ETH ZÜRICH

Impact of highly non-affine, 3D, hybrid meshes on the convergence of Iterative Methods

Bachelor Thesis

Andreas Hug

June 2016

Advisors: Prof. Dr. Ralf Hiptmair, Raffael Casagrande
Department of Mathematics, ETH Zürich

Abstract

It has been demonstrated that significant improvements in accuracy can be gained if a properly chosen anisotropic mesh is used in the numerical solution of problems that exhibit anisotropic features. However, it is also known that such anisotropic meshes cause the system matrix to be ill-conditioned. So far, investigations have been confined to affine meshes and bounds on the condition number have been established.

In this thesis, we derive a bound on the condition number for non-affine meshes. The bound is solely based on mesh properties and its quality is investigated in numerical experiments. Moreover, the effect of Diagonal- and Incomplete Cholesky preconditioners in combination with the Conjugate Gradient algorithm is experimentally studied. Numerical results show that the Incomplete Cholesky preconditioner performs very well on non-affine meshes.

Finally, an element-wise mesh quality measure is derived which can be used to compare the quality of different elements in the same mesh.

Contents

Introduction

1.1 Overview

The *Finite Element Method* (FEM) is a popular and widely used technique to solve boundary value problems for partial differential equations numerically. The FEM requires that the spatial domain, on which the partial differential equation is posed, is subdivided into smaller, disjoint elements. In 3D, tetrahedra, prisms or hexahedra are often used for this. In order to discretize complex domains, these elements must be transformed in an affine or even non-affine way to approximate the domain well. This transformation can be described by a mapping $F_K : \hat{K} \to K$ which is defined on the reference element $\hat{K}$ and maps it to the transformed element K. For each element type there is exactly one reference element. For hexahedral elements this is the unit-cube.

In engineering applications, these elements are often highly deformed to resolve the underlying physics better. An example are boundary flows where the velocity varies strongly in normal direction but not in tangential direction (w.r.t. the surface). When simulating such flows using e.g. the *Finite Volume Method* the computational elements are therefore often stretched along the boundary and very thin in the normal direction. If the fluid solver is coupled to a FEM solver (e.g. for electromagnetic field computations), these anisotropic or even non-affine elements can cause the linear system to be ill-conditioned.

1.2 Motivation

A mesh element K that can be described by an affine mapping $F_K : \hat{K} \to K$ is called affine and a mesh element where all side-lengths are roughly the same is called isotropic.

In particular a mesh element is anisotropic if it has high aspect ratio. A mesh in which all elements are affine/isotropic is called an affine/isotropic mesh. Moreover, a mesh is uniform if all elements have comparable volumes. Finally, we call a mesh non-affine if it contains non-affine elements

If all mappings F_K are *affine*, one can give bounds on the condition number κ in terms of mesh parameters (see e.g [1], [4]). In particular one can show that non-uniform or anisotropic meshes are generally more ill-conditioned than comparable isotropic, uniform meshes. However, as shown in Kamenski et al. [4], a diagonal preconditioner can reduce the effects of the anisotropy/non-uniformity of the mesh on the condition number κ. In practice, it has been observed, that Incomplete Cholesky/Incomplete LU (ILU) preconditioners also perform very well on anisotropic meshes (see [3]).

To the authors knowledge, the effect of non-affine meshes on the condition number has not been extensively studied yet and it is not known if diagonal preconditioners or ILU offer a remedy for the bad condition number. It's also not known whether estimates similar to those provided in [1] or [4] still apply.

1.3 Scope of this work

The goal of this work is to study the effect of non-affine meshes on the condition number. We are going to experimentally investigate whether the already mentioned preconditioners (Diagonal, ILU) still remedy the high condition number of ill-conditioned systems.

In the theoretical part, the existing bounds for affine meshes are generalized to non-affine meshes. This extension provides an a priori mesh quality measure, which can be used during the mesh generation process to locate problematic mesh elements.

In a later chapter, the results of numerical experiments are presented. We conduct numerical experiments to compare our bound for the condition number with the actual condition number and we study the effect of different preconditioners.

1.4 Structure of the thesis

The remainder of this thesis is structured as follows:

- **Chapter 2** gives a short summary of existing bounds for affine meshes. Afterwards, our more general bound is derived.

- **Chapter 3** provides some details about the implementation.
- **Chapter 4** presents numerical experiments performed on different meshes.
- **Chapter 5** shows a real-world application.
- **Chapter 6** presents the conclusion of this thesis.

General theory

2.1 Boundary Value Problem

We consider the following elliptic boundary value problem posed on a three-dimensional, bounded, connected domain $\Omega \subset \mathbb{R}^3$.

Problem description.

$$\begin{aligned} -\Delta u &= f \quad && \text{in } \Omega, \\ u &= g \quad && \text{on } \Gamma_D, \\ \mathbf{n} \cdot \mathbf{grad}\, u &= 0 \quad && \text{on } \Gamma_N. \end{aligned} \tag{2.1}$$

Here, u is a scalar valued function and Γ_D and Γ_N are separate parts of the domain boundary with $\Gamma_D \cup \Gamma_N = \partial\Omega$ and the function g specifies the Dirichlet data. We assume that $\Gamma_N \neq \partial\Omega$.

Variational formulation.
The variational formulation for the above boundary value problem is

$$\begin{gathered} \text{Find } u \in H^1(\Omega), u_{|\Gamma_D} = g \text{ s.t.} \quad a(u,v) = l(v) \qquad \forall v \in V_0, \\ \text{where } V_0 := \{v \in H^1(\Omega) \mid v_{|\Gamma_D} \equiv 0\}, \\ a(u,v) = \int_\Omega \mathbf{grad}\, u \cdot \mathbf{grad}\, v \,\mathrm{d}\mathbf{x}, \qquad l(v) = \int_\Omega f v \,\mathrm{d}\mathbf{x}. \end{gathered} \tag{2.2}$$

The discretization is based on a fully hybrid mesh which involves first order tetrahedra, hexahedra, prism as well as pyramids. Let $\mathcal{T}^h$ be such a triangulation of the domain Ω

and let $F_K : \hat{K} \to K$ be the mapping from the reference element to a particular transformed element $K \in \mathcal{T}^h$. We construct the standard finite element space $V_0^h \subset V_0$ using first order Lagrangian shape functions and denote the number of elements and interior vertices by N and N_{v_i}, respectively. We assume that the vertices are ordered such that the first N_{v_i} vertices do not lie in Γ_D. Choosing a basis $V_0^h = \text{span}\{\phi_1, \ldots, \phi_{N_{v_i}}\}$, we can write any $u_h \in V^h$ as a linear combination $u_h = \sum_{j=1}^{N_{v_i}} u_j \phi_j + \tilde{u}_h$ where ϕ_j is the basis function associated with the j^{th} vertex and $\tilde{u}_h \in V^h$ is such that $\tilde{u}_h|_{\Gamma_D} = g$. By plugging this expansion into (2.2) we get the equation in matrix form and the definition of the stiffness matrix,

$$A\mathbf{u} = \boldsymbol{\varphi} \qquad \text{with} \qquad \begin{cases} A_{ij} = \int_\Omega \mathbf{grad}\, \phi_i \cdot \mathbf{grad}\, \phi_j \, \mathrm{d}\mathbf{x}, \\ \boldsymbol{\varphi}_i = \int_\Omega f \phi_i \, \mathrm{d}\mathbf{x} - \int_\Omega \mathbf{grad}\, \tilde{u}_h \cdot \mathbf{grad}\, \phi_i \, \mathrm{d}\mathbf{x}. \end{cases}$$

Hereafter, we will use the shorthand notation

$$\sum_j = \sum_{j=1}^{N_{v_i}}.$$

2.2 Affine meshes

Bounds for the condition number of the stiffness matrix A have been developed by Kamenski et al. [4] for arbitrary, non-uniform but affine meshes. They improved existing bounds significantly and presented stronger bounds for both, the smallest and the largest eigenvalue of the mass matrix, as well as the stiffness matrix. Note that the bounds in this form are only valid for linear finite elements and meshes containing only tetrahedra.

