Symbolic Methods for the Element Preconditioning Technique *

U. Langer[1], S. Reitzinger[1], and J. Schicho[2]

[1] Institute of Computational Mathematics, University of Linz, Austria,
{ulanger,reitz}@numa.uni-linz.ac.at

[2] Research Institute of Symbolic Computation, University of Linz, Austria,
Josef.Schicho@risc.uni-linz.ac.at

Abstract. The method of element preconditioning requires the construction of an M-matrix which is as close as possible to a given symmetric positive definite matrix in the spectral sense. In this paper we give a symbolic solution of the arising optimization problem for various subclasses. This improves the performance of the resulting algorithm considerably.

1 Introduction

The condition number of the stiffness matrix K_h arising from the finite element discretization of second-order elliptic boundary value problems typically behaves like $O(h^{-2})$ as h tends to zero, where h denotes the usual discretization parameter characterizing a regular finite element discretization. This means practically, that the stiffness matrix has a large condition number on a fine grid. That is the reason why the classical iterative methods exhibit slow convergence This drawback can be avoided by multigrid methods (see, e.g. [6]) or multigrid preconditioned iterative methods [8, 9].

In contrast to the geometric multigrid method (MGM) that is based on a hierarchy of finer and finer meshes, the algebraic multigrid (AMG) method needs only single grid information, usually the matrix and the right-hand side of the system that is to be solved. In AMG, the hierarchy of coarser and coarser representation of the fine grid problem must be generated algebraically.

It is well known that an AMG method works well if K_h is an M-matrix (see Section 2.2 for the definition of M-matrices), but this is hardly the case in many real life applications. Thus, if K_h is not an M-matrix then it is desirable to derive an AMG preconditioner for K_h from a nearby, spectrally equivalent M-matrix B_h that is sometimes called regularizator [9]. In [5], we construct such an M-matrix regularizator. The main idea is to localize the problem: For each element, we compute an M-matrix which is as close as possible (in the spectral sense) to

* This work has been supported by the Austrian Science Fund under the grant 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' - under the grant SFB F013 'Numerical and Symbolic Scientific Computing' and under the project P 14953 'Robust Algebraic Multigrid Methods and their Parallelization'.

the element stiffness matrix. Then we assemble these M-matrices by the help of the element connectivity relations and get our regularizator B_h from which we afterwards derive the AMG-preconditioner (see Subsection 2.2 for details).

In [5], we solve these optimization problems numerically, by a sequential quadratic programming algorithm using a Quasi-Newton update formula for estimating the Hessian of the objective function. Unfortunately, this subtask turns out to be a bottleneck, because in some practically important cases we have to solve a large number of such optimization problems. Moreover, the numerical solution does not always get close to the global optimum, because there are several local optima, and on some of them the objective function is not even differentiable.

In this paper, we solve various cases of these optimization problems symbolically. For various element matrices involving only one symbolic parameter, we can find a closed form solution in terms of polynomials. A similar formula is given for general 2×2 matrices, but here there are several closed forms, and some inequalities need to be checked in order to determine which one has to taken. For general 3×3 matrices, a similar closed form would be theoretically possible, except that we need also square roots and – in one case – roots of higher degree polynomials (see Section 3). But such a closed formula would be too large to be useful, so we prefer to give a "formula" consisting of a program with arithmetic or square root (and in one case higher order root) assignments and **if then else** branches, but no loops. This is done in Section 3. Using these formula, we can compute the optimal preconditioner faster and more accurately (see Section 4).

2 Problem Formulation

In this section we explain the idea of the element preconditioning technique proposed in [5], and isolate the most crucial subproblem, namely the problem of the construction of M-matrices which are as close as possible to the finite element stiffness matrices in some spectral sense.

2.1 Finite Element Discretization

For simplicity, let us consider the weak formulation

$$\text{Find } u \in \mathbb{V}: \quad \int_\Omega (\nabla^t u \nabla v + \sigma u v)\, dx = \int_\Omega f v \, dx \quad \forall v \in \mathbb{V} \tag{1}$$

of the potential equation $-\Delta u + \sigma u = f$ in Ω under homogeneous Neumann conditions on the boundary $\partial\Omega$ of Ω as some model problem for explaining the element preconditioning technique. The computational domain $\Omega \subset \mathbb{R}^d$ (with $d = 2, 3$ the spatial dimension) is bounded with a sufficiently smooth boundary $\partial\Omega$. The test space $\mathbb{V}$ coincides with the Sobolev space $H^1(\Omega)$, $\sigma \geq 0$ denotes a real parameter and $f \in L_2(\Omega)$ is a given right-hand side. In order to ensure solvability of the boundary value problem (1) in the case $\sigma = 0$, we assume that f is L_2-orthogonal to all constants.

