{ij} are the eigenvectors and eigenvalues of the tridagonal matrix:

$$T_j = \begin{pmatrix} \lambda_j & \delta_j & & \\ \delta_j & \ddots & \ddots & \\ & \ddots & \ddots & \delta_j \\ & & \delta_j & \lambda_j \end{pmatrix} ,$$

∎

As an improvement over the plain Fourier Analysis method, which has complexity $\mathcal{O}(knlog_2n)$, block-cyclic reduction techniques can be combined with fast Fourier transforms to get an algorithm for solving the interface system for the regular sized strips case, which has asymptotic complexity $\mathcal{O}(knlog_2log_2n)$ [3, 9] .

4. Analysis of some preconditioners

As we mentioned before, the system (3.1) was previously solved by the preconditioned conjugate gradient method. Several preconditioners have been proposed in order to improve the convergence of the method. For the case where there is only one interface ($k = 1$), Dryja [6] proposed as preconditioner for C the square root of the one-dimensional discrete Laplacian. This matrix, which will be denoted by M_D, can be inverted by fast Fourier transform, since it has the following decomposition:

$$M_D = W diag(\lambda_1^D, \lambda_2^D, \cdots, \lambda_n^D) W^T \tag{4.1}$$

where the columns of W are given by (3.4) and

$$\lambda_j^D = -2\sqrt{\sigma_j} \tag{4.2}$$

with σ_j given by (3.5). Golub and Mayers [7] improve Dryja's results with a preconditioner given by

$$M_G = W diag(\lambda_1^G, \lambda_2^G, \cdots, \lambda_n^G) W^T \tag{4.3}$$

where

$$\lambda_j^G \equiv -2\sqrt{\sigma_j + \frac{\sigma_j^2}{4}} \quad . \tag{4.4}$$

For the two strips case, Bjorstad and Widlund [1] give the following approximation to C as a preconditioner:

$$M_B = L_3 - 2P_{1,1}^T L_1^{-1} P_{1,1}$$

When the two strips have the same dimensions, this preconditioner is exact, although their method involves an extra solve in the subdomain Ω_1 in order to invert M_B, as opposed to solving by only one fast Fourier transform on the interface.

In [5], Dryja and Proskurowski propose an algorithm for the case of multiple strips. By ignoring the off-diagonal blocks B_i in (3.1) and using the preconditioners for the two-strips case on the diagonal, they propose using either the block-diagonal matrix diag(M_D) or diag(M_G).

Using the results of the previous section, we can analyze these preconditioners. It is a well known fact that the rate of convergence of the preconditioned conjugate gradient method depends on the condition number of the matrix $M^{-1}C$, where M is the preconditioner, being close to one. We will use the 2-norm condition number in this paper, and for simplicity, we will only consider the case of regular sized strips, with $m_i = m$. In [4], Chan shows for the two-strips case that $K(M_G^{-1}C)$ and $K(M_D^{-1}C)$ depend on the aspect ratio

$$\alpha = (m+1)/(n+1) \tag{4.5}$$

and in particular,

$$\lim_{h \to 0} K(M_G^{-1}C) = \frac{1+e^{-2\pi\alpha}}{1-e^{-2\pi\alpha}} \quad . \tag{4.6}$$

The asymptotic expression (4.6) is plotted in Fig. 2. As we can see, the condition number will be large for small α.

For the case of multiple strips, we have to analyze the effect of dropping the blocks B_i's. Note that when the eigenvalues of B_i, namely δ_j, are small compared to λ_j, the system (3.22) becomes diagonal and therefore, it can be trivially solved. We will see that the values of δ_j are small compared to λ_j when the aspect ratio (4.5) is large. This can be derived from Lemma 3.2, where we give an asymptotic expression for the ratio $\frac{\delta_j}{\lambda_j}$ in terms of the aspect ratio α when $h \to 0$.

By computing (3.10) and (3.9) for the regular strips case, we can see that

$$\frac{\delta_j}{\lambda_j} = \frac{\gamma_j^{(m+1)/2}}{1+\gamma_j^{m+1}} \quad , \tag{4.7}$$

which is a decreasing function of j. Therefore,

$$\frac{\delta_1}{\lambda_1} = \max_{1 \le j \le n} \frac{\delta_j}{\lambda_j} \quad .$$

Moreover, we have:

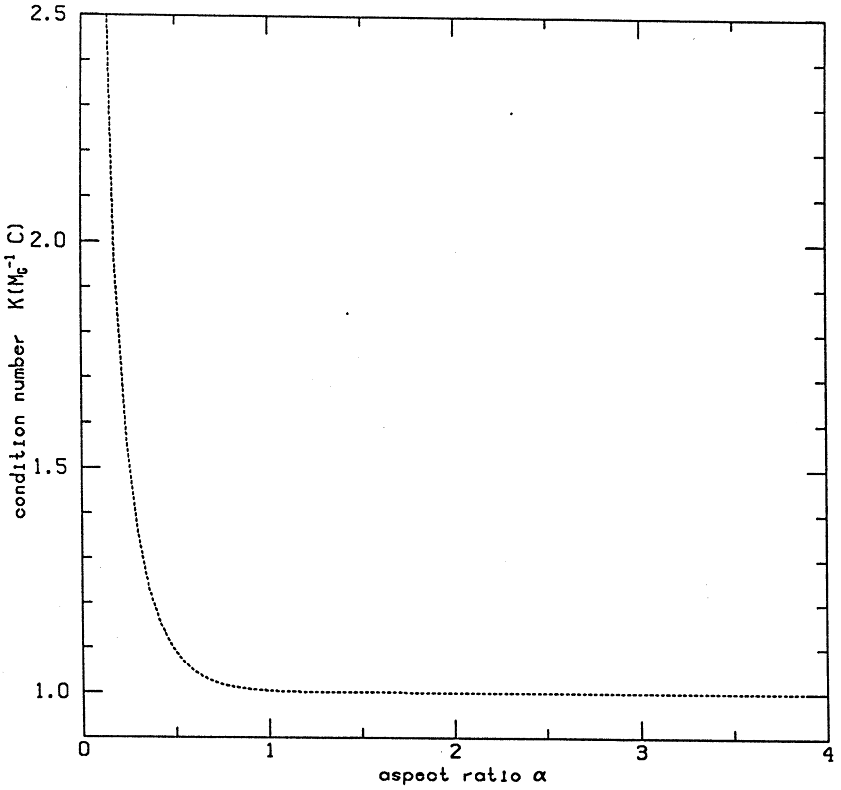

Figure 2: Condition number of the preconditioned capacitance matrix with Golub and Mayers' preconditoner (asymptote for $h \to 0$).

Lemma 4.1.

$$\lim_{h\to 0} \frac{\delta_1}{\lambda_1} = \frac{1}{e^{\alpha\pi} + e^{-\alpha\pi}} \quad .$$

Proof. Since $\sigma_1 \to 0$ when $h \to 0$, we have

$$\lim_{h\to 0}(\log \gamma_1^{m+1}) = \alpha \lim_{h\to 0} \frac{1}{\gamma_1}\frac{d\gamma_1}{dh} = -2\alpha\pi \quad .$$

Therefore,

$$\lim_{h\to 0} \gamma_1^{m+1} = e^{-2\alpha\pi} \quad . \tag{4.8}$$

By (4.7) and (4.8), we have

$$\lim_{h\to 0} \frac{\delta_1}{\lambda_1} = \frac{e^{-\alpha\pi}}{1+e^{-2\alpha\pi}} = \frac{1}{e^{\alpha\pi} + e^{-\alpha\pi}} \quad .$$

∎

When α is large enough, the δ_j's can be ignored, so that the system (3.1) becomes block-diagonal and moreover, for α large, the eigenvalues λ_{ij} approach (4.4) and in that case, (4.3) is a

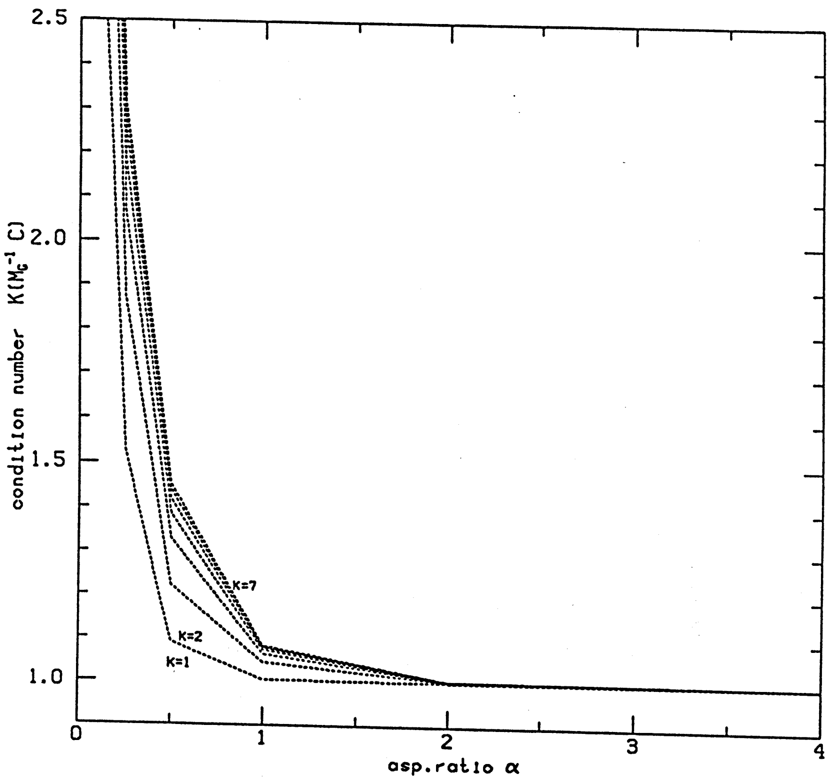

Figure 3: Condition number of the preconditioned capacitance matrix with Golub and Mayers' preconditoner for two, three, and up to eight strips (k is the number of interfaces).

good approximation to C. For α small, the condition number of the preconditioned capacitance matrix becomes large. Fig. 3 shows the dependence of the condition number on the aspect ratio α for different numbers of strips.