2.2.1 Bound on the largest eigenvalue of the stiffness matrix

Lemma 2.1 (Lemma 4.1 [4], Largest eigenvalue)**.** *The largest eigenvalue of the stiffness matrix $A = (A_{ij})$ of the BVP (2.1) is bounded by*

$$\lambda_{max}(A) \leq (d+1)\, C_{\hat{\phi}} \max_j \sum_{K \in \omega_j} |K| \left\| J_K^{-1} J_K^{-T} \right\|_2 .$$

The factor $(d+1)$ equals the number of vertices of a d-dimensional simplex and the constant $C_{\hat{\phi}}$ depends only on the basis functions defined on the reference element and not the triangulation itself. In the following we will denote the Jacobian of F_K on the

element K by J_K. In the given case of an affine mapping F_K, J_K is constant on the element K.

2.2.2 Bound on the smallest eigenvalue of the stiffness matrix

The bound for the smallest eigenvalue has a more complex form.

Lemma 2.2 (Lemma 5.1 [4], Smallest eigenvalue)**.** *The smallest eigenvalue of the stiffness matrix $A = (A_{ij})$ of the BVP* (2.1) *is bounded from below by*

$$\lambda_{min} \geq C d_{min} N^{-1} \begin{cases} 1, & \text{for } d = 1, \\ \left(1 + \ln \frac{|\bar{K}|}{|K_{min}|}\right)^{-1}, & \text{for } d = 2, \\ \left(\frac{1}{N} \sum_{K \in T_h} \left(\frac{|\bar{K}|}{|K|}\right)^{\frac{d-2}{2}}\right)^{-\frac{2}{d}}, & \text{for } d \geq 3. \end{cases}$$

In the three-dimensional case, the second factor already reveals the bad influence of the non-uniformity on the smallest eigenvalue.

2.2.3 Bound on the condition number of the stiffness matrix

Combining the two lemmas above, one receives a bound on the condition number of the stiffness matrix.

Theorem 2.3 (Theorem 5.2 [4], Condition number of the stiffness matrix)**.** *The condition number of the stiffness matrix of the linear finite element approximation of the BVP* (2.1) *is bounded by*

$$\kappa(A) \leq C N^{\frac{2}{d}} \left(\frac{N^{1-\frac{2}{d}}}{d_{min}} \max_j \sum_{K \in \omega_j} |K| \left\| J_K^{-1} J_K^{-T} \right\|_2 \right) \times \begin{cases} 1 + \ln \frac{|\bar{K}|}{|K_{min}|}, & \text{for } d = 2, \\ \left(\frac{1}{N} \sum_{K \in T_h} \left(\frac{|\bar{K}|}{|K|}\right)^{\frac{d-2}{2}}\right)^{\frac{2}{d}} & \text{for } d \geq 3. \end{cases}$$

Remarks: This bound consists of three independent factors. The first factor $N^{\frac{2}{d}}$ corresponds to the condition number on a uniform mesh.

The second factor

$$\frac{N^{1-\frac{2}{d}}}{d_{min}} \max_j \sum_{K \in \omega_j} |K| \left\| J_K^{-1} J_K^{-T} \right\|_2$$

can be understood as a volume-weighted equidistribution quality measure. It combines the effect of the element size and the aspect ratio.

The third factor

$$\begin{cases} 1 + \ln \frac{|\bar{K}|}{|K_{min}|}, & \text{for } d = 2, \\ \left(\frac{1}{N} \sum_{K \in T_h} \left(\frac{|\bar{K}|}{|K|} \right)^{\frac{d-2}{2}} \right)^{\frac{2}{d}}, & \text{for } d \geq 3. \end{cases}$$

reflects the influence of non-uniformity on the condition number.

We are now going to consider similar estimates for non-affine meshes. The derivation is based on the approach of Kamenski et al.[4].

2.3 Non-affine meshes

2.3.1 Bound on the largest eigenvalue of the stiffness matrix

We will first derive a bound on the largest eigenvalue that depends on the diagonal entries of the stiffness matrix. Afterwards, we will derive a bound on the diagonal entries of the stiffness matrix to get a bound which depends only on mesh parameters. The final bound is presented in Lemma 2.5.

Lemma 2.4 (Largest eigenvalue)**.** *Let $\mathcal{T}^h$ be a possibly non-affine mesh in which all elements are of the same type. Then the largest eigenvalue of the stiffness matrix $A = (A_{ij})$ of the BVP (2.1) in three dimensions is bounded by*

$$\max_i A_{ii} \leq \lambda_{max} \leq n_v \max_i A_{ii}, \tag{2.3}$$

where n_v denotes the number of vertices per element.

Proof. For this proof we first recall, that for symmetric, positive-semidefinite matrices we have the inequality[1]

[1] This follows directly from

$$0 \leq (\mathbf{x} - \mathbf{y})^T M (\mathbf{x} - \mathbf{y}) = \mathbf{x}^T M \mathbf{x} - 2\mathbf{x}^T M \mathbf{y} + \mathbf{y}^T M \mathbf{y}$$
$$\Rightarrow 2\mathbf{x}^T M \mathbf{y} \leq \mathbf{x}^T M \mathbf{x} + \mathbf{y}^T M \mathbf{y}.$$

$$\mathbf{x}^T M \mathbf{y} \leq \frac{1}{2}(\mathbf{x}^T M \mathbf{x} + \mathbf{y}^T M \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{2.4}$$

The idea of the proof is now to rearrange the sum of integrals over an element as a sum over all vertices and an integral over the respective vertex patches. For this reason we define the patch ω_j as the support of the basis function ϕ_j associated with the j^{th} vertex.