A finite element (FE) discretization of the computational domain Ω results in some FE-mesh $\mathcal{T}_h = \{\overline{\Omega}^r : r \in \tau_h\}$ (see Figure 1 for an FE-discretization of the unit square by rectangular and triangular elements) such that

$$\overline{\Omega} = \bigcup_{r \in \tau_h} \overline{\Omega}^r ,$$

with the index set τ_h of all finite elements, the set of all nodes $\{x_i : i \in \overline{\omega}_h\}$, the index set $\overline{\omega}_h$ of all nodes and the typical mesh size h. The FE-basis $\Phi =$

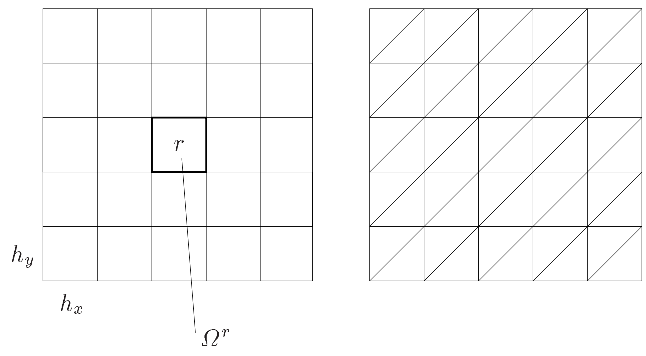

Fig. 1. Finite element discretizations of the unit square.

$\{\varphi^{[j]}(x) , j \in \overline{\omega}_h\}$ spans the FE-space $\mathbb{V}_h = \mathrm{span}\{\Phi\} \subset \mathbb{V}$ and changes (1) into the FE-scheme

$$\text{Find } u_h \in \mathbb{V}_h \ : \quad \int_\Omega (\nabla^t u_h \nabla v_h + \sigma u_h v_h)\, dx = \int_\Omega f v_h \, dx \quad \forall v_h \in \mathbb{V}_h \tag{2}$$

that is equivalent to the linear system of FE-equations

$$\text{Find } \underline{u}_h \in \mathbb{R}^{N_h} \ : \quad K_h \underline{u}_h = \underline{f}_h \quad \text{in} \ \ \mathbb{R}^{N_h} \tag{3}$$

for defining the coefficients $u_h^{[i]}$ in the FE-ansatz

$$u_h(x) = \sum_{i \in \overline{\omega}_h} u_h^{[i]} \cdot \varphi^{[i]}(x), \tag{4}$$

where $K_h \in \mathbb{R}^{N_h \times N_h}$ denotes the stiffness matrix, $\underline{f}_h \in \mathbb{R}^{N_h}$ denotes the load vector, and $\underline{u}_h = (u_h^{[i]}) \in \mathbb{R}^{N_h}$ is the solution vector of nodal unknowns $u_h^{[i]}$. We assume that the FE-basis functions are of Lagrangian type such that $\varphi^{[j]}(x_i) = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta. Since the FE-bases is assumed to have local support the corresponding integrals are evaluated locally on each element, i.e.,

$$\int_\Omega (\nabla^t u_h \nabla v_h + \sigma u_h \, v_h)\, dx = \sum_{r \in \tau_h} \int_{\Omega^r} (\nabla^t u_h \nabla v_h + \sigma u_h \, v_h)\, dx \, .$$

By using the FE-ansatz (4) we get, on each element, an integral of the form

$$(K_h^r)_{ji} = \int_{\Omega^r} (\nabla^t \varphi^{[i]} \nabla \varphi^{[j]} + \sigma \varphi^{[i]} \varphi^{[j]})\, dx$$

for calculating the coefficients $(K_h^r)_{ji}$ of the element stiffness matrix K_h^r. In the case of n_r bases functions on element $\overline{\Omega}^r$ ($r \in \tau_h$), we arrive at an $K_h^r \in \mathbb{R}^{n_r \times n_r}$ that is symmetric and positive (semi)definite. Then the global stiffness matrix $K_h \in \mathbb{R}^{N_h \times N_h}$ will be assembled from the local element stiffness matrix by the usual assembling procedure that can be represented in the form

$$K_h = \sum_{r \in \tau_h} C_r^t K_h^r C_r \,, \tag{5}$$

where the matrices $C_r \in \mathbb{R}^{n_r \times N_h}$ denote the so-called element connectivity matrices.

2.2 Element Preconditioning

The condition number of some regular matrix $A \in \mathbb{R}^{n \times n}$ is defined by

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|,$$

where $\|\cdot\|$ is an appropriate matrix norm. For an SPD matrix A, the so-called spectral condition number

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$$

is based on the spectral norm $\|A\| = \sqrt{\lambda_{\max}(A^t A)} = \lambda_{\max}(A)$, where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denotes the maximal and minimal eigenvalue of the matrix A, respectively. The matrix A^t denotes the transpose of A.

The SPD matrices A, $B \in \mathbb{R}^{n \times n}$ are called spectrally equivalent if

$$\exists c_1, c_2 \in \mathbb{R}^+ : c_1 \cdot \langle Bu, u \rangle \le \langle Au, u \rangle \le c_2 \cdot \langle Bu, u \rangle \qquad \forall u \in \mathbb{R}^n \tag{6}$$

which is briefly denoted by $c_1 \cdot B \le A \le c_2 \cdot B$. Therein $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. In addition, $\langle \cdot, \cdot \rangle_A$ denotes the A-energy inner product corresponding to the SPD matrix A, i.e., $\langle \cdot, \cdot \rangle_A = \langle A \cdot, \cdot \rangle$. Obviously the best possible constants c_1 and c_2 are given by the solution of the generalized eigenvalue problem

$$Au = \lambda Bu$$

with $c_1 = \lambda_{\min}(B^{-1/2} A B^{-1/2})$ and $c_2 = \lambda_{\max}(B^{-1/2} A B^{-1/2})$.

