3D Eddy–Current Computation Using Krylov Subspace Methods

Martin Reissel
Fachbereich Mathematik
Universität Kaiserslautern
D-67653 Kaiserslautern
Germany

6/1993

Abstract

This paper considers the numerical solution of a transmission boundary-value problem for the time-harmonic Maxwell equations with the help of a special finite volume discretization. Applying this technique to several three-dimensional test problems, we obtain large, sparse, complex linear systems, which are solved by using BiCG, CGS, BiCGSTAB resp., GMRES. We combine these methods with suitably chosen preconditioning matrices and compare the speed of convergence.

1 Introduction

The investigations described in this paper were motivated by a problem which was brought to our attention by a company manufacturing generators for power plants. During the last two decades, the size of the newly built power plants has been continuously increasing, thus causing a significant change in the type of generators being installed. For reasons of effectivity one uses one large instead of several small generators. Since these extremely large and expensive machines are designed individually for each customer, there is no possibility for empirical improvements.

It was for that reason, that in the late 70s the development of a computer program for the optimization of generators was started. One important part of this software tool is an algorithm for the numerical computation of induced electromagnetic fields.

In the first step, the whole situation was simplified by considering only cross-sections of a generator, thus reducing the original three dimensional problem to two dimensions. In some cases, this simplification showed an acceptable behaviour, but for large machines, where very strong fields occur, the results were not very convincing. So it became obvious, that three dimensional computations were unavoidable.

After having made first attempts to implement such a method, it soon turned out, that most of the nice numerical properties of the two dimensional problem are lost in the three dimensional case. The main difficulties are:

- The linear systems, which were generated by a finite-volume discretization, could in general not be solved by any standard algorithm.
- In some of the few cases where results were obtained, the computed fields showed a very strange, physically meaningless behaviour.

In this paper we first present some theoretical results, which give an answer to the second problem. We then introduce a finite-volume method for the discretization of the corresponding equations. In Chapter 5 we give a unified description of the most popular Krylov subspace methods BiCG, CGS, CGSTAB, GMRES and derive the iteration procedures which were used in our implementation.

In Chapter 6 we present the results obtained for some model problems and compare the performance of the linear system solvers combined with different preconditioning techniques.

Finally we give a short outlook and discuss several aspects of the theoretical and numerical background of our problem.

2 The physical model

The consideration of the electromagnetic fields in generators leads, as many other problems in electrical engineering, to a transmission boundary-value problem for the time-harmonic Maxwell equations.

A bounded domain of conducting material $D_- \subset R^3$ is surrounded by an isolator, usually air. In $D_+ = R^3 \setminus \bar{D}_-$, a given, time-harmonic current density $\tilde{J}(x,t) = J(x)e^{-i\omega t}$ induces electromagnetic fields in D_- (Fig. 1).

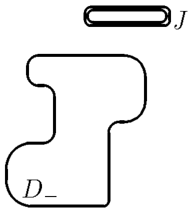

Fig. 1

The behaviour of the electric fields E_+ resp. E_- and the magnetic fields H_+ resp. H_- in D_+ resp. D_- is described by the *classical transmission boundary-value problem*

$$\begin{aligned} \operatorname{curl} H_+ &= J - i\omega\epsilon_+ E_+ \\ \operatorname{curl} E_+ &= i\omega\mu_+ H_+ \end{aligned} \quad \text{in } D_+, \qquad \begin{aligned} \operatorname{curl} H_- &= (\sigma_- - i\omega\epsilon_-)E_- \\ \operatorname{curl} E_- &= i\omega\mu_- H_- \end{aligned} \quad \text{in } D_-,$$

$$\begin{aligned} n \wedge H_+ &= n \wedge H_- \\ n \wedge E_+ &= n \wedge E_- \end{aligned} \qquad \text{on } \Gamma := \partial D_+ = \partial D_-,$$

with *Silver-Müller radiation condition*

$$H_+ \wedge \frac{x}{|x|} - E_+ = o\left(\frac{1}{|x|}\right), \qquad \text{uniformly for} \quad |x| \to \infty$$

and coefficients

- $\omega > 0$ frequency,
- $\epsilon_+, \epsilon_- > 0$ electric permittivity in D_+, D_-,
- $\mu_+, \mu_- > 0$ magnetic permeability in D_+ ,D_-,
- $\sigma_- > 0$ electric conductivity in D_-.

This problem is well investigated [4] [10].

In connection with devices working at low frequencies, especially power frequencies, the displacement currents in the above equations are neglected :

$$\begin{array}{lcll} \operatorname{curl} H_+ = J & \text{in } D_+ & \operatorname{curl} H_- = \sigma_- E_- & \text{in } D_-. \\ \operatorname{curl} E_+ = i\omega\mu_+ H_+ & & \operatorname{curl} E_- = i\omega\mu_- H_- & \end{array} \tag{1}$$

Moreover the boundary and radiation conditions are changed to

$$\begin{array}{ll} n \wedge H_+ = n \wedge H_- & \text{on } \Gamma, \\ n \cdot (\mu_+ H_+) = n \cdot (\mu_- H_-) & \\ & \\ H_+(x) = o(1) & \text{uniformly for} |x| \to \infty. \\ E_+(x) = o(1) & \end{array} \tag{2}$$

In quite a number of applications, the unbounded domain D_+ is cut off (Fig. 2). Instead of D_+, D_-, we consider the domains D_+^c, D_- (Fig. 2).

