On domain decomposition solvers for domains with substructures

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computing Science

RANA 01-15
April 2001

On domain decomposition solvers for domains with substructures

by

W. Drenth and J. Maubach

Reports on Applied and Numerical Analysis
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
ISSN: 0926-4507

On domain decomposition solvers for domains with substructures

W. Drenth and J. Maubach
Department of Mathematics and Computer Science
Eindhoven University of Technology, The Netherlands

Abstract

This paper considers domain decomposition based solvers. The domains of interest are those with specific recurrent substructures, resolved with bisection refinement. For elliptic problems and a range of different domain decompositions, we present a qualitative analysis of the Schur complement's spectral condition number $\kappa_2(\mathbf{S})$. The result, a best condition number of $O(H^{-1})$, is similar to that for regular triangulations in [12]. The Schur complement with the best spectral condition number involves a matrix block, related to local refinement along a line. For this block, we present a numbering of the degrees of freedom, which induces a band-width comparable to the band-width-optimizing reversed Cuthill-McKee ordering [6], and a fill-in comparable to the fill-in-optimized minimum degree algorithm [7], and [15]. This combination of close to optimal band-width and optimal fill-in (np-hard problems) permits the construction of efficient solvers for the Schur complement block.

1 Introduction

Domain decomposition techniques are applied when different levels of detail play a role: When a domain has similar subregions (see chapter 10 of [12] for a recent overview), when the equation's material coefficients differ in scale between subregions, and when governing equations differ from one subregion to another. Further reasons for domain decomposition are varying solution smoothness and the desire to use available subdomain solvers, in a parallel fashion.

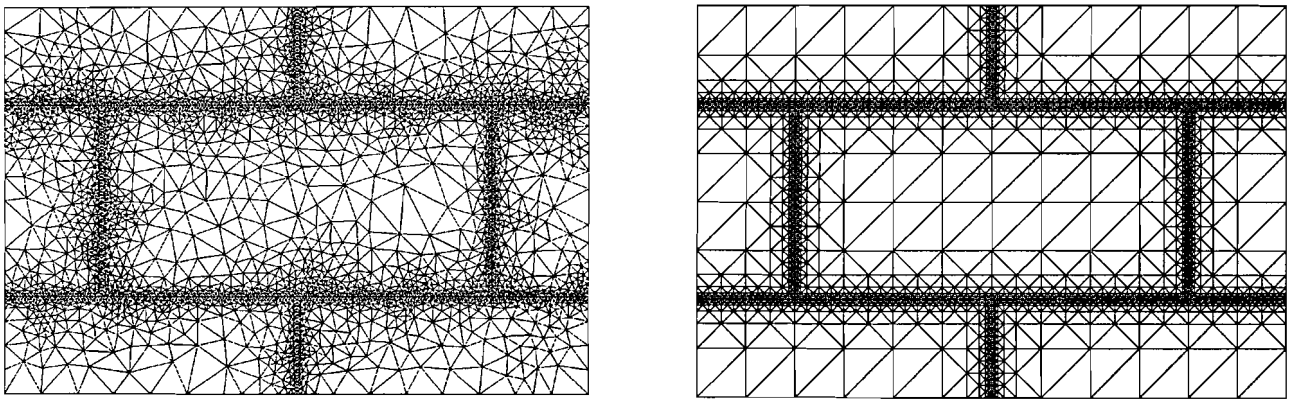


Figure 1: A piece of wall covered by an unstructured and a semi-regular grid

This paper focuses on the case where the domain can be divided into similar substructures. For instance a piece of a brick wall, as shown in figure 1, where moisture and salt ion concentrations must be computed. The prediction of salt ion transport is important, because imbalanced salt concentrations damage bricks and houses.

The simplest moisture transport model in [4], is a non-trivial potential model (elliptic). However, more complicated models make use of convection, material coefficient jumps, and different state equations in mortar and bricks. We focus on the elliptic problem but the to be presented Gaussian elimination based subdomain solver can be used as well when convection is involved.

The local bisection refinement from [11] best preserves the regular substructures of the domain of interest, much better than the more common Delaunay type gridding (see [8]). Both grids in figure 1 – left Delaunay, right bisection – are obtained with the use of the same input data. The bisection refinement preserves the substructures best, and in fact leads to semi-regular grids.

An even more important stimulance to use grids created by bisection refinement are: Semi-regular grids are obtained also in 3 and more dimensions, efficient storage scheme exist, and optimal order preconditioners exist for such refinement along lines [10].

Several efficient solvers have been published for elliptic problems, discretized with finite elements on refined grids. For instance, the optimal AMLI [1, 2], and optimal BPX [3]. In principle, these solvers can solve the entire problem without domain decomposition, in the elliptic case. When convection dominates, the solvers no longer function.

For a range of different domain decompositions, this paper presents a qualitative analysis of the spectral condition number of the Schur complement. We find that the domain decomposition which produces the best condition number $\kappa_2(\mathbf{S})$ is the one which captures all local refinement in one subdomain. This subdomain contains the line along which all mesh refinement is situated.

For this special subdomain, we demonstrate that a Gaussian elimination can be done such that a near optimal band-width and near optimal fill-in is achieved. The results are based on a proper numbering of the degrees of freedom. The minimum degree algorithm [7] has a bit smaller fill-in, even though no optimal "tie breaking" is strategy known when different degrees of freedom have the same connectivity.

When solving the related system of linear equations, our exact factors can be replaced with an incomplete version (see for instance [12], [9], [5] and [13]), combined with Krylov-space based iterative solvers [16], [14]. However, because the focus of this paper is on demonstrating that we get optimal fill-in and band width matrix factors for the full Gaussian elimination process, we don't provide an analysis of the incomplete case.