An important subclass of regular matrices are M-matrices. For SPD matrices, this class is defined as follows:

$$M_n = \left\{ A \in \mathbb{R}^{n \times n} \ : \ a_{ii} > 0,\, a_{ij} \le 0, i \ne j, \sum_{j=1}^{n} a_{ij} \ge 0 \right\} .$$

A crucial point in the FE-simulation is the solution of the linear, symmetric and positive semidefinite (positive definite in the case $\sigma > 0$) system of equations (3). It is well known that an AMG method works well if $K_h \in M_{N_h}$, but this is hardly the case in many real life applications. Thus, it is desirable to derive an AMG preconditioner for K_h from a nearby, spectrally equivalent matrix B_h that is sometimes called regularizator [9]. Especially, in our case we need an M-matrix for applying the standard AMG efficiently. Consequently, if we are able to construct such a symmetric, positive definite regularizator B_h in the class of the M-matrices, then we can derive a good preconditioner C_h for K_h by applying a symmetric AMG cycle to B_h instead of K_h. Finally, the constructed preconditioner C_h is used in a preconditioned conjugate gradient solver. This approach was presented in [5].

In order to be able to construct an M-matrix B_h efficiently we have to localize the problem. The basic idea is the content of the following lemma.

Lemma 1. *Let the stiffness matrix $K_h \in \mathbb{R}^{N_h \times N_h}$ be SPD, and K_h be assembled from SPD element matrices $K_h^r \in \mathbb{R}^{n_r \times n_r}$, $r \in \tau_h$, i.e., K_h can be represented in the form (5). Further, let us suppose that, for all $r \in \tau_h$, there are SPD matrices $B_h^r \in M_{n_r}$ such that the spectral equivalence inequalities*

$$c_1^r \cdot B_h^r \leq K_h^r \leq c_2^r \cdot B_h^r \qquad \forall r \in \tau_h \tag{7}$$

hold, with h-independent, positive spectral equivalence constants c_1^r and c_2^r. Then the matrix

$$B_h = \sum_{r \in \tau_h} C_r^t B_h^r C_r \tag{8}$$

is spectrally equivalent to the stiffness matrix K_h, i.e.,

$$c_1 \cdot B_h \leq K_h \leq c_2 \cdot B_h, \tag{9}$$

with the spectral equivalence constants

$$c_1 = \min_{r \in \tau_h} \{c_1^r\} \ \ \textit{and} \ \ c_2 = \max_{r \in \tau_h} \{c_2^r\}.$$

Additionally, the matrix B_h is SPD and belongs to the class M_{N_h}.

Proof. see [5].

Our particular interest in this paper consists in the construction of such an SPD M-matrices B_h^r that are as close as possible to K_h^r in the spectral sense. Thus, Lemma 1 provides the theoretical background for Algorithm 1. This algorithm returns the best SPD matrix $B_h \in M_{N_h}$ in the sense of the localized problem.

Remark 1.

1. We note that in the 2D case $n_r = 3$ and $n_r = 4$ correspond to linear and bilinear elements, respectively. Similarly, linear and trilinear elements for the 3D case are represented by $n_r = 4$ and $n_r = 8$, respectively.

Algorithm 1 GeneralSpectralMatrix ()

```
for all r ∈ τ_h do
  Get the element matrix K_h^r ∈ ℝ^{n_r × n_r}
  if K_h^r ∉ M_{n_r} then
    Calculate B_h^r from the restricted minimization problem

      λ_max((B_h^r)^{-1/2} K_h^r (B_h^r)^{-1/2}) / λ_min((B_h^r)^{-1/2} K_h^r (B_h^r)^{-1/2}) → min

    subject to B_h^r ∈ M_{n_r} and B_h^r is SPD
  else
    Set B_h^r = K_h^r
  end if
  Assemble B_h^r
  Assemble K_h^r
end for
```

2. In the case of symmetric positive semidefinite element matrices K_h^r the technique applies again. Now the generalized spectral condition number has to be minimized, i.e., the eigenvalue $\lambda_{\min}((B_h^r)^{-1/2}K_h^r(B_h^r)^{-1/2})$ is replaced by $\lambda_{\min} = \min\{\lambda((B_h^r)^{-1/2}K_h^r(B_h^r)^{-1/2}),\ \lambda \neq 0\}$.
3. For our special case $\sigma = 0$, the element stiffness matrices are symmetric positive semidefinite. But such K_h^r can be transformed to an SPD one by eliminating the last row and column (kernel elimination), see [5].

For the rest of the paper we skip the indices of the element stiffness matrices, i.e. $K = K_h^r$ whenever no ambiguities can occur. In the following we give three typical examples from the FE-discretization where the M-matrix property is lost.

Example 1. We study the case of anisotropic rectangular and triangular elements (see Figure 2) where we can establish the dependency of the condition number on the anisotropic parameter q explicitly. Further we always use the variational form (1) for our examples.