$$\begin{aligned}
\mathbf{u}^T A \mathbf{u} &= \int_\Omega \mathbf{grad}\, u_h \cdot \mathbf{grad}\, u_h \,\mathrm{d}\mathbf{x} \\
&= \sum_{K \in T_h} \int_K \mathbf{grad}\, u_h \cdot \mathbf{grad}\, u_h \,\mathrm{d}\mathbf{x} \\
&= \sum_{K \in T_h} \int_K \left(\sum_{i_K=1}^{n_v} u_h^{i_K} \mathbf{grad}\, \phi_{i_K} \right) \cdot \left(\sum_{j_K=1}^{n_v} u_h^{j_K} \mathbf{grad}\, \phi_{j_K} \right) \mathrm{d}\mathbf{x} \\
&= \sum_{K \in T_h} \int_K \sum_{i_K=1}^{n_v} \sum_{j_K=1}^{n_v} u_{i_K} u_{j_K} \mathbf{grad}\, \phi_{i_K} \cdot \mathbf{grad}\, \phi_{j_K} \,\mathrm{d}\mathbf{x} \\
&\overset{(2.4)}{\leq} \sum_{K \in T_h} \int_K \sum_{i_K=1}^{n_v} \sum_{j_K=1}^{n_v} \frac{1}{2} \left(u_{i_K}^2 \mathbf{grad}\, \phi_{i_K} \cdot \mathbf{grad}\, \phi_{i_K} + u_{j_K}^2 \mathbf{grad}\, \phi_{j_K} \cdot \mathbf{grad}\, \phi_{j_K} \right) \mathrm{d}\mathbf{x} \\
&= n_v \sum_{K \in T_h} \int_K \sum_{i_K=1}^{n_v} u_{i_K}^2 \mathbf{grad}\, \phi_{i_K} \cdot \mathbf{grad}\, \phi_{i_K} \,\mathrm{d}\mathbf{x} \\
&= n_v \sum_{K \in T_h} \int_K \sum_{i_K=1}^{n_v} u_{i_K}^2 \left\| \mathbf{grad}\, \phi_{i_K} \right\|_2^2 \mathrm{d}\mathbf{x} \\
&= n_v \sum_i u_i^2 \int_{\omega_i} \left\| \mathbf{grad}\, \phi_i \right\|_2^2 \mathrm{d}\mathbf{x} \\
&= n_v \sum_i u_i^2 A_{ii} \\
&\leq n_v \max_i A_{ii} \left\| \mathbf{u} \right\|_2^2 .
\end{aligned}$$

Here n_v denotes the number of vertices per element (e.g. 8 for a hexahedron). We assumed to have a mesh containing only one element type. However, this bound would also be valid for hybrid meshes by setting $n_v = \max_{K \in T_h} n(K)$, where $n(K)$ is the number of vertices of element K.

Additionally we also get a lower bound for the largest eigenvalue by using the canonical basis vectors $\mathbf{e_j}$:

$$\lambda_{max}(A) \geq \mathbf{e_j}^T A \mathbf{e_j} = A_{jj}, \quad j = 1, \ldots, N_{v_i}.$$

□

The bound on the largest eigenvalue doesn't involve the solution $\mathbf{u}$ directly but still depends on the maximal diagonal entry of the stiffness matrix. In the next step, we are going to derive a bound on the diagonal entries in terms of mesh properties.

Bound on the diagonal entries of the stiffness matrix

By a transformation back to the reference element and the substitution rule for integrals, we get a bound on the diagonal entry of the stiffness matrix.

$$
\begin{aligned}
A_{ii} &= \int_{\omega_i} \mathbf{grad}\,\phi_i \cdot \mathbf{grad}\,\phi_i \,\mathrm{d}\mathbf{x} \\
&= \sum_{K\in\omega_i} \int_K \mathbf{grad}\,\phi_i \cdot \mathbf{grad}\,\phi_i \,\mathrm{d}\mathbf{x} \\
&= \sum_{K\in\omega_i} \int_{\hat{K}} \left(J_K^{-T}\mathbf{grad}\,\hat{\phi}_i\right)\cdot\left(J_K^{-T}\mathbf{grad}\,\hat{\phi}_i\right) \det J_K \,\mathrm{d}\hat{\mathbf{x}} \\
&= \sum_{K\in\omega_i} \int_{\hat{K}} \mathbf{grad}\,\hat{\phi}_i{}^T [J_K^{-1}J_K^{-T}]\,\mathbf{grad}\,\hat{\phi}_i \det J_K \,\mathrm{d}\hat{\mathbf{x}} \\
&= \sum_{K\in\omega_i} \int_{\hat{K}} \frac{\mathbf{grad}\,\hat{\phi}_i{}^T [J_K^{-1}J_K^{-T}]\,\mathbf{grad}\,\hat{\phi}_i}{\mathbf{grad}\,\hat{\phi}_i^T\,\mathbf{grad}\,\hat{\phi}_i} ||\mathbf{grad}\,\hat{\phi}_i||_2^2 \det J_K \,\mathrm{d}\hat{\mathbf{x}} \\
&\leq \sum_{K\in\omega_i} \int_{\hat{K}} \max_{\mathbf{x}\in\mathbb{R}^d} \frac{\mathbf{x}^T\, J_K^{-1}J_K^{-T}\,\mathbf{x}}{\mathbf{x}^T\mathbf{x}} ||\mathbf{grad}\,\hat{\phi}_i||_2^2 \det J_K \,\mathrm{d}\hat{\mathbf{x}} \\
&= \sum_{K\in\omega_i} \int_{\hat{K}} ||J_K^{-1}J_K^{-T}||_2 \cdot ||\mathbf{grad}\,\hat{\phi}_i||_2^2 \det J_K \,\mathrm{d}\hat{\mathbf{x}} \\
&\leq C_{\hat{\phi}} \sum_{K\in\omega_i} \int_{\hat{K}} ||J_K^{-1}J_K^{-T}||_2 \det J_K \,\mathrm{d}\hat{\mathbf{x}} \\
&= C_{\hat{\phi}} \int_{\omega_i} ||J_K^{-1}J_K^{-T}||_2 \,\mathrm{d}\mathbf{x}.
\end{aligned}
$$

Where $C_{\hat{\phi}} = \max\limits_{i_K=1,\dots,n_v} \max\limits_{\hat{x}\in\hat{K}} ||\mathbf{grad}\,\hat{\phi}_{i_K}(\hat{x})||_2^2$ corresponds to the norm of the largest basis function gradient on this element.

Altogether we get a bound on the largest eigenvalue of the stiffness matrix depending only on mesh properties.

Lemma 2.5 (Largest eigenvalue)**.** *Let $\mathcal{T}^h$ be a possibly non-affine mesh in which all elements are of the same type. Then the largest eigenvalue of the stiffness matrix $A = (A_{ij})$ of the BVP* (2.1) *in three dimensions is bounded by*

$$\lambda_{max} \leq n_v \, C_{\hat{\phi}} \max_i \int_{\omega_i} ||J_K^{-1} J_K^{-T}||_2 \, \mathrm{d}\mathbf{x}. \tag{2.5}$$

2.3.2 Bound on the smallest eigenvalue of the stiffness matrix

Lemma 2.6 (Smallest eigenvalue)**.** *Let $\mathcal{T}^h$ be a possibly non-affine mesh in which all elements are of the same type. Then the smallest eigenvalue of the stiffness matrix $A = (A_{ij})$ of the BVP (2.1) in three dimensions is bounded from below by*

$$\lambda_{min} \geq \frac{C_P C_S C_{\hat{K}} p_{min}}{1 + C_P} \left(\sum_{K \in T_h} \left[\min_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}}.$$

Here C_P and C_S only depend on the domain Ω while $C_{\hat{K}}$ depends only on the type of shape functions being used and the reference element type (which is assumed to be the same for all elements in the mesh). p_{min} is the smallest number of elements in a patch ω_j.