5. Dividing the domain into boxes

In the parallel implementation of domain decomposition, the ratio between the amount of communication and the amount of computation required grows as the strips become narrower. For this reason, it seems natural to divide the domain into boxes.

Boxes introduce a new difficulty, the cross points. Cross points are more difficult to handle, since they represent a strong coupling between interfaces. We refer the interested reader to [2].

One way around cross points is to use algorithm DDFPS recursively. We can, for example, apply algorithm DDFPS to the domain Ω divided into horizontal strips Ω_i, and then solve the subproblems in steps 1 and 5 by subdividing each strip Ω_i in vertical strips Ω_{ij}. Another approach is given by nested dissection, which can be described recursively as follows: the domain Ω is divided in two strips Ω_1 and Ω_2. We will call this partition P_1, and given a partition P_i, P_{i+1} is obtained by subdividing each subdomain of P_i into two (vertical or horizontal) strips. An example of the partition procedure is given in Fig. 4 Algorithm DDFPS with $k = 1$ is applied to P_1, and it is

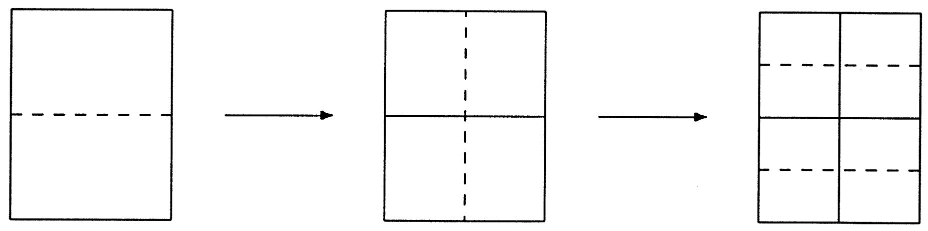

Figure 4: An example of nested dissection

applied recursively in steps 1 and 5 to solve for the problems in the subdomains. Conventional fast solvers are used for the subproblems in the finest partition.

6. Concluding remarks

We present a fast direct Poisson solver on a rectangle divided into parallel strips or boxes. The method, as opposed to others, does not involve any iteration in the solution of the system for the interface variables. It is especially suited for parallel implementation, since the independent problems in the subdomains can be solved in parallel, and the communication can be reduced to k vectors of length n, where n is the number of points in the interface.

Since the method is based on the fact that C can be solved by FFT's, it can be generalized to other separable operators with coefficients which are constant in the x or y direction, and other boundary conditons.

It appears that domain decomposition requires more computation than a conventional fast solver on the whole domain, since the algorithm requires two solves on each subdomain, namely steps 1 and 5. Many operations can be saved, however, since only one row of the solution is needed in step 1 for the computation of the right hand side (2.7). Also, some of the computations in step 1 can be saved for step 5, since the right hand side for the subproblems in steps 1 and 5 differ only at one or two rows of the grid points. With these savings, algorithm DDFPS can be made computationally equivalent to the fast Poisson solver on a rectangle using FFT's in the x-direction and solving n tridiagonal systems in the y-direction.

References

[1] P. Bjorstad and O. Widlund, On Some Trends in Elliptic Problem Solvers, G. Birkhoff and A. Schoenstadt ed., *Elliptic Problem Solvers,* Academic Press, New York, 1984.

[2] J.H. Bramble, J.E. Pasciak and A.H. Schatz, *Preconditioners for Interface Problems on Mesh Domains,* manuscript.

[3] B. L. Buzbee, G. H. Golub and C. W. Nielson, *On Direct Methods for Solving Poisson's Equations,* SIAM J. Numer. Anal., 7 (1970), pp. 627–656.

[4] T.F. Chan, *Analysis of Preconditioners for Domain Decomposition,* Technical Report YALEU/DCS/RR-408, Yale Computer Science Department, 1985.

[5] M. Dryja and W. Proskurowski, Capacitance Matrix Method using Strips with Alternating Neumann and Dirichlet Boundary Conditions, R. Vichnevetsky and R. S. Stepleman ed., *Advances in Computer Methods for Partial Differential Equations - V,* IMACS, 1984, pp. 360–368.

[6] M. Dryja, *A Capacitance Matrix Method for Dirichlet Problem on Polygonal Region,* Numer. Math., 39 (1982), pp. 51–64.

[7] G. H. Golub and D. Mayers, *The Use of Pre-Conditioning over Irregular Regions,* 1983. Lecture at Sixth Int. Conf. on Computing Methods in Applied Sciences and Engineering, Versailles, Dec. 1983.

[8] P. N. Swarztrauber, *A Direct Method for the Discrete Solution of Separable Elliptic Equations,* SIAM J. Numer. Anal., 11 (1974), pp. 1136–1150.

[9] ————, *A Direct Method for the Discrete Solution of Separable Elliptic Equations,* SIAM Review, 19 (1976), pp. 490–501.