1. Let us consider the case of an anisotropic rectangular element with bilinear FE-functions and set $\sigma = 0$. The element stiffness matrix has the form
$$K = \frac{1}{6q}\begin{pmatrix} 2+2q^2 & 1-2q^2 & -2+q^2 & -1-q^2 \\ 1-2q^2 & 2+2q^2 & -1-q^2 & -2+q^2 \\ -2+q^2 & -1-q^2 & 2+2q^2 & 1-2q^2 \\ -1-q^2 & -2+q^2 & 1-2q^2 & 2+2q^2 \end{pmatrix}. \tag{10}$$
After eliminating the last row and column (i.e. kernel elimination, see Remark 1.3), the element stiffness matrix $\tilde{K} \notin M_3$ for $0 < q < \sqrt{1/2}$.
2. The second example is due to the triangle with $\sigma = 0$. In this case the element stiffness matrix has the form
$$K = \frac{1}{4q}\begin{pmatrix} 1+q^2 & 1-q^2 & -2 \\ 1-q^2 & 1+q^2 & -2 \\ -2 & -2 & 4 \end{pmatrix}. \tag{11}$$

Again after a proper reduction to the SPD case $\tilde{K} \notin M_2$ for $0 < q < 1$.

3. As a third example we consider our model bilinearform with $\sigma = 1$ and calculate the element stiffness matrix on the triangle, where some mass lumping is used, i.e.,
$$K = \frac{1}{4q} \begin{pmatrix} 1+2q^2 & 1-q^2 & -2 \\ 1-q^2 & 1+2q^2 & -2 \\ -2 & -2 & 4+q^2 \end{pmatrix}. \tag{12}$$
Again, $K \notin M_3$ for $0 < q < 1$.

Fig. 2. Thin finite element structures.

In the case of higher-order ansatz functions that are very important in practical application, we can not expect to get M-matrices at all.

2.3 An Optimization Problem

The critical step in Algorithm 1 is the solution of the restricted minimization problem: Given an SPD matrix K, find an SPD M-matrix B such that the condition number of $B^{-1/2}KB^{-1/2}$ is as small as possible. In this situation, we say that B is the closest M-matrix to K in the spectral sense.

We need to solve many instances of the problem, in general, one for each element. If the dimension of the matrices is equal to n, then we have an optimization problem with $\frac{(n+1)(n-2)}{2}$ variables (for the coefficients of B, up to scalar multiplication) depending on the same number of parameters (for the coefficients of K, up to scalar multiplication). The number n is relatively small, i.e. $n \leq 30$; in typical applications (i.e. low-order finite elements), we have $n = 2, 3, 4$ (see Example 1).

The search space is a convex polyhedron in projective space, which is independent of the parameters. We can reformulate the problem in such a way that the objective function is independent of the parameters (of course, this makes the search space vary).

We write $K = A^t A$ by the Cholesky decomposition. Then the inverse of the matrix $B^{-1/2}KB^{-1/2}$ is similar to
$$C := AB^{-1/2}(B^{1/2}K^{-1}B^{1/2})B^{1/2}A^{-1} = (A^t)^{-1}BA^{-1}.$$

Finding the closest M-Matrix B is equivalent to finding an SPD matrix C such that A^tCA is an M-Matrix (equal to B), which has smallest possible condition number.

In the reformulation, the search space is described by linear inequalities in the coefficients of C. More precisely, the inequalities are of the type

$$\langle a_i, a_j \rangle_C \leq 0 \text{ for } 1 \leq i < j \leq n+1,$$

where $a_1, \ldots, a_n$ are the columns of A, and $a_{n+1} := -a_1 - \cdots - a_n$, and $\langle u, v \rangle_C = \langle u, Cv \rangle$ is again the scalar product defined by the SPD matrix C. This follows from the fact that $\langle a_i, a_j \rangle_C$ is the (i,j)-th entry of the matrix A^tCA for $i, j \leq n$, and that $\langle a_i, -a_{n+1} \rangle_C$ is the i-th row sum. The search space depends on the vectors $a_1, \ldots, a_{n+1}$ in a symmetric way. The vectors $a_1, \ldots, a_{n+1}$ fulfill the symmetric condition $a_1 + \cdots + a_{n+1} = 0$ and the restriction that each n of them are linearly independent.

In [5], the restricted optimization problem is solved by a sequential quadratic programming algorithm using a Quasi-Newton update formula for estimating the Hessian of the objective. An additional difficulty is the fact that the objective function is not everywhere differentiable: The gradient does not exist for matrices with multiple maximal or minimal eigenvalue. This is a subset of measure zero, but unfortunately it contains the optimum in some cases, as experiments have shown.

3 A Symbolic Solution of the Optimization Problem

As the restricted problem is strictly algebraic, we can approach it by general quantifier elimination methods such as the method of Gröbner bases [1, 2] or the method of cylindrical algebraic decomposition [3, 7, 4]. For $n = 2$, this indeed gives a formula for the solution (see Remark 2 below). For $n = 3$ or higher, the number of variables is too large for such an approach to work. It is therefore necessary to exploit the specific structure of the problem.

3.1 Cases where the objective function is differentiable

In the space of all SPD matrices modulo scalar multiplication, the objective function has only one local minimum, namely the identity matrix I. If the optimum is assumed in the interior of the search space $\Sigma(a_1, \ldots, a_{n+1})$, then this optimum must be equal to this local optimum. Clearly, this happens only if the given matrix K is already an M-matrix.