The proof of the lower bound on the smallest eigenvalue is a bit more involved than the proof for the largest eigenvalue. We are going to use Sobolev's [2, Theorem 7.10], Poincaré's and Hölder's inequality.

Proof.

$$\begin{aligned}
\mathbf{u}^T A \mathbf{u} &= \int_\Omega \mathbf{grad}\, u_h \cdot \mathbf{grad}\, u_h \, \mathrm{d}\mathbf{x} \\
&= |u_h|^2_{H^1(\Omega)} \\
&\overset{P.I.}{\geq} \frac{C_P}{1 + C_P} \|u_h\|^2_{H^1(\Omega)} \\
&\overset{S.I.}{\geq} \frac{C_P C_S}{1 + C_P} \|u_h\|^2_{L^6(\Omega)} \\
&= \frac{C_P C_S}{1 + C_P} \left(\sum_{K \in T_h} \|u_h\|^6_{L^6(K)} \right)^{\frac{1}{3}} \\
&= \frac{C_P C_S}{1 + C_P} \left(\sum_{K \in T_h} \alpha_K^{\frac{3}{2}} \right)^{-\frac{2}{3}} \left(\sum_{K \in T_h} \alpha_K^{\frac{3}{2}} \right)^{\frac{2}{3}} \left(\sum_{K \in T_h} \|u_h\|^6_{L^6(K)} \right)^{\frac{1}{3}}
\end{aligned}$$

$$
\begin{aligned}
&\overset{H.I.}{\geq} \frac{C_P C_S}{1+C_P} \left(\sum_{K \in T_h} \alpha_K^{\frac{3}{2}} \right)^{-\frac{2}{3}} \sum_{K \in T_h} \alpha_K \left\| u_h \right\|_{L^6(K)}^2 \\
&= \frac{C_P C_S}{1+C_P} \left(\sum_{K \in T_h} \alpha_K^{\frac{3}{2}} \right)^{-\frac{2}{3}} \sum_{K \in T_h} \alpha_K \left(\int_{\hat{K}} \hat{u}_h^6 \det J_K \, \mathrm{d}\hat{\mathbf{x}} \right)^{\frac{2}{6}} \\
&\geq \frac{C_P C_S}{1+C_P} \left(\sum_{K \in T_h} \alpha_K^{\frac{3}{2}} \right)^{-\frac{2}{3}} \sum_{K \in T_h} \alpha_K \cdot \left(\min_{\hat{x} \in \hat{K}} \det J_K \right)^{\frac{1}{3}} \left\| \hat{u}_h \right\|_{L^6(\hat{K})}^2 \\
&\geq \frac{C_P C_S C_{\hat{K}}}{1+C_P} \left(\sum_{K \in T_h} \alpha_K^{\frac{3}{2}} \right)^{-\frac{2}{3}} \sum_{K \in T_h} \alpha_K \cdot \left(\min_{\hat{x} \in \hat{K}} \det J_K \right)^{\frac{1}{3}} \left\| \mathbf{u}_K \right\|_2^2 .
\end{aligned}
$$

In the last step we used the equivalence of the $\|\cdot\|_{L^6(\hat{K})}^6$ and $\|\cdot\|_2$ norm on the reference element.

With the choice $\alpha_K \coloneqq \left[\min_{\hat{x} \in \hat{K}} \det J_K\right]^{-\frac{1}{3}}$ we get:

$$
\begin{aligned}
\mathbf{u}^T A \mathbf{u} &\geq \frac{C_P C_S C_{\hat{K}}}{1+C_P} \left(\sum_{K \in T_h} \left[\min_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}} \sum_{K \in T_h} \left\| \mathbf{u}_K \right\|_2^2 && (2.6) \\
&= \frac{C_P C_S C_{\hat{K}}}{1+C_P} \left(\sum_{K \in T_h} \left[\min_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}} \sum_j u_j^2 \sum_{K \in \omega_j} 1 && (2.7) \\
&\geq \frac{C_P C_S C_{\hat{K}} p_{min}}{1+C_P} \left(\sum_{K \in T_h} \left[\min_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}} \sum_j u_j^2 . && (2.8)
\end{aligned}
$$

□

2.3.3 Bound on the condition number of the stiffness matrix

By combining the two lemmas above we get an upper bound on the condition number of the stiffness matrix.

Theorem 2.7 (Condition number of the stiffness matrix)**.** *Let $\mathcal{T}^h$ be a possibly non-affine mesh in which all elements are of the same type. Then the condition number of*

the stiffness matrix $A = (A_{ij})$ *of the BVP* (2.1) *in three dimensions is bounded by*

$$\kappa(A) \leq \frac{C(\Omega, V_0^h)}{p_{min}} \frac{\max\limits_i \int\limits_{\omega_i} ||J_K^{-1} J_K^{-T}||_2 \, \mathrm{d}\mathbf{x}}{\left(\sum\limits_{K \in T_h} \left[\min\limits_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}}},$$

$$\textit{where} \quad C(\Omega, V_0^h) = \frac{n_v C_{\hat{\phi}} (1 + C_P)}{C_P C_S C_{\hat{K}}}.$$

It is not important to note that this estimate does *not* solely depend on the domain Ω but also on the triangulation $\mathcal{T}^h$ and the finite element space being used. The constants C_S and C_P only depend on the domain Ω. $C_{\hat{K}}$ and $C_{\hat{\phi}}$ depend only on the type of shape functions being used and the reference element type (which is assumed to be the same for all elements in the mesh). Moreover, p_{min} is the smallest number of elements in a patch ω_j and therefore depends on the triangulation.

Implementation

The numerical experiments are implemented in C++ and are based on the *HyDi* framework used by ABB Corporate research. HyDi is a finite element library standing for *Hy*brid *Di*scontinuous Finite Elements.

3.1 Condition number estimator

The implementation of the estimator for the condition number derived in Theorem 2.7 consists of two separate parts for the numerator and the denominator. The numerator requires the evaluation of

$$\int_{\omega_i} ||J_K^{-1} J_K^{-T}||_2 \, \mathrm{d}\mathbf{x}.$$

HyDi already provides the necessary functionality for numerical integration so that only the two-norm of the matrix $J_K^{-1} J_K^{-T}$ is to be computed at each quadrature point. The latter is achieved by a simple SVD decomposition.

The evaluation of the denominator

$$\left(\sum_{K \in T_h} \left[\min_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}}$$

is more involved due to the element-wise minimum. Finding the minimum of the determinant inside the reference element turns out to be more complex; A simple minded Quasi-Newton approach does not work very well and efficiently, so we decided to use the third-party library *CppNumericalSolvers*[1] which provides a solver for box-constraint minimization problems. This library applies a variation of the Quasi-Newton algorithm that is tuned for efficiency and accuracy.