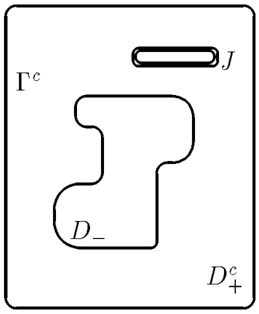

Fig. 2

On the new boundary Γ^c, the normal component of the magnetization $\mu_+ H_+$ is prescribed. The corresponding data is usually given by measurements or is approximated in some way. The resulting problem

$$\begin{array}{lcll} \operatorname{curl} H_+^c = J & \text{in } D_+^c, & \operatorname{curl} H_- = \sigma_- E_- & \text{in } D_-, \\ \operatorname{curl} E_+^c = i\omega\mu_+ H_+^c & & \operatorname{curl} E_- = i\omega\mu_- H_- & \end{array} \tag{3}$$

$$\begin{array}{ll} n \wedge H_+^c = n \wedge H_- & \text{on } \Gamma, \\ n \cdot (\mu_+ H_+^c) = n \cdot (\mu_- H_-) & \end{array} \tag{4}$$

$$n \cdot (\mu_+ H_+^c) = f \qquad \text{on } \Gamma^c, \tag{5}$$

is from now on called *bounded problem*. In contrast to this, we refer to (1),(2) as the *unbounded problem*.

In Chapter 3 we give existence and uniqueness results for both, the bounded and the unbounded case. From these theorems we see, that the introduction of the additional boundary Γ^c and corresponding artificial boundary conditions does not change the internal structure of the transmission boundary-value problem in the sense that we get the same type of existence and uniqueness results.

Therefore we concentrate in the remaining chapters on the bounded problem (3)–(5).

3 Existence and uniqueness

To establish existence and uniqueness results, we first have to specify some assumptions on D_+, D_+^c, D_- and the coefficients $\omega, \mu_+, \mu_-, \sigma_-$.

Let $D_- \subset R^3$ be an open, bounded domain of class C^2. The complement $D_+ = R^3 \setminus \bar{D}_-$ should be connected. D_- is the union of m connected components D_-^j, $j = 1, \ldots, m$, having topological genus p_j. The boundaries Γ^j are disjoint, closed surfaces of class C^2. Defining $\Gamma = \bigcup_{j=1}^m \Gamma^j$ we get $\Gamma = \partial D_+ = \partial D_-$.

The topological genus of D_+ resp. D_- is $p = \sum_{j=1}^m p_j$. There exist p surfaces $\Sigma_+^i \subset D_+$ and $\Sigma_-^i \subset D_-$, $i = 1, \ldots, p$, such that $D_+ \setminus \bigcup_{i=1}^n \Sigma_+^i$ and $D_- \setminus \bigcup_{i=1}^n \Sigma_-^i$ are simply connected. The boundary curves $\gamma_+^i = \partial \Sigma_+^i$ and $\gamma_-^i = \partial \Sigma_-^i$ lie on Γ.

Let D^c be a simply connected, open, bounded domain in R^3, such that $\bar{D}_-, \bar{\Sigma}_+^i \subset D^c$. The boundary $\Gamma^c = \partial D^c$ is assumed to be C^2. D_+^c is now defined as $D_+^c := D^c \setminus \bar{D}_-$. Therefore, $D_+^c \setminus \bigcup_{i=1}^n \Sigma_+^i$ is simply connected and the topological genus of D_+^c is also p.

The material coefficients $\omega, \mu_+, \mu_-, \sigma_-$ are real, positive constants. For the prescribed data in (1),(2) resp., (3)–(5) we suppose

$$J \in C^1(R^3), \quad \mathrm{div} J = 0, \quad \mathrm{supp(J)} \subset D_J^c,$$

resp.,

$$J \in C^1(R^3), \quad \mathrm{div} J = 0, \quad \mathrm{supp}(J) \subset D_J^c,$$

$$f \in C^{0\alpha}(\Gamma^c),$$

where the domains D_J, $(\bar{D}_J \subset D_+)$ resp., D_J^c, $(\bar{D}_J^c \subset D_+^c)$ are bounded. Moreover, we are looking for classical solutions of (1),(2)

$$E_+, H_+ \in C^1(D_+) \cap C(\bar{D}_+),$$

$$E_-, H_- \in C^1(D_-) \cap C(\bar{D}_-)$$

and (3),(4),(5)

$$E_+^c, H_+^c \in C^1(D_+^c) \cap C(\bar{D}_+^c),$$
$$E_-, H_- \in C^1(D_-) \cap C(\bar{D}_-).$$

Under these assumptions, the following results can be shown [5]:

Theorem 1 *For $J \in C^1(R^3)$, $divJ = 0$, $supp(J) \subset D_J$, the unbounded problem (1),(2) is solvable.*

In the homogeneous case $J = 0$ we get exactly p linear independent solutions H_+, E_-, H_-, where p denotes the topological genus of D_+ resp.,D_-. The different solutions are characterized by their circulations

$$h_i = \int_{\gamma_+^i} \tau \cdot H_+ \, dl, \qquad i = 1, \ldots, p, \tag{6}$$

along γ_+^i.