In the other case, the optimum is assumed on the boundary. We like to distinguish cases specifying on which face the boundary is assumed. In order to do this in a convenient way, we introduce some terminology.

Let C be a point on the boundary of $\Sigma(a_1, \ldots, a_{n+1})$ (or Σ for short). Then there is a unique face F such that C is contained in the interior of of F, where the interior of a face is defined as the face minus the union of all subfaces. We call this face $F(C)$. The linear subspace carrying $F(C)$ is denoted by $E(C)$. It is defined by equalities $\langle a_i, a_j\rangle_C = 0$ for a set of index pairs (i, j), $i \neq j$; this set is denoted by $\pi(C)$. If we replace each defining equation by its corresponding inequality $\langle a_i, a_j\rangle_C \leq 0$, then we get a convex cone, which we denote by $U(C)$.

Example 2. Assume that C lies on the interior of the maximal face (also called facet) defined by vectors a_1, a_2. Then we have

- $E(C) = \{C \mid \langle a_1, a_2\rangle_C = 0\}$;
- $F(C) = E(C) \cap \Sigma$;
- $U(C) = \{C \mid \langle a_1, a_2\rangle_C \leq 0\}$;
- $\pi(C) = \{(1, 2)\}$.

In order to find the optimum in the case where it lies in the interior of a face F, we restrict the objective function to the corresponding linear subspace E and study the local minima.

Assume that E is a hyperplane, defined by $\langle a_i, a_j\rangle_C = 0$, where $\langle a_i, a_j\rangle > 0$. (If $\langle a_i, a_j\rangle \leq 0$, then the unity matrix is the global optimum of the half space.) Let $\overline{a_i}$, $\overline{a_j}$ be the normalizations of a_i, a_j to unit length. We set $s := \overline{a_i} + \overline{a_j}$ and $d := \overline{a_i} - \overline{a_j}$. For any matrix C in E, we have $\langle s, s\rangle_C = \langle d, d\rangle_C$, and therefore

$$\mathrm{cond}(C) = \frac{\max_{\|u\|=1}\langle u, u\rangle_C}{\min_{\|v\|=1}\langle v, v\rangle_C} \geq \frac{\langle d, d\rangle_C/\|d\|^2}{\langle s, s\rangle_C/\|s\|^2} = \frac{\|s\|^2}{\|d\|^2}.$$

Equality is assumed iff s and d are eigenvectors to the eigenvalues $\frac{\|s\|^2}{\|d\|^2}\lambda$ and λ, respectively, for some scaling factor λ, and all other eigenvalues lie between these two values. Since s and d are orthogonal, this is indeed possible. The set of all matrices satisfying these conditions is denoted by $\Pi(a_i, a_j)$.

Remark 2. In the case $n = 2$, $\Pi(a_i, a_j)$ has exactly one element (up to scaling). This element is easy to compute. Moreover, the hyperplanes are the only possible faces, because any matrix in a lower-dimensional subspace is already singular. Carrying out the computation, one obtains the following formula for the closest M-matrix to $K = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$:

1. If $b > 0$ then $B = \begin{pmatrix} a & 0 \\ 0 & c \end{pmatrix}$.
2. If $a + b < 0$, then $B = \begin{pmatrix} a & -a \\ -a & 2a + 2b + c \end{pmatrix}$.
3. If $b + c < 0$, then $B = \begin{pmatrix} a + 2b + 2c & -c \\ -c & c \end{pmatrix}$.
4. Otherwise, K is already an M-matrix, and $B = K$.

The same result can also be obtained by the general method of Gröbner bases mentioned above.

The following lemma is useful because it allows to reduce other cases to the hyperplane case.

Lemma 2. *Let C_0 be a boundary point of Σ. Assume that the objective function* cond*, restricted to $U(C_0)$, has a local minimum at C_0. Assume that the maximal and the minimal eigenvalue of C_0 are simple. Then C_0 is a global minimum. Moreover, there exist two vectors $a, b \in \mathbb{R}^n$, such that*

(a) $E(C_0)$ *is contained in the set* $\{C \mid \langle a, b\rangle_C = 0\}$.
(b) $U(C_0)$ *is contained in the set* $\{C \mid \langle a, b\rangle_C \leq 0\}$.
(c) $\cos(\measuredangle(a, b)) = \frac{\mathrm{cond}(C_0)-1}{\mathrm{cond}(C_0)+1}$.

We call the pair of vectors (a, b) the "guard" of C_0.

Proof. Let u and v be the normalized eigenvectors to the eigenvalues λ_{max} and λ_{min} of C_0. The gradient of the cond function at C_0 is equal to (see [5])

$$(\mathrm{grad\ cond})_{C_0} = \left(\frac{\lambda_{min} u_i u_j - \lambda_{max} v_i v_j}{\lambda_{min}^2}\right)_{i,j}.$$

Since C_0 is a local minimum in $E(C_0)$, the gradient must be orthogonal to $E(C_0)$, or equivalently

$$\lambda_{min}\langle u, u\rangle_C - \lambda_{max}\langle v, v\rangle_C = 0$$

for all $C \in E(C_0)$. With

$$a := \sqrt{\lambda_{min}}u + \sqrt{\lambda_{max}}v, \quad b := -\sqrt{\lambda_{min}}u + \sqrt{\lambda_{max}}v,$$

we obtain (a).