The paper is composed as follows. In section 2 we present the standard finite element discretization of interest, and several suitable domain decompositions. For these domain decompositions, section 3 illustrates how $\kappa(\mathbf{S})$ depends on the largest and smallest mesh-widths. For the domain decomposition which induces the smallest condition number $\kappa(\mathbf{S})$, section 4 presents a numbering of degrees of freedom for which Gaussian elimination leads to an amount of fill-in close to that of the fill-in-optimized minimum degree algorithm [7], and on top leads to a band-width which is close to that of the reversed Cuthill-McKee ordering [6].

2 The domain decompositions

First, we formulate the standard Galerkin finite element discretization for a elliptic problems. Then, we present the different domain decompositions of interest, for a specific semi-regular refined grid. Thereafter, in anticipation of the numerical results in section 3, an overview of some known properties of $\kappa_2(\mathbf{A})$ for regular grids is provided.

For the remainder of this paper, we first restrict our attention on the bisection refinement along a straight line as in figure 2:

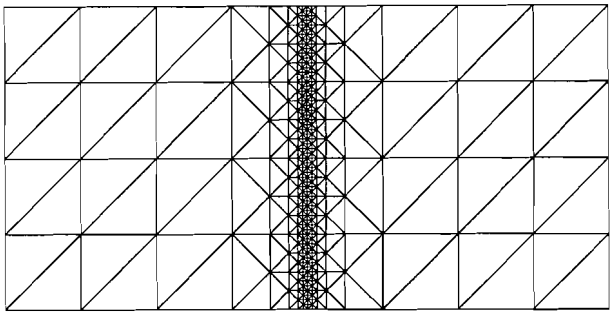


Figure 2: Uniform refined quadrilateral with local refinement.

Furthermore, in order to restrict the amount of possible domain decompositions, we assume that its building blocks are those presented in figure 3.

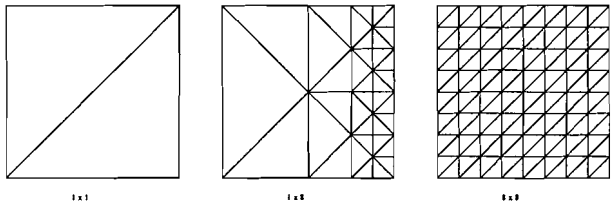


Figure 3: Grids which capture different levels of detail (1x1, 1x8 and 8x8).

That is, each subdomain will contain either a uniform grid, or all or half of the refinement.

The problem for which the spectral condition number of the Schur complement is calculated is the standard stiffness problem, in a bounded simply connected polyhedral region $\Omega \subset \mathbb{R}^2$. For $\epsilon > 0$, the problem of interest is: Find $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases} -\nabla \cdot \epsilon \nabla u &= f, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega. \end{cases} \tag{1}$$

Let $H_0^1(\Omega)$ be the standard Sobolev space of functions that are square integrable, vanish on $\partial\Omega$ and have square integrable partial derivatives. We use homogeneous boundary conditions solely to simplify the presentation. The Galerkin variational formulation related to (1) is: Find $u \in H_0^1(\Omega)$ such that

$$\int_\Omega \epsilon \nabla u^T \nabla v = \int_\Omega f v, \qquad \forall v \in H_0^1(\Omega). \tag{2}$$

Assume that the domain Ω is covered with a grid consisting of a finite amount of topologically closed triangles, which cover Ω but whose interiors' intersection is empty. Assume that the smallest triangle's diameter is $h > 0$. Let $V_h \subset H_0^1(\Omega)$ be the related finite dimensional subspace, spanned by a basis $\{v_i\}_{i=1}^N$ of conforming linear basis functions. The finite element problem to be solved is: Find $u = \sum u_i v_i$ such that

$$\int_\Omega \epsilon \nabla u^T \nabla v = \int_\Omega f v, \qquad \forall v \in V_h. \tag{3}$$

The coefficient vector $\mathbf{x} = [u_1, \ldots, u_N]$ related to the solution of (3) solves

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4}$$

where

$$[\mathbf{A}]_{ij} = \int_\Omega \epsilon \nabla v_j^T \nabla v_i \quad \text{and} \quad \mathbf{b}_i = \int_\Omega f v_i, \tag{5}$$

for all i.

In the remainder, we will further restrict the domain decompositions of interest by taking at most three subregions. For such cases, for a proper numbering of the degrees of freedom $[u_1, \ldots, u_N]$, the matrix $\mathbf{A}$ in (5) has the structure:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B}_{11} & 0 & \mathbf{B}_{13} \\ 0 & \mathbf{B}_{22} & \mathbf{B}_{23} \\ \mathbf{B}_{31} & \mathbf{B}_{32} & \mathbf{B}_{33} \end{bmatrix}. \tag{6}$$

The actual choice of subregions determines $\mathbf{B}_{11}$ and $\mathbf{B}_{22}$, and the related Schur complement. The matrix blocks $\mathbf{B}_{ij}$ are a result of the application of a suitable numbering (permutation) of the degrees of freedom, and *not* a result of the additivity of the integration. That is, $\mathbf{B}_{ii}$ is *not* exactly the stiffness matrix related to Ω_i.

Let $s > 0$. For a domain Ω decomposed into s disjoint subregions Ω_i, the Schur complement is the matrix $\mathbf{S} := \mathbf{B}_{ss} - \sum_{i=1}^{s-1} \mathbf{B}_{si}\mathbf{B}_{ii}^{-1}\mathbf{B}_{is}$. For the remainder of this paper, let $\Omega = (-1, 1) \times (0, 1)$ and $\epsilon = 1$. In section 3, we calculate $\kappa_2(\mathbf{S})$ for the four cases of domain decomposition shown in figures 4 – 7.