[1] `https://github.com/PatWie/CppNumericalSolvers`

3.2 Mesh quality measure

In this section we introduce an element-wise mesh quality measure that can be used to identify "bad" elements in the mesh. Our final result from Theorem 2.7 provides an estimate for the condition number solely based on mesh properties. Unfortunately, this estimate is not localized in the way that it directly provides an estimate for the quality of an individual mesh element. However, we can further derive the following bound:

$$\kappa(A) \leq \frac{C(\Omega, V_0^h)}{p_{min}} \frac{\max\limits_i \int\limits_{\omega_i} ||J_K^{-1} J_K^{-T}||_2 \, \mathrm{d}\mathbf{x}}{\left(\sum\limits_{K \in T_h} \left[\min\limits_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{2}} \right)^{-\frac{2}{3}}} \leq C(\Omega, V_0^h) \frac{p_{max}}{p_{min}} \frac{\max\limits_{K \in \mathcal{T}^h} \int_K ||J_K^{-1} J_K^{-T}||_2 \, \mathrm{d}\mathbf{x}}{N^{-\frac{2}{3}} \min\limits_{K \in \mathcal{T}^h} \left[\min\limits_{\hat{x} \in \hat{K}} \det J_K \right]^{\frac{1}{3}}}.$$

Here, p_{max} denotes the largest number of elements in a patch ω_j. If we now define the element-wise mesh quality measure β_K as

$$\beta_K \coloneqq \max \left\{ \int_K ||J_K^{-1} J_K^{-T}||_2 \, \mathrm{d}\mathbf{x} \, , \, \left[\min_{\hat{x} \in \hat{K}} \det J_K \right]^{-\frac{1}{3}} \right\},$$

then the condition number can be bounded by

$$\kappa(A) \leq C(\Omega, V_0^h) \frac{p_{max}}{p_{min}} N^{2/3} \max_{K \in \mathcal{T}^h} \beta_K^2. \qquad (\star)$$

For this reason we think β_K is a sensible choice as a mesh quality measure. It is directly integrated into the HyDi framework and can e.g. be used for visualization purposes like in section 5.1.

Remark: The mesh quality measure β_K should only be used to compare elements of the same mesh. Comparing elements between two different meshes based on β_K is meaningless because of the constants in $(\star)$.

3.3 Computation of condition number

For the numerical experiments in chapter 4, we also need to calculate the exact condition number of the stiffness matrix in order to compare it to our estimate. Since the condition number $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$, this task is reduced to finding the extremal eigenvalues of the stiffness matrix. Since we are dealing with large, sparse matrices the *power method* is the natural choice for computing the largest eigenvalue of the stiffness matrix A. Applied to A^{-1} the power method yields the largest eigenvalue of A^{-1}, i.e. the smallest eigenvalue of A. This

variation is called *inverse power method*. However, the straightforward implementation doesn't work well because the smallest eigenvalues are not well separated from each other. Therefore, we resort to using the more sophisticated implementation provided by the third-party library *Spectra*[2].

[2] `https://github.com/yixuan/spectra`

Numerical experiments

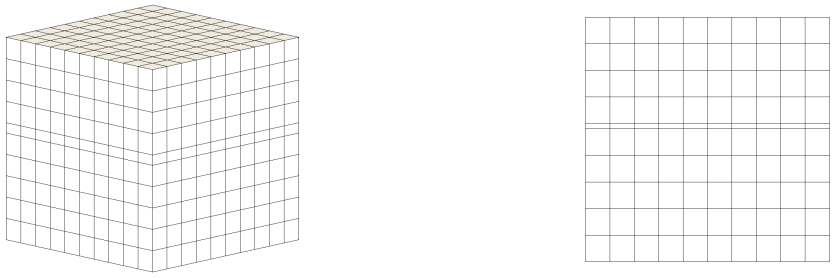
In this chapter we are going to investigate how good the bounds which we developed in the previous section behave in practice. For simplicity we look only at hexahedral meshes.

Besides the quality of the bounds, we are also going to investigate the effect of different preconditioners on the number of iterations needed by an iterative solver (Conjugate Gradient) for meshes with varying degree of non-affinity and/or non-isotropy. In particular we consider the diagonal- and ILU preconditioners. For a stable and reliable implementation of the ILU preconditioner, we switched to MATLAB. In MATLAB we used the built-in *ichol* function for the incomplete Cholesky factorization.

Note that the constants in bound (2.7) depend on the domain Ω. Therefore we had to deal with the difficulty of creating meshes of arbitrary non-affinity without changing the actual domain.

4.1 Cube with a thin layer

In these experiments the unit cube was meshed with hexahedral elements, containing a thin layer in one direction. By changing the thickness of that thin layer, we can study the effect of high aspect ratios. The mesh is still affine since the elements are only stretched/compressed in one direction. This allows us to verify the correctness of our bound and the implementation by comparing it to the existing bounds for affine meshes. The cube with the thin layer is depicted in Figure 4.1.

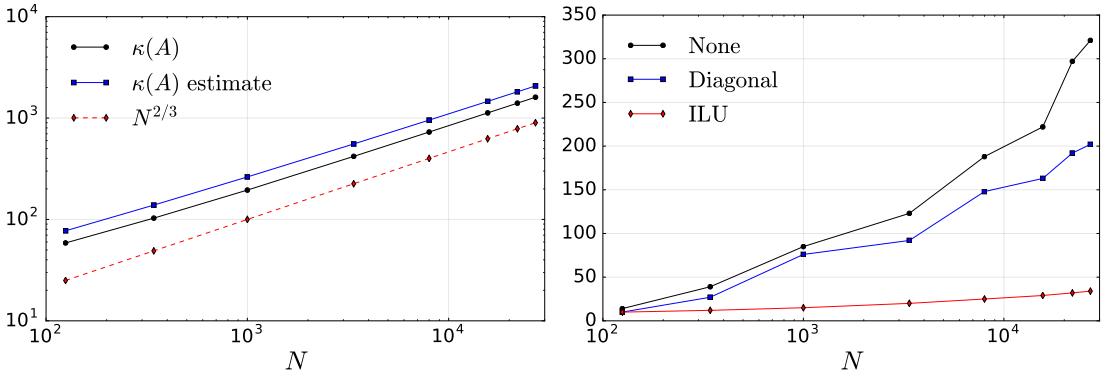


$\mathcal{O}(N^{2/3})$ compressed elements. Side view.

FIGURE 4.1: Unit cube with a thin layer.

4.1.1 Varying the number of elements

In the first experiment, only the dependence on the number of elements is investigated. We refined the mesh uniformly while keeping the aspect ratio of the elements in the thin layer fixed at 25:25:1. The condition number and the CG iterations for different number of elements N is presented in the following figure.



(a) Condition number for varying number of elements. The aspect ratio is fixed at 25:25:1.

(b) Number of CG (Conjugate Gradient) iterations for different preconditioners.

FIGURE 4.2: Results for different number of elements N for the mesh covering the unit cube, containing a thin layer.