Since C_0 is also a local minimum in $U(C_0)$, the gradient must have nonnegative scalar product with all vectors from C_0 into $U(C_0)$. Equivalently, we can write

$$\lambda_{min}\langle u, u\rangle_C - \lambda_{max}\langle v, v\rangle_C \geq 0$$

for all $C \in U(C_0)$, and (b) follows.

Since u and v are orthogonal, the equation (c) follows by a straightforward computation. Finally, for all $C \in U(C_0)$ we have

$$\mathrm{cond}(C) \geq \frac{\langle u, u\rangle_C}{\langle v, v\rangle_C} \geq \frac{\lambda_{max}}{\lambda_{min}} \tag{13}$$

by the above equation. For C_0, equality holds in (13) and hence the minimum is global.

Let π be a set of pairs of indices, and let F, E and U be the corresponding face, linear subspace, and convex cone. If the optimum lies in the interior of F, then the guard must satisfy the conditions (a) and (b) of Lemma 2. Therefore, we call any pair (a, b) of vectors satisfying (b), (c), and $\langle a, b\rangle > 0$ – this is a consequence of (c) in Lemma 2 – a "possible guard" for π.

The set of all possible guards may be infinite. (We consider two guards as different if they do not arise one from the other by scaling or by switching the vectors. Note that we can scale both factors independently with positive factors, and the whole pair with -1.) For instance, let $\pi := \{(1,2),(1,3)\}$, and assume that $\langle a_1, a_2\rangle > 0$ and $\langle a_1, a_3\rangle > 0$. Then $(a_1, \lambda a_2 + \mu a_3)$ is a possible guard for any $\lambda > 0, \mu > 0$.

Let (a, b) a possible guard. Then we can show that $\mathrm{cond}(C) \geq \frac{1+\cos(\measuredangle(a,b))}{1-\cos(\measuredangle(a,b))}$ for any $C \in U$, as in the proof for the hyperplane case above. Therefore each possible guard gives a lower bound for the objective function. If (a, b) are guards for the optimum C_0, then this lower bound is assumed for C_0. If (a', b') is another guard with $\measuredangle(a', b') > \measuredangle(a, b)$, then (a', b') gives a lower bound which cannot be achieved inside U, because it is smaller than the condition number of the global optimum. Therefore, we only need to consider the possible guards enclosing the smallest possible angle.

For the above example, we will see later that the set of all possible guards is equal to $\{(a_1, \lambda a_2 + \mu a_3) \mid \lambda \geq 0, \mu \geq 0, \lambda \text{ or } \mu > 0\}$. If the orthogonal projection b of a_1 into the plane spanned by a_2, a_3 is a positive linear combination of a_1, a_2, then (a_1, b) is the positive guard enclosing the smallest possible angle.

Algorithm 2 OptimizeByGuards (n=3)

```
for all sets π of pairs of indices do
  compute the possible guard (a, b) enclosing the smallest possible angle
  if Π(a, b) ∩ Σ ≠ ∅ then
    return a matrix in Π(a, b) ∩ Σ
    exit
  end if
end for
return failure
```

In the case $n = 3$, we can compute the possible guard enclosing the smallest possible angle by sorting out cases, as shown below. For each guard (a, b), the set $\Pi(a, b)$ of possible optima is a line segment. It can be represented by its two deliminating points, which can be computed easily. Intersecting a line segment with a polyhedron given by linear inequalities is again easy: We only need to evaluate the left hand side at the two deliminating points, check the signs, and compute the linear combination giving a zero value if the two signs are different. Using Algorithm 2, we can compute the global optimum assuming that its largest and smallest eigenvalue are simple, i.e. that the objective function is differentiable at this point.

In the following case by case analysis, we make the following global assumptions: The indices i, j, k, l are pairwise different. We assume that (u, v) is a possible guard. The vectors u, v are expressed in the basis a_i, a_j, a_k: $u = u_1 a_i + u_2 a_j + u_3 a_k$, $v = v_1 a_i + v_2 a_j + v_3 a_k$. We define positive semi-definite matrices $C_1 := a_i a_i^t$, $C_2 := a_j a_j^t$, $C_3 := a_k a_k^t$, $C_4 := (a_i - a_j)(a_i - a_j)^t$, $C_5 := (a_j - a_k)(a_j - a_k)^t$, $C_6 := (a_i - a_k)(a_i - a_k)^t$. Note that if one of these matrices C_r lies in E, (resp. U), then it must also satisfy $\langle u, v\rangle_{C_r} =$ (resp. $\leq$) 0, by condition (a) and (b) of Lemma 2 and by continuity.

Pairs: $\pi = \{(i,j)\}$. Obviously, the only possible guard is (a_i, a_j).

3-chains: $\pi = \{(i,j),(j,k)\}$. Since C_1, C_2, C_3, C_6 are in E, we get

$$u_1v_1 = u_2v_2 = u_3v_3 = (u_1 - u_3)(v_1 - v_3) = 0.$$

Up to scaling and switching, the general solution is

$$u_1 = u_3 = v_2 = 0, u_2 = 1, v_1 = \lambda, v_3 = \mu.$$

Since C_4, C_5 are in U, we get $\lambda \geq 0$, $\mu \geq 0$. The smallest possible angle is enclosed for choosing v to be the projection of $u = a_1$ to the plane spanned by a_2, a_3, if this projection is a positive linear combination, or by choosing $v = a_2$ or a_3 otherwise.