E_+ is not uniquely determined.

Theorem 2 *For $J \in C^1(R^3)$, $divJ = 0$, $supp(J) \subset D_J^c$, $f \in C^{0\alpha}(\Gamma^c)$ the bounded problem (3)–(5) is solvable if and only if*

$$\int_{\Gamma^c} f\, ds = 0. \tag{7}$$

In the homogeneous case $J = 0$ we get exactly p linear independent solutions H_+^c, E_-, H_-, where p denotes the topological genus of D_+^c resp., D_-. The different solutions are characterized by their circulations

$$h_i^c = \int_{\gamma_+^i} \tau \cdot H_+^c\, dl, \qquad i = 1, \ldots, p, \tag{8}$$

along γ_+^i.

E_+^c is not uniquely determined.

We see, that in the simply connected case $p = 1$ we get unique fields H_+, E_-, H_- resp., H_+^c, E_-, H_-, provided that in the second case condition (7) is fulfilled.

For multiply connected domains we have for both problems a p-dimensional hyperplane of solutions. The different solutions are uniquley determined if the circulations h_i resp., h_i^c are prescribed.

The nonphysical results mentioned in the introduction are caused by the nonuniqueness of the solutions in the multiply connected case. Through the naive application of finite-volume methods to this problem one randomly picks out one of these solutions which might have nothing to do with reality.

4 The finite volume approximation

Now we will take care of the discretization of the bounded problem (3)–(5). Due to Theorem 2 we know, that there exists a p-dimensional hyperplane of solutions, where p denotes the topological genus of D_- resp., D_+^c. In the first part of this chapter, we modify (3)– (5) and obtain an uniquely solvable system of differential equations, which enables us to construct the whole set of solutions of the bounded problem. In the second section we are concerned with the discretization of this modified problem.

4.1 Modification of the bounded problem

The solutions of (3)–(5) can be divided into two classes:

- The nonhomogeneous solution ($J \neq 0$, $f \neq 0$) with zero circulations

$$h_i^c = \int_{\gamma_+^i} \tau \cdot H_+^c\, dl = 0, \qquad i = 1, \ldots, p.$$

- The homogeneous solutions ($J = 0$, $f = 0$) with nonzero circulations.

Following Theorem 2, the hyperplane of solutions is completely determined, if the nonhomogeneous solution of (3)–(5) with vanishing circulations h_i^c and the p different homogeneous solutions obtained by setting $h_i^c = \delta_{ij}$, $i = 1, \ldots, p$, $j = 1, \ldots, p$ are known.

We now want to transform the equations (3)–(5) in order to get an uniform description of both problems described above.

We start with the nonhomogeneous case. Writing down once again the corresponding equations

$$\left.\begin{array}{l} \text{curl } H_+^c = J \\ \text{curl } E_+^c = i\omega\mu_+ H_+^c \end{array}\right. \quad \text{in } D_+^c, \qquad \left.\begin{array}{l} \text{curl } H_- = \sigma_- E_- \\ \text{curl } E_- = i\omega\mu_- H_- \end{array}\right. \quad \text{in } D_-,$$

$$\left.\begin{aligned} n \wedge H_+^c &= n \wedge H_- \\ n \cdot (\mu_+ H_+^c) &= n \cdot (\mu_- H_-) \end{aligned}\right\} \quad \text{on } \Gamma,$$

$$n \cdot (\mu_+ H_+^c) = f \qquad \text{on } \Gamma^c,$$

we observe, that due to $J \in C^1(R^3)$, $\mathrm{div} J = 0$ and the law of Biot and Savart, there exists a field H_{BS} with

$$\mathrm{curl}\, H_{BS} = J, \qquad \mathrm{div} H_{BS} = 0 \qquad \text{in } R^3,$$

$$\int_{\gamma_+^i} \tau \cdot H_{BS}\, dl = 0, \qquad i = 1, \ldots, p,$$

so that

$$\mathrm{curl}(H_+^c - H_{BS}) = 0, \qquad \mathrm{div}(H_+^c - H_{BS}) = 0,$$

$$\int_{\gamma_+^i} \tau \cdot (H_+^c - H_{BS}) = 0.$$

As is well known from potential theory [3], this entails

$$H_+^c = H_{BS} + \nabla \phi \tag{9}$$

and since $\mathrm{div} H_+^c = \mathrm{div}(H_+^c - H_{BS}) = 0$ we get

$$\triangle \phi = 0.$$

Therefore the magnetic field in D_+^c is split up into the Biot-Savart field corresponding to J