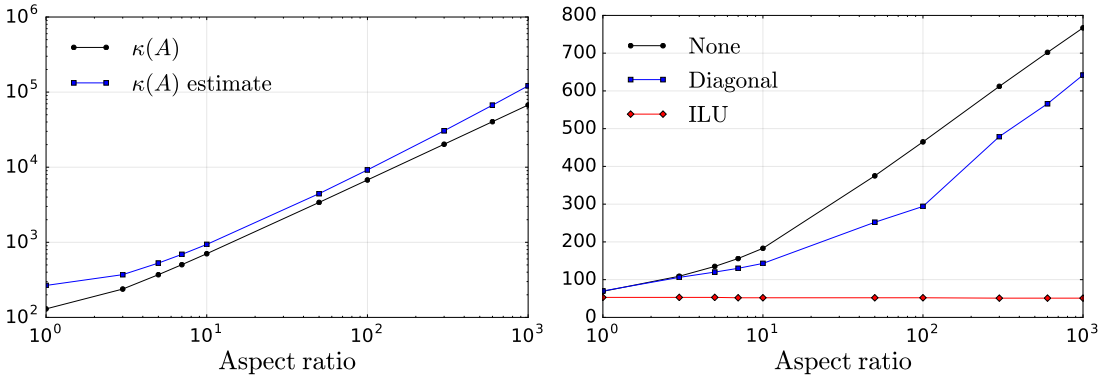
In Figure 4.2 a), one can see that our estimate is able to predict the dependence on the number of elements N correctly for affine meshes. The dashed line corresponds to an algebraic growth of $N^{2/d}$, where d is the dimension ($d = 3$ in our case). This term corresponds to the dominating factor in the condition estimate 2.3 that captures the dependence on the number of elements.

In Figure 4.2 b), the number of CG iterations needed for convergence is plotted for different number of elements. The diagonal preconditioner performs poorly and is not able

to significantly remedy the effect of the high condition number. The ILU preconditioner can reduce the effect almost completely.

4.1.2 Varying the aspect ratio

In a second experiment, we analyze the dependence on the aspect ratio. The second factor in estimate 2.3 already gives an idea, how the condition number behaves when the aspect ratio of the elements is changed. We accomplished this by fixing the number of elements in the mesh and changing the height of the thin layer. The resulting condition number and the number of CG-iterations needed for different aspect ratios is shown in the following two figures.



(a) Exact and estimated condition number for varying aspect ratio.

(b) Number of CG iterations using different preconditioners.

FIGURE 4.3: Results for varying aspect ratios of the elements in the thin layer. The number of elements is fixed at $N = 31^3 = 29791$.

As before, our estimate predicts the condition number for different aspect ratios very well. For the number of iterations of the solver we get a similar result as before. Although the diagonal preconditioner performs better than CG without a preconditioner, it is not able to really reduce the number of iterations. On the contrary, the ILU preconditioner is able to completely remedy the effect of elements with high aspect ratios.

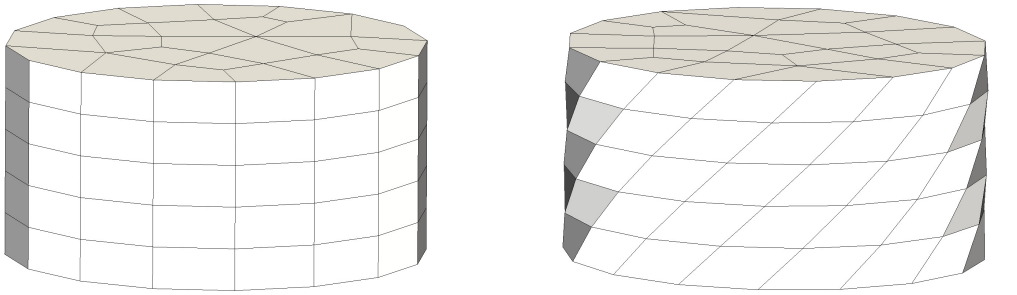
These results underpin the incomplete Cholesky preconditioner as a valuable choice for affine meshes.

4.2 Cylinder with rotated layers

In this numerical experiment, we look at the spatial discretization of a cylinder. As in the previous experiment, the mesh contains only hexahedral elements. The mesh is

generated by *Gmsh*. We first meshed the circular base area with quadrilaterals and later extruded it to a cylinder consisting of several layers. In contrast to the cube with the thin layer, the elements in this mesh are no longer affine. The mesh is depicted in Figure 4.4.

As in the previous section, we conduct three separate tests for the cylindrical mesh to investigate different effects separately.



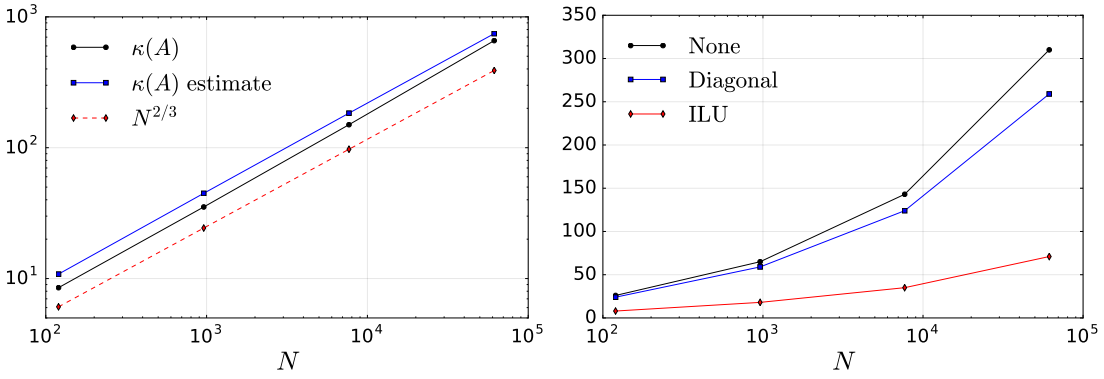
(a) Undeformed cylinder.

(b) Cylinder with a total rotation of $\frac{\pi}{3}$.

FIGURE 4.4: Cylindrical mesh (left) consisting of five layers and the same mesh with rotated layers (right).

4.2.1 Varying the number of elements

In a first step, we refined the unrotated mesh uniformly to investigate the dependence on the number of elements.



(a) Exact and estimated condition number for varying number of elements.

(b) Number of CG iterations using different preconditioners.

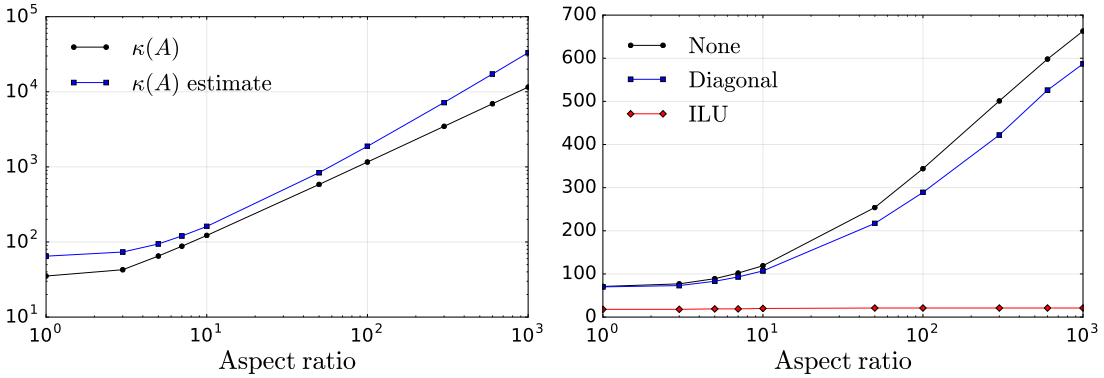
FIGURE 4.5: Results for cylindrical mesh with different number of elements, using uniform refinement.