Double pairs: $\pi = \{(i,j),(k,l)\}$. Since C_1, C_2, C_5, C_6 are in E, we get

$$u_1v_1 = u_2v_2 = (u_1 - u_3)(v_1 - v_3) = (u_2 - u_3)(v_2 = v_3) = 0.$$

Up to scaling and switching, we get the following three solutions:

$$u = (1,0,0), v = \pm(0,1,0);$$
$$u = (1,0,1), v = \pm(0,1,1);$$
$$u = (1,0,0), v = \pm(1,1,1).$$

Since C_3, C_4 are in U, we can discard the second solution. Therefore, we have precisely two possible pairs, namely (a_i, a_j) and (a_k, a_l).

Triangles: $\pi = \{(i,j),(j,k),(i,k)\}$. Since C_1, C_2, C_3 are in E, we get

$$u_1v_1 = u_2v_2 = u_3v_3 = 0.$$

All solutions have already appeared as possible pairs of a 3-chain. Therefore, the set of possible pairs is equal to the union of the sets of possible pairs of the three 3-chains contained in π.

4-chains: $\pi = \{(i,j),(j,k),(k,l)\}$. Since C_1, C_2, C_6 are in E, we get

$$u_1v_1 = u_2v_2 = (u_1 - u_3)(v_1 - v_3) = 0.$$

The are three parameterized solutions:

$$u = (1,0,1), v = (0,\lambda,\mu);$$
$$u = (0,1,0), v = (\lambda,0,\mu);$$
$$u = (0,0,1), v = (\lambda,\mu,\lambda).$$

The second and the third have already appeared as possible guards for the two 3-chains contained in π. The first is new. Since C_3, C_4, C_6 are in U, we get the additional restrictions $\lambda \leq 0$, $\mu \geq 0$. The smallest possible angle is enclosed for choosing v to be the projection of $u = a_1 + a_2 = -a_3 - a_4$ to the plane spanned by a_2, a_3, if this projection is a linear combination with coefficients $\lambda \leq 0$, and $\mu \geq 0$.

Other: It is not possible to have an a_i which is orthogonal to all a_j, $j \neq i$ – these other a_j form a basis of $\mathbb{R}^3$ and a_i, being different from zero, cannot be orthogonal to the whole $\mathbb{R}^3$. We also cannot have a 4-cycle, because this leads to the contradiction

$$\langle a_i + a_k, a_i + a_k \rangle_C = -\langle a_i + a_k, a_j + a_l \rangle_C = 0.$$

Therefore there are no other cases.

Remark 3. For $n \geq 4$, a similar case by case analysis is possible, and we can determine the possible guards enclosing a minimal angle. It is less obvious to determine whether $\Pi(u, v) \cap \Sigma \neq \emptyset$ in this case. But if $\Pi(u, v)$ has a non-empty intersection with the border of Σ, then there is a superset of index pairs such that an optimum can be found on the corresponding smaller linear subspace. Thus, it suffices to do this check assuming that the $\Pi(u, v)$ has empty intersection with the border, and this is easy: Take a single element and check whether it is contained in Σ.

For $n = 3$, it is easier to do the full completeness check, because in this way we can omit testing possible pairs that are also possible pairs of a subset of pairs of indices.

3.2 Cases where the objective function is not differentiable

If there is no optimum C with largest and smallest eigenvalue being simple, then Lemma 2 does not help in finding the optimum. Our strategy is to restrict the objective function so that it becomes differentiable again. We could give a complete symbolic solution in the case $n = 3$.

Let D be the set of all SPD 3×3 matrices with at most two different eigenvalues. Naive dimension counting would suggest that $\dim(D) = 5$, because we have one algebraic condition, namely the discriminant of the characteristic polynomial has to vanish. But we have two degrees of freedom to choose the eigenvalues, and two degrees of freedom to choose the eigenvector for the simple eigenvalue – the two-dimensional eigenspace for the double eigenvalue is uniquely determined as the plane orthogonal to the simple eigenvector. Thus, we get $\dim(D) = 4$.

Let C be a matrix in D. Let λ be the unique eigenvalue which is at least double. Then $C - \lambda I$ has rank at most 1. Therefore we can write C as $\lambda I \pm x x^t$ with $x \in \mathbb{R}^3$. This representation is unique and gives a parameterization of D by 4 parameters. As we check local minima on linear subspaces, we get additional restrictions on the parameters, which are linear in λ and quadratic in the three coordinates of x. The objective function is given by $\frac{\lambda + x^t x}{\lambda}$ or $\frac{\lambda}{\lambda - x^t x}$, depending on the sign of the rank 1 summand.

By Lemma 2, we only need to compute local minima in the cases where there is no local minimum on U with distinct eigenvalues (in other words, if U has a local minimum that can be found by guards then we can cross out this case even if this local minimum does not lie in Σ). Especially, we do not need to consider pairs and 3-chains. It remains to compute the local minima in the cases of double pairs, triangles, and 4-chains. Here is a case by case analysis.