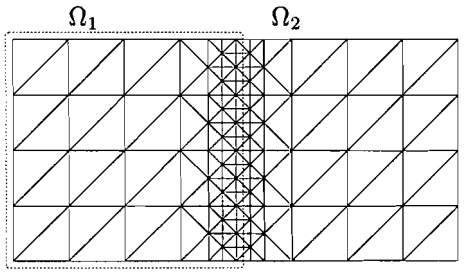


Figure 4: Two subdomains, s = 2.

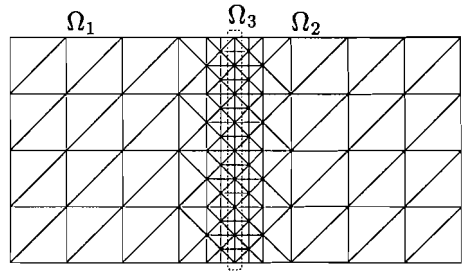


Figure 5: Three subdomains, s = 3.

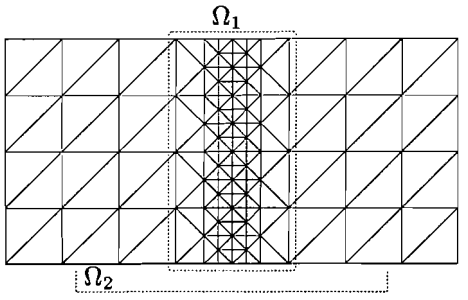


Figure 6: Two subdomains, s = 2.

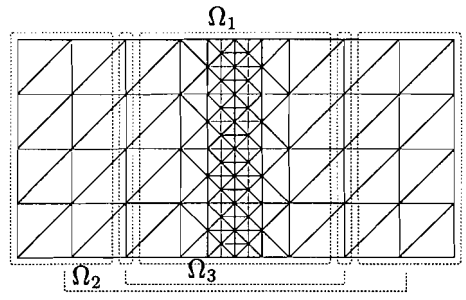
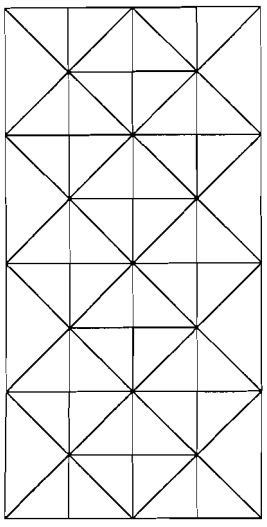


Figure 7: Three subdomains, s = 3.

All domain decompositions involve the same type of grid, refined around the center line, and contain either two of three subdomains.

For the subdomain which contains all refinement, we denote the amount of horizontal blocks (layers) with K, and called the related grids $G_{K,k}$.



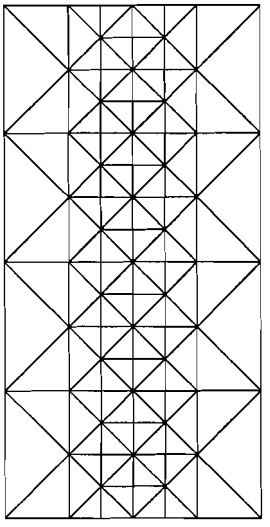
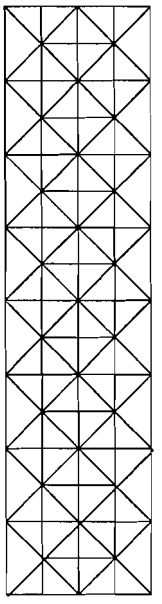


Figure 8: Refined grids $G_{K,k}$ for $K = 4$, $k = 1, 2$.



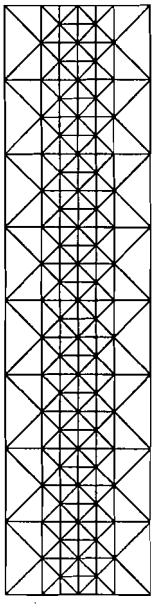


Figure 9: Refined grids $G_{K,k}$ for $K = 8$, $k = 1, 2$.

Let $k > 0$. Each horizontal block is obtained with the application of $2(k-1)$ levels of bisection refinement. Its connectivity-graph is called B_k. (Each two subsequent refinements shrink the smallest edge-length by a factor two.) Figure 8 and figure 9 show the refined grids related to $K = 4$ and $K = 8$ for both $k = 1$ and $k = 2$. With B_k, we associate the coarse grid element-size $H = 1/K$ en smallest element-size $h = H \cdot 2^{-k}$. The graphs B_k are defined in more detail in section 4.

3 The spectral condition number of the Schur complement

The Schur complement's spectral condition number is calculated from its extremal eigenvalues. Because numerical spectra computations are expensive, problem sizes are restricted. Nevertheless, the numerical results in table 1 indicate a clear trend.

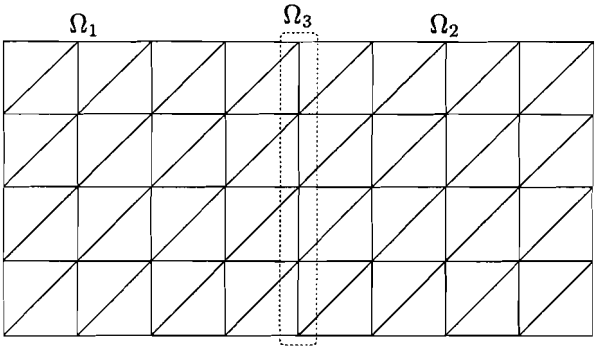


Figure 10: The standard domain decomposition for a regular grid

For the standard decomposed uniform grid in figure 10, where the Schur complement is related to the degrees of freedom situated at the center-line, [12] shows that $\kappa_2(\mathbf{S}) = O(h^{-1})$. It follows that the condition number only depends on the aspect ratios related to the grid which covers the Schur complement's subdomain.