The result for the condition number looks in principle the same. It seems like the non-affinity of the elements itself doesn't significantly change the dependence on the number of elements when using uniform refinement. It still shows a growth of $N^{2/3}$ as before for the cube with the thin layer.

Similarly, the results for the number of CG iterations exhibit the same behavior as the previous mesh. Only the ILU preconditioner is able to reduce the number of iterations notably.

4.2.2 Varying the aspect ratio

For this mesh, as before, we also analyze the effect of elements with high aspect ratios. In analogy to the cube with the thin layer, we change the height of one layer in the cylindrical mesh.



(a) Exact and estimated condition number for varying aspect ratio.

(b) Number of CG iterations using different preconditioners.

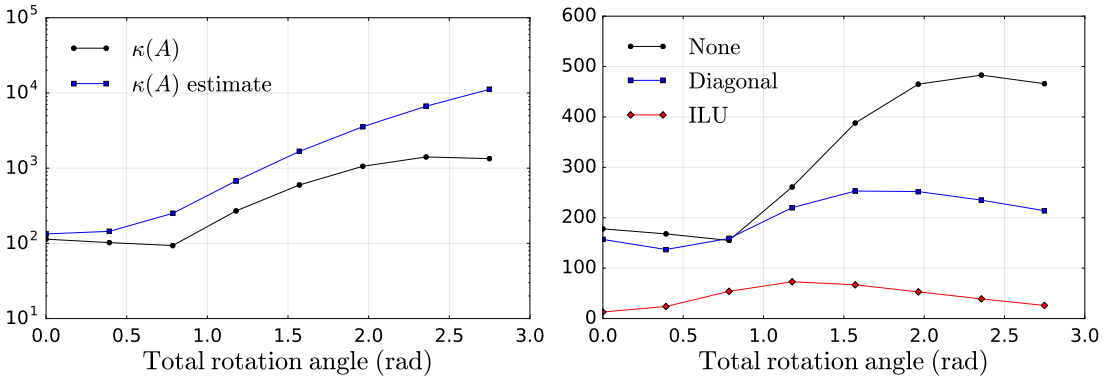
FIGURE 4.6: Result for varying aspect ratios of the elements in one layer. The number of elements is fixed at $N = 2590$.

As before, the ILU preconditioner is able to completely remedy the high aspect ratios of the elements. The diagonal preconditioner does not significantly improves the convergence.

4.2.3 Varying the rotation angle

In this experiment we want to analyze the effect of extremely distorted, non-affine elements. For this reason, we rotate each layer around the center point and look at the dependence on the rotation angle between the bottom and the top layer. Figure 4.4 shows such a mesh for the angle $\pi/3$, i.e. the top and bottom layer have been rotated

against each other by $\pi/3$ rad. The rotation angle for the layers in-between is linearly interpolated.



(a) Exact and estimated condition number for varying total rotation angle.

(b) Number of CG iterations using different preconditioners.

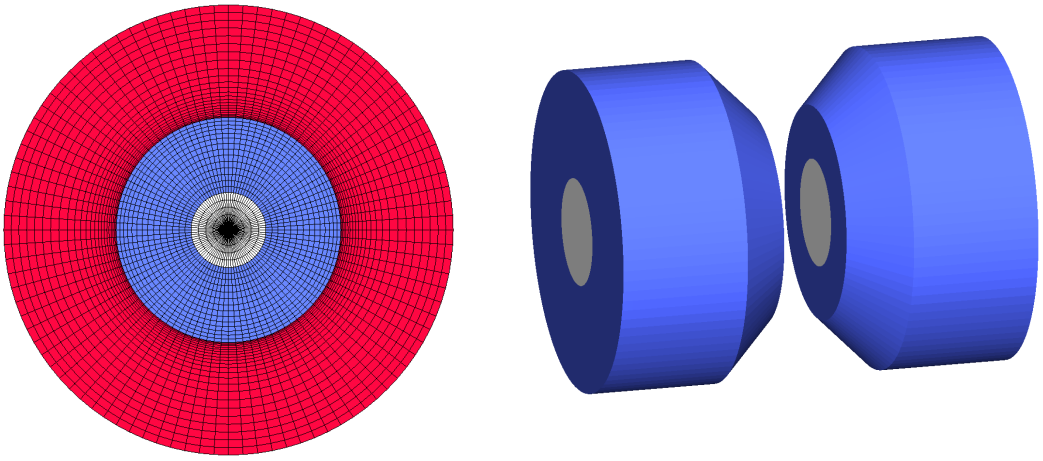
FIGURE 4.7: Results for cylindrical mesh with the layer rotated around the center axis. $N = 2590$

In Figure 4.7 a) one can see that our estimate isn't as good as before but qualitatively it still shows the correct behavior. The reason for this is probably the bound for the smallest eigenvalue, which involves the term $\min \det J_K$ (see section 2.6). For highly non-affine elements, where the determinant varies a lot inside the element, this estimate gets worse and worse.
In Figure 4.7 b) the number of CG-iterations for different rotation angles is plotted. In this experiment, the diagonal preconditioner clearly outperforms CG without a preconditioner. However, the ILU preconditioner is, even in the case of highly non-affine elements, able to remedy this effect and keep the number of iterations more or less constant.

Remark: Strictly speaking, the domain covered by the cylindrical mesh slightly changes when rotating the layers and thus the constant in our bound does not remain the same. Nevertheless, we assume that this effect is negligible for our experiment. A proper analysis would be out of the scope of this thesis.

Industrial use

In this chapter we apply the mesh quality measure 3.2 to an industrial sample mesh provided by ABB Corporate research. The mesh consists of three different materials which model two conductors (white) surrounded by an insulator (blue) with a small gap in-between filled with a plasma (red). On the right of the following figure the separated conductors and the insulator without the surrounding plasma is depicted.



(a) Top view of the cylindrical mesh.

(b) Two contacts, the conductor (gray) and the insulator (blue).

FIGURE 5.1: Two views of the mesh. The materials are colored differently.

The mesh consists only of hexahedral elements. Moreover, all the elements except for the conductor are symmetric around the center. The conductor itself consists of a finer mesh similar to the cylindrical mesh presented in Figure 4.4. These elements are also no longer affine (cf. Figure 5.2).

5.1 Mesh quality measure β

To see our mesh quality measure in action, we apply it on the mesh from above. In Figure 5.2 we visualize the result for the cross-section of the conductor which contains the non-affine elements. Apparently, the stretched elements on the outside seem to be less problematic then non-affine elements which are not stretched.