Double pairs: The search space is given by two equations in λ, x_1, x_2, x_3. The linear variable λ can be eliminated, so that the search space is actually a conic in the projective plane (factoring out scalar multiplication as usual). Using well-known algorithms for curve parameterization, see e.g. [10], we can parameterize the conic, thereby reducing the search space to a projective line. The objective function transforms to a rational function of degree 4. There are 6 (maybe complex) stationary points, which are candidates for being local minima.

Triangles and 4-chains: The three equations in λ, x_1, x_2, x_3 can be solved symbolically by standard methods. There are four solutions, all of them real; three of them are singular because the λ-component vanishes. So, the search space restricts to a single point.

4 Comparison

In this section the efficiency and robustness of the symbolic approach to the element preconditioning technique is shown, by comparing with the numerical approach used in [5]. All numerical studies are done on an SGI Octane 300MHz.

First of all we calculate the best possible M-matrices of Example 1 and compare it to the numerical solution. For these cases the matrices which depend on a single parameter $0 < q < 1$ can be calculated once in a Maple implementation, and this leads to the following results:

Example 3. Now we give the results of Example 1. Only the transformed matrix is presented.

1. First of all the anisotropic rectangle with $\sigma = 0$ is considered. The best possible M-matrix is given by

$$\tilde{B} = \begin{pmatrix} 1+4q^2 & 0 & -1-q^2 \\ 0 & 1+4q^2 & -q^2 \\ -1-q^2 & -q^2 & 1+4q^2 \end{pmatrix}$$

   with a condition number of $\kappa = 3/(1+4q^2)$ for $0 < q < \sqrt{1/2}$.
2. Next the anisotropic triangle with $\sigma = 0$ was already explicitly solved in Remark 2 and leads to the matrix

$$\tilde{B} = \begin{pmatrix} 1+q^2 & 0 \\ 0 & 1+q^2 \end{pmatrix}$$

   with condition number $\kappa = 1/q^2$.
3. Finally the anisotropic triangle with $\sigma = 1$ yields

$$B = \begin{pmatrix} q^2+2 & 0 & -2 \\ 0 & q^2+2 & -2 \\ -2 & -2 & q^2+4 \end{pmatrix}$$

   and the condition number behaves like $\kappa = \frac{q^2+2}{3q^2}$.

For such types of elements where the element stiffness matrix depends only on one parameter the solution can be done in advanced. However, for more general matrices the number of parameters is too large to be efficiently solved with the Maple implementation. Therefore we make a C++ implementation in order to get a fast solution of the optimization problem. Note that no initial guess is necessary for the calculation and we always get the optimal solution. This is in contrast to a numerical approach where a different initial guess might lead to a different result, see Table 1.

Table 1. Different initial guess for the numerical case.

q^2	$\kappa((B_h^r)^{-1/2}K_h^r(B_h^r)^{-1/2})$	$\kappa((B_h^r)^{-1/2}K_h^r(B_h^r)^{-1/2})$
1.0	1.00	1.00
0.9	2.35	1.23
0.5	6.50	4.00
0.1	150.5	100

Finally we compare the CPU-time of the numerical approach and the symbolic approach. These studies are done on the triangle with $\sigma = 1$ and on the rectangle with $\sigma = 0$ (see Example 1). In Table 2 the results are presented. The C++ implementation is much faster than the numerical approach (by the factor 10) and has the advantage that no initial guess has to be used. We emphasize that in general the optimization problem must be solved for each finite element separately. In such cases, our symbolic approach can obviously save a lot of computer time on fine meshes with several hundred thousands or millions of elements.

Table 2. Comparison of numerical and symbolic approaches.

name	numerical (sec)	symbolic (sec)
triangle	0.0052	0.00047
rectangle	0.0060	0.00048

References

[1] B. Buchberger, *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal*, Ph.D. thesis, Universitat Innsbruck, Institut fur Mathematik, 1965, German.

[2] ———, *Gröbner bases: An algorithmic method in polynomial ideal theory*, Recent Trends in Multidimensional Systems Theory (N. K. Bose, ed.), D. Riedel Publ. Comp., 1985.

[3] G. E. Collins, *Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition*, Lecture Notes In Computer Science, Springer, 1975, Vol. 33, pp. 134–183.
[4] G. E. Collins and H. Hong, *Partial cylindrical algebraic decomposition for quantifier elimination*, J. Symb. Comp. **12** (1991), no. 3, 299–328.
[5] G. Haase, U. Langer, S. Reitzinger, and J. Schöberl, *Algebraic multigrid methods based on element preconditioning*, International Journal of Computer Mathematics **78** (2001), no. 4, 575–598.
[6] W. Hackbusch, *Multigrid methods and application*, Springer Verlag, Berlin, Heidelberg, New York, 1985.
[7] H. Hong, *Improvements in cad–based quantifier elimination*, Ph.D. thesis, The Ohio State University, 1990.
[8] M. Jung and U. Langer, *Applications of multilevel methods to practical problems*, Surveys Math. Indust. **1** (1991), 217–257.
[9] M. Jung, U. Langer, A. Meyer, W. Queck, and M. Schneider, *Multigrid preconditioners and their application*, Proceedings of the 3rd GDR Multigrid Seminar held at Biesenthal, Karl-Weierstraß-Institut für Mathematik, May 1989, pp. 11–52.
[10] F. Winkler, *Polynomial algorithms in computer algebra*, Springer, 1996.