Our numerical results for the refined case are similar: Also in these refined cases, the Schur complement's condition number $\kappa_2(\mathbf{S})$ only depends on the aspect ratios related to the grid which covers the Schur complement's subdomain.

The orders of $\kappa_2(\mathbf{S})$, estimated from table 1, are presented in table 1.

H	h	Fig. 4	Fig. 5	Fig. 6	Fig. 7
2^{-2}	2^{-3}	11.69	6.72	7.55	4.14
2^{-2}	2^{-4}	24.99	13.68	7.55	4.14
2^{-2}	2^{-5}	54.24	27.44	7.55	4.14
2^{-2}	2^{-6}	115.68	54.91	7.55	4.13
2^{-2}	2^{-7}	242.14	109.83	7.55	4.13
2^{-3}	2^{-4}	39.00	14.03	36.07	8.20
2^{-3}	2^{-5}	68.69	28.14	36.08	8.13
2^{-3}	2^{-6}	138.59	56.31	36.08	8.13
2^{-3}	2^{-7}	285.07	112.64	36.08	8.13
2^{-4}	2^{-5}	138.94	28.53	155.36	19.29
2^{-4}	2^{-6}	188.21	57.09	155.36	19.12
2^{-4}	2^{-7}	327.91	114.20	155.36	19.12

Fig. 4	Fig. 5	Fig. 6	Fig. 7
$O(H^{-2}h^{-1})$	$O(H^{-1}h^{-1})$	$O(H^{-2})$	$O(H^{-1})$

Table 1: The spectral condition number $\kappa_2(\mathbf{S})$, and derived order estimates

4 An efficient node ordering for the refinement along a line

This section calculates the amount of fill-in when Gaussian elimination is applied to the stiffness matrix in (5), on the grids in figure 8 and figure 9. The amount of fill-in is compared to the amount of entries in the original stiffness matrix (5), and is shown to be logarithmically dependent on the level of refinement. The corresponding amount of fill-in is compared to the amount of fill-in resulting from a the symmetric minimum degree ordering, as implemented in Matlab. The convective case is not examined: The induced extra non-zero matrix entries have little or no effect on the order of the amount of fill-in.

The to be presented degree of freedom and amount(s) of fill-in counts make use of a certain substructure caused by the bisection refinement. First, the connectivity graphs B_k related to the horizontal layers in figures 8 and 9 are defined in more detail.

First, observe that the connectivity graphs 11 and 12 related to horizonal layers in figures 8 and 9 are mutually related. Figure 12 shows that B_{k+1} consists of two sub-graphs of type B_k, and of two identical caps C, one on each side. The caps are related to a grid which contains three triangles.

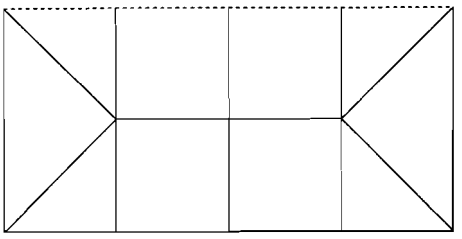


Figure 11: The connectivity graph related to $k = 1$: B_1

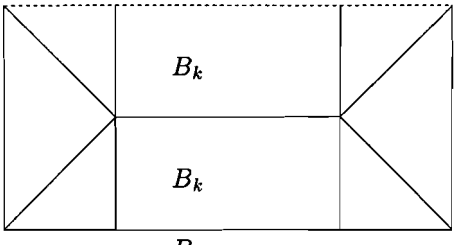


Figure 12: The connectivity graph of level $k + 1$: B_{k+1}

The edges of all non-directed graphs (see for instance 11) correspond to the non-zero stiffness matrix entries, and the vertices to degrees of freedom. Thus, the diagonal edges from the central division line are not present (because all related entries are zero). When counting degrees of freedom, we don't count the vertices situated at the top, and when counting fill-in, we neither count the fill-in caused by vertices at the top. Therefore, the related collection of edges at the top is drawn with dotted lines.

We first count the amount of degrees of freedom related to B_k.

Theorem 1. *Let N_k denote the amount of degrees of freedom related to B_k, excluding the degrees of freedom related to vertices situated at the top. Then*

$$N_k = 5 \cdot 2^k - 2. \tag{7}$$

Proof. First consider the case $k = 1$. According to figure 11, the amount of degrees of freedom is 8, which is equal to $5 \cdot 2^1 - 2$.

Next, assume that our claim holds for some $k > 0$. Assume that $N_k = 5 \cdot 2^k - 2$. Due to the construction of B_{k+1} in figure 12, N_{k+1} is double the amount of N_k, with an additional two degrees of freedom situated at the *bottom* corners. Thus,

$$N_{k+1} = 2(5 \cdot 2^k - 2) + 2 = 5 \cdot 2^{k+1} - 2.$$

□

For $K > 1$, K graphs B_k are superimposed. The related amount of degrees of freedom $N_{K,k}$ is $K \cdot N_k$ plus the amount of vertices on the lower bottom line of B_k, which is $3 + 2k$. Thus, the total degrees of freedom is:

$$N_{K,k} = K \cdot (5 \cdot 2^k - 2) + 3 + 2k.$$

Note that we pretend not to eliminate the vertices at the Dirichlet boundaries (top and bottom). The elimination would make the fill-in count below even more elaborate, but has little effect on the amount of fill-in. Thus, as a matter of fact, we count the fill-in of a stiffness matrix $\mathbf{A}$ induced by Neumann boundary conditions on the domain's top and bottom.

The Gaussian elimination process applied to $\mathbf{A}$ leads to the standard upper and lower triangular factors such that $\mathbf{A} = \mathbf{LU}$. We count the amount of potential non-zero entries E in $\mathbf{U}$, excluding its diagonal. The actual amount of fill-in follows from E and the amount of non-zero entries in $\mathbf{A}$. Because the grid depends on K and k, so do $\mathbf{A}$ and E, and are denoted with $\mathbf{A}_{K,k}$ and $E_{K,k}$ when necessary.

The amount of entries E depends on the numbering of the degrees of freedom (numbering of the vertices of the graphs B_k). We distinguish the following numbering schemes:

$$\begin{array}{ll} (LR) & \text{First from left to right, next from bottom to top;} \\ (BT) & \text{First from bottom to top, next from left to right.} \end{array} \tag{8}$$

The (LR) scheme is the one which leads to near optimal band-width *combined with* near optimal fill-in. The (BT) scheme performs poor with respect to both fill-in and band-width. The main reason that the (LR) numbering performs so well turns out to be that **the degrees of freedom are numbered first in a direction orthogonal to the resolved line, and next in the direction tangent to this line**. This numbering scheme shortens possible paths which can lead to the creation of fill-in.

George and Liu [6] have characterized paths which can cause fill-in. Let $i < k$ be two numbed degrees of freedom. A related (potential) non-zero in is created in U during Gaussian elimination if there is a *fill-in path*: A path from vertex v_i to vertex v_k through vertices $v_0, \ldots, v_{i-1}$ ([12], Theorem 3.3, page 139). All counts for E depend on the (LR) numbering. When graph B_k is related to a middle layer of a grid $G_{K,k}$, its fill-in paths can leave and reenter through its bottom horizontal line. Such paths are called *reentrant* paths, all other paths are called non-reentrant, or internal.

Definition 1. For $k > 0$, define

1. P_k, the amount of non-top vertices induced fill-in paths (inc. reentrant) of B_k;
2. T_k, the amount of top vertices induced fill-in paths of B_k;
3. R_k, the amount of bottom vertices reentrant fill-in paths of B_k;

Then

$$E_{K,k} = K \cdot P_k + T_k - R_k. \tag{9}$$

In theorem 2, which formulates $E_{K,k}$ for the (LR) scheme, several types of fill-in paths are distinguished:

- To the right, or diagonally or vertically up (also non-zero entry in $\mathbf{A}$);
- To the left, and next diagonally or vertically up;
- If possible, first down from v_i, next left, and finally up to a vertex on a horizontal line above v_i;
- If possible, first down from v_i, next right, and finally up to a vertex on a horizontal line above or containing v_i.

Here is how $E_{K,k}$ is a function of the fill-in paths related to B_k.

Theorem 2. *Let $\mathbf{A}_{K,k}$ be the standard stiffness matrix obtained on the grids $G_{K,k}$ in figures 8 - 9, using an (LR) numbering scheme. Assume that $\mathbf{A}_{K,k} = \mathbf{L}_{K,k}\mathbf{U}_{K,k}$ is factored with the use of Gaussian elimination. Then the amount of potential non-zero entries in the strictly upper triangular part of $\mathbf{U}_{K,k}$ is*

$$E_{K,k} = K \cdot \left((10k + 16) \cdot 2^k - 4k - 7\right) + 2(1 + k). \tag{10}$$

Proof. The refined grid consists of K horizontal layers, each related to a graph B_k. From the definition 9 of $E_{K,k}$ it follows that we have to count all of (1) P_k, (2) T_k and (3) R_k.

First, consider case (1). To determine P_k, we first count the non-zero entries related to the bottom corner vertices of B_k. This amount is:

$$4k + 7. \tag{11}$$

The proof makes use of induction with respect to the level k. Note that fill-in paths counted for P_k can be reentrant.

For $k = 1$, the left corner vertex v_k^l has 6 fill-in paths ending at higher numbered vertices, and the right corner vertex v_k^r has 5 such paths, two of which are depicted in figure 13

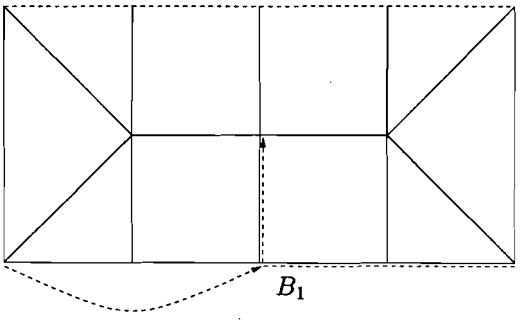


Figure 13: One fill-in path from v_1^l, and one from v_1^r.

Thus, for $k = 1$, the total amount of fill-in paths is:

$$6 + 5 = 2 \cdot 1 + 7$$

Next, assume that (11) holds for some $k \geq 1$. According to figure 12, B_{k+1} has two subgraphs of type B_k, and two cap graphs C, each introducing a new lower corner vertex v_{k+1}^l and v_{k+1}^r. Now v_{k+1}^l has a fill-in path to all vertices v_k^l had a fill-in path to, and v_{k+1}^r has a fill-in path to all vertices v_k^r had a fill-in path to. Each has two additional fill-paths with the two new top left and right vertices of graph B_{k+1}. So, the new bottom left and right corner vertices have a total of

$$4k + 7 + 2 + 2 = 4(k + 1) + 7$$

fill-in paths.

For case (1), we next assert that:

$$P_k = (10k + 16) \cdot 2^k - 4k - 7. \tag{12}$$

Again, the proof makes use of induction with respect to k. First, let $k = 1$. The vertex-wise fill-in path count in figure 14:

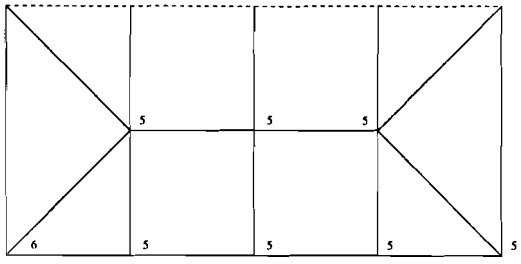


Figure 14: The amount of fill-in paths for each vertex in B_1.

shows that the total amount, for 8 vertices, is:

$$7 \cdot 5 + 1 \cdot 6 = (10 \cdot 1 + 16) \cdot 2^1 - 4 \cdot 1 - 7.$$

Now, suppose (12) holds for some $k \geq 1$. Recall, that B_{k+1} has two subgraphs of type B_k, and two cap graphs C (see figure 12). Thus, the total amount of fill-in paths is:

$2 \cdot (10k \cdot 2^k + 16 \cdot 2^k - 4k - 7)$	fill-in paths from for 2 graphs B_k (see (7));
$2 \cdot 2 \cdot (5 \cdot 2^k - 2)$	2 extra fill-in paths for 2 graphs B_k;
$4(k+1) + 7$	fill-in paths for the new bottom corners (see (11)).

As desired, this adds up to $(10(k+1) + 16) \cdot 2^{k+1} - 4(k+1) - 7$.

Next, consider case (2). Here we count T_p, the amount of fill-in paths related to top vertices of B_k:

$$T_p = (k+1)(2k+3). \tag{13}$$

In order to see that this holds, first look at B_1. Figure 15

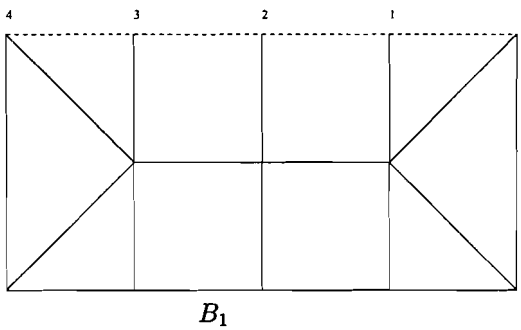


Figure 15: The amount of extra paths for each individual vertex at the top row of the top block B_1.

shows the amount of extra paths for each individual vertex at the top. The amount is

$$4 + 3 + 2 + 1 + 0 = 10 = (1+1)(2 \cdot 1 + 3).$$

Next, assume (13) holds for certain $k \geq 1$. Note that the amount of vertices situated at the top horizonal line (also at bottom horizontal line) of B_k is:

$$2k + 3. \tag{14}$$

Then the fill-in paths from the vertices at the top of B_{k+1} are:

$(k+1)(2k+3)$	from top vertices of upper B_k, inside B_k (see (13));
$2k+3$	1 extra from top vertices of upper B_k to new upper right corner(see (14));
$2(k+1)+3-1$	paths extra from new left vertex.

Again, this adds up to the desired result $((k+1)+1)(2(k+1)+3)$. Keeping (13) in mind, the total amount of fill-in paths in case (2) is:

$$(10k+16) \cdot 2^k - 4k - 7 + ((k+1)(2k+3)).$$

Finally, consider case (3). Here, we count the amount of reentrant fill-in paths R_k related to bottom vertices of graph B_k. To this end, we first count the total amount of paths related to bottom vertices, and next count and subtract the amount of internal paths.

The total amount of fill-in paths related to bottom vertices of B_k is:

$$4k^2 + 13k + 9. \tag{15}$$

For $k = 1$, this holds, counting all such paths in figure 11. Assume this amount is correct for certain $k > 0$. Then for $k+1$:

$4k^2+13k+9$	for each bottom vertex in bottom B_k (see 15);
$2 \cdot (2k+3)$	2 extra for each bottom vertex in bottom B_k (see (14));
$4(k+1)+7$	extra from two new vertices C graphs.

The total amount now is indeed $4(k+1)^2 + 13(k+1) + 9$. Next, we count the amount of internal paths:

$$2k^2 + 10k + 8. \tag{16}$$

For $k = 1$, this holds (see figure 11). Assume this amount is correct for certain $k > 0$. Then for $k+1$:

$2k^2+10k+8$	for each bottom vertex in bottom B_k (see 16);
$2k+3-1$	1 extra for each bottom vertex in bottom B_k, except right corner (see (14));
2	2 extra for right bottom corner vertex in B_k;
2	for the new left bottom corner vertex (in C graph);
$2k+3+2$	for the new right bottom corner vertex (in C graph).

From (16) and (15) we obtain:

$$R_k = (4k^2 + 13k + 9) - (2k^2 + 10k + 8) = 2k^2 + 3k + 1.$$

Finally, note that

$$T_k - R_k = (k+1)(2k+3) - (2k^2 + 3k + 1) = 2(1+k),$$

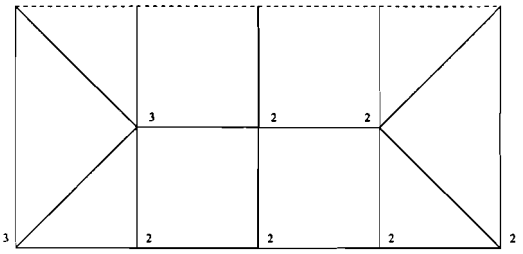
which shows that (10) holds. □

Remark 1. Theorem 2 shows that the amount of entries in the factors of $\mathbf{A}$ is of order $O(k \cdot 2^k)$. The amount of entries in $\mathbf{A}$ depends in a different manner on k:

Theorem 3. *Let $\mathbf{A}_{K,k}$ be the standard stiffness matrix obtained on the grids $G_{K,k}$ in figures 8 - 9. Then the amount of non-zero entries in $\mathbf{A}_{K,k}$ is:*

$$A_{K,k} = 25 \cdot 2^{k-1} - 7. \tag{17}$$

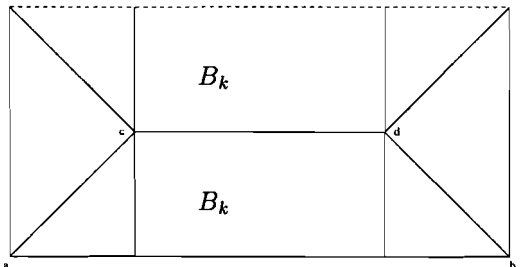
Proof. Also here, we use the fact that each graph B_{k+1} contains two B_k subgraphs, and cap graphs C, as in figures 11 and 12. For $k = 1$, figure 16 shows that $\mathbf{A}_{K,1}$ contains 18 non-zero entries.



B_1

Figure 16: Amount of non-zero entries of $\mathbf{A}_{K,1}$.

Next, assume (17) holds for a certain $k > 0$. Then, looking at figure 17



B_{k+1}

Figure 17: Amount of non-zero entries of $\mathbf{A}_{K,k+1}$.

all non-zero entries in $\mathbf{A}_{K,k+1}$ are:

$2(A_{K,k})$	one for each subgraph B_k;
3	for the new right bottom corner vertex a (in C graph).
2	for the new left bottom corner vertex b (in C graph);
$2 \cdot 1$	one each extra for vertices c and d.

This adds up to the desired amount for $k + 1$ in 17. □

Remark 2. Theorem 3 shows that the amount of entries in $\mathbf{A}_{K,k}$ is of order $O(2^k)$. Thus, using Gauss elimination for the (LR) scheme produces factors which are roughly k-times larger than the matrix itself. In the typical case, where the average amount of entries per row of $\mathbf{A}$ is small, a factor $k = 10$ or $k = 20$ is a

reasonable fill-in which in fact can compensate for $k = 10$ or $k = 20$ iterations, had an iterative method been used to solve $\mathbf{Ax} = \mathbf{b}$. Note, $k = 20$ corresponds to a difference in scales of 10^{-6}.

Figure 18 depicts the amount of non-zero entries in the **LU** factors for the (LR), (BT) and minimum degree schemes:

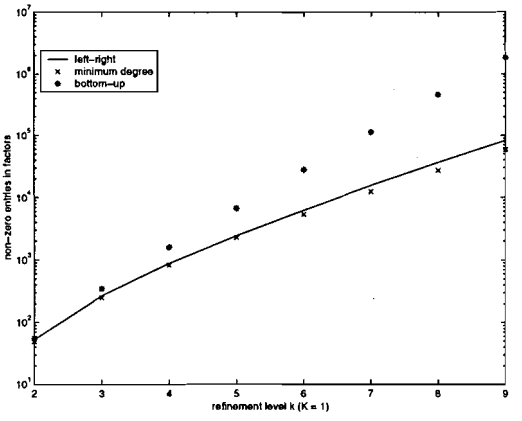


Figure 18: Fill-in for $K = 4$, $k = 0, \ldots 7$.

The figure clearly shows that the (LR) scheme fill-in is close to the minimum degree scheme's. Because an exact count of minimum degree scheme's fill-in is not available in the literature, we can not do comparisons for $k\infty$.

Figures 19 through 26 compare the band-width related to the (LR), minimum degree and reversed Cuthill-McKee schemes:

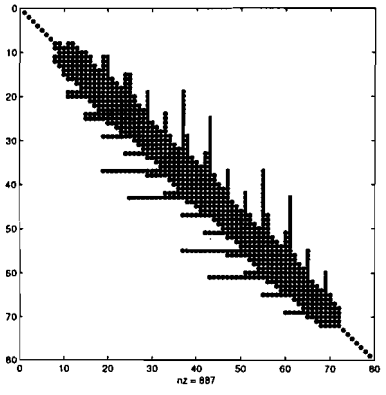


Figure 19: Factors left-right ordering $(K, k) = (4, 2)$.

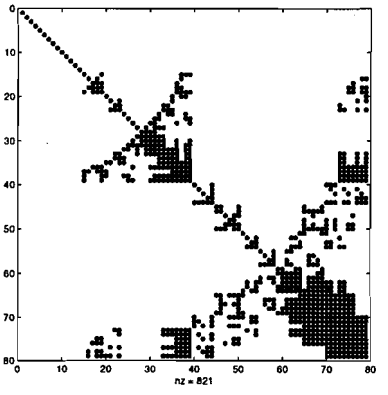


Figure 20: Factors minimum degree ordering $(K, k) = (4, 2)$.

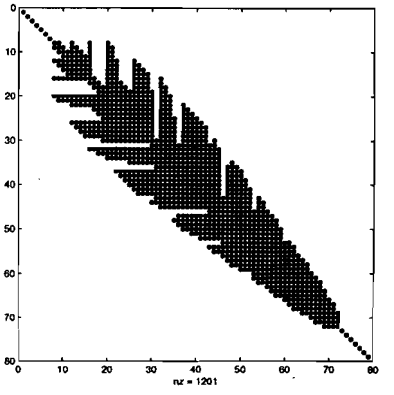


Figure 21: Factors rev. Cuthill-McKee ordering $(K, k) = (4, 2)$.

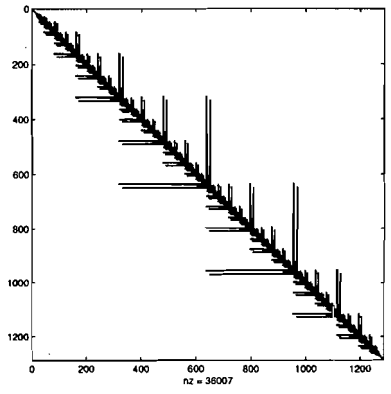


Figure 22: Factors left-right ordering $(K, k) = (4, 6)$.

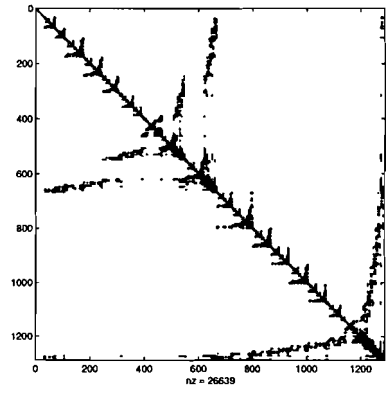


Figure 23: Factors minimum degree ordering $(K, k) = (4, 6)$.

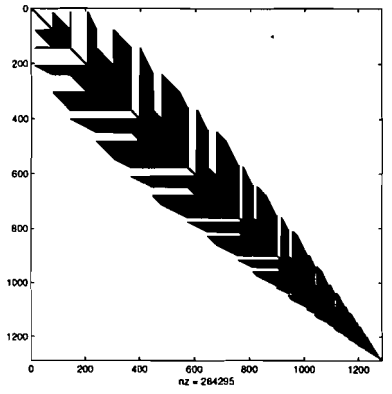


Figure 24: Factors rev. Cuthill-McKee ordering $(K, k) = (4, 6)$.

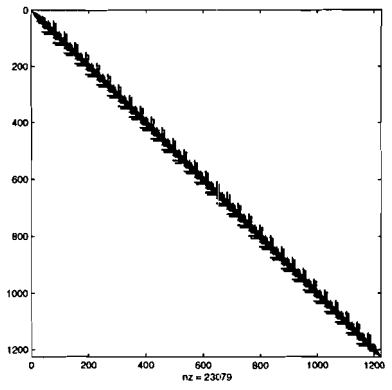


Figure 25: Factors left-right ordering $(K, k) = (32, 6)$.

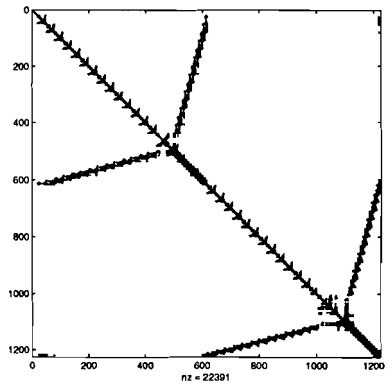


Figure 26: Factors minimum degree ordering $(K, k) = (32, 6)$.

As will be clear from these figure, the envelope of the factors of the reordered minimum degree matrix is almost maximal: Almost identical to the total amount of degrees of freedom. Thus using a minimum degree ordering, matrix-vector implementations for $\mathbf{L}$ and $\mathbf{U}$ must make use of column indexing. The $\mathbf{L}$ and $\mathbf{U}$ factors from the (LR) do not require such an indexing, whence matrix-vector multiplication is likely to be faster.

5 Conclusions

The presented qualitative analysis indicates that the Schur complement's condition number $\kappa_2(\mathbf{S})$ depends on the dimension of Ω_k, and on the aspect ratios inside this region. The situation appears to be similar to the one where a uniform grid is decomposed with the use of lines. Furthermore, we have shown that for a certain numbering of degrees of freedom exact Gaussian elimination leads to fill-in, logarithmic in the amount of levels of refinement. The factor's envelope is much smaller than in the case of a minimum degree based factorisation.

References

[1] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. I. *Numerische Mathematik*, 56:157–177, 1989.

[2] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. II. *SIAM Journal on Numerical Analysis*, 27:1569–1590, 1990.

[3] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55:1–22, 1990.

[4] H. J. P. Brocken. *Moisture transport in brick masonry: the grey area between bricks.* PhD thesis, Eindhoven University of Technology, Eindhoven, 1998.

[5] I. S. Duff and H. A. van der Vorst. Developments and trends in the parallel solution of linear systems. *Parallel Computing*, 25:1931–1970, 1999.

[6] A. George and J. Liu. *Computer solution of large sparse positive definite systems.* Prentice Hall, Englewood Cliffs, New Jersey, 1981.

[7] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Application*, 13:333–356, 1992.

[8] B. Joe. Construction of a k-dimensional delaunay triangulation using local transformations. *SIAM Journal on Scientific Computing*, 14:1415–1436, 1993.

[9] R. E. Kaasschieter. A general finite element preconditioning for the conjugate gradient mathod. *BIT*, 29:824–849, 1989.

[10] S. Margenov and J. Maubach. Optimal algebraic multilevel preconditioning for local refinement along a line. *Numerical Linear Algebra with Applications*, 2:347–361, 1995.

[11] J. Maubach. Local bisection refinement for n-simplicial grids generated by reflections. *SIAM Journal on Scientific Computing*, 16:210–227, 1995.

[12] G. A. Meurant. *Computer solution of the large linear systems.* Elsevier North Holland, Amsterdam, 1999.

[13] M. Magolu monga Made and H. A. van der Vorst. Paric: A family of parallel icomplete cholesky preconditioners. In M. Bubak, H. Afsarmanesh, R. Williams, and B. Hertzberger, editors, *High Performance Computing and Networking, 8th International Conference*, pages 89–98, Belin, 2000.

[14] Y. Saad and M. H. Schulz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[15] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of IEEE*, 55:1801–1809, 1967.

[16] H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13:631–644, 1992.