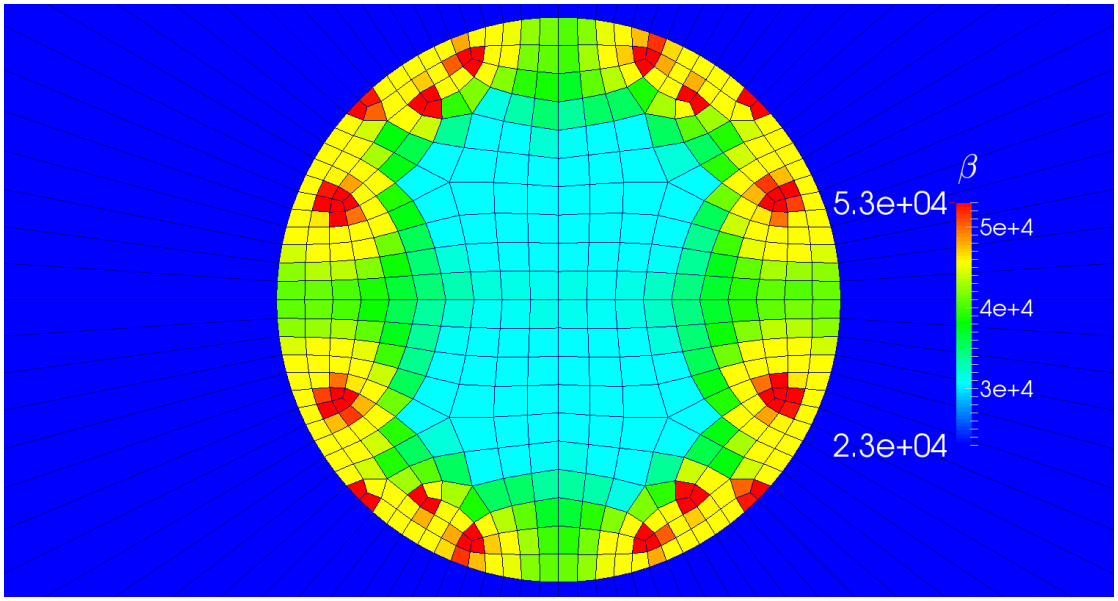


FIGURE 5.2: Close up of the conductor center (cf. Figure 5.1). Mesh quality measure β (cf. 3.2) evaluated on the cross-section of the conductor.

5.2 Different preconditioners

For this mesh, we solve the Poisson's equation for the electric potential. The electric potential on one end of the conductor is set to $u = 1$ and $u = 0$ on the other. The conductivity of the insulator is set to 0 and is excluded from the domain whereas the conductivity of the surrounding plasma is set to 10^{-4}.

Likewise, we solve the resulting linear system with different preconditioners and the conjugate gradient algorithm. The number of iterations for each preconditioner is depicted in Figure 5.3.

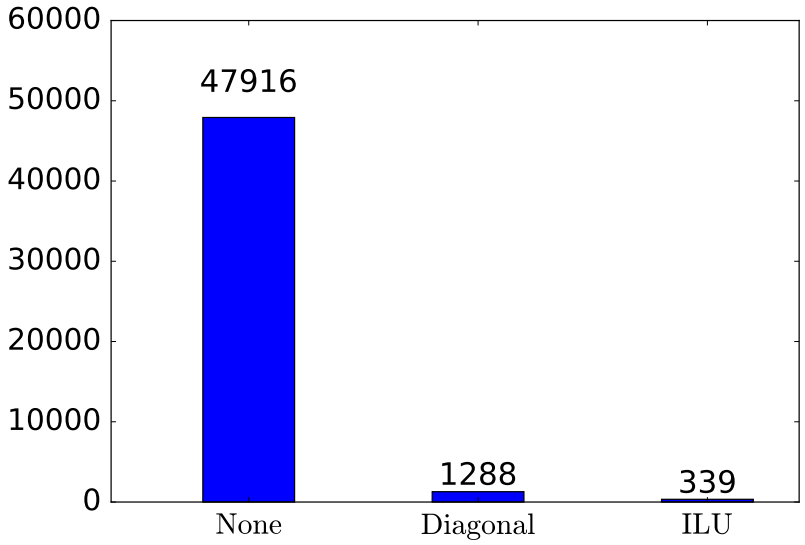


FIGURE 5.3: Conjugate gradient iterations for solving the Poisson's equation on the above mesh.

In this case, both preconditioners are able to significantly reduce the number of iterations. The diagonal preconditioner reduces the number of iterations around factor 35. Nevertheless, the ILU preconditioner performs once more even better than the diagonal preconditioner and is able to reduce the number of iterations by the factor 140.

Conclusion

We have extended Theorem 2.3, which is due to Kamenski et al. and only holds for affine, simplical meshes, to non-affine meshes in Theorem 2.7. This theorem gives an upper bound on the condition number κ of the BVP 2.1 that depends only on mesh characteristics and the domain Ω. We transformed this estimate to an element-wise measure which helps to identify "bad elements" during mesh creation. This mesh quality measure was directly integrated into the HyDi FEM framework to guide the mesh development process. Numerical experiments show that Theorem 2.7 is quite sharp for meshes which are not *highly* non-affine. The bound still remains valid for highly non-affine meshes like in subsection 4.2.3 but the predicted condition number can be too high. A tighter bound on the smallest eigenvalue would improve the estimate in such cases.

The performance of different preconditioners in combination with the conjugate gradient algorithm is also experimentally studied. Our results are consistent with the observations already made in [3]: the ILU preconditioner performs exceptionally well for most of the meshes. The ILU preconditioner is able to completely remedy the effect of anisotropic meshes i.e. meshes with high aspect ratios. The diagonal preconditioner on the other side does not show significant improvements. Only for the highly non-affine, cylindrical mesh of section 4.2.3 it is able to reduce the number of CG iterations. This might qualify the diagonal preconditioner as a good and computationally cheap choice for highly non-affine meshes. Overall the ILU preconditioner outperforms the diagonal preconditioner by far and seems to be the method of choice. This was once more confirmed in 5 where the ILU preconditioner also yielded the best result.

Bibliography

[1] Randolph E. Bank and L. Ridgway Scott. "On the Conditioning of Finite Element Equations with Highly Refined Meshes". In: *SIAM Journal on Numerical Analysis* 26.6 (1989), pp. 1383–1394.

[2] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order*. Grundlehren der mathematischen Wissenschaften. Berlin, New York: Springer-Verlag, 1983.

[3] Weizhang Huang. "Metric tensors for anisotropic mesh generation". In: *Journal of Computational Physics* 204.2 (2005), pp. 633–665.

[4] Lennard Kamenski, Weizhang Huang, and Hongguo Xu. "Conditioning of Finite Element Equations with Arbitrary Anisotropic Meshes". In: (2012).

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Eigenständigkeitserklärung

Die unterzeichnete Eigenständigkeitserklärung ist Bestandteil jeder während des Studiums verfassten Semester-, Bachelor- und Master-Arbeit oder anderen Abschlussarbeit (auch der jeweils elektronischen Version).

Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

Titel der Arbeit (in Druckschrift):

Verfasst von (in Druckschrift):
Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.

Name(n):	**Vorname(n):**

Ich bestätige mit meiner Unterschrift:
- Ich habe keine im Merkblatt „Zitier-Knigge“ beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

Ort, Datum	**Unterschrift(en)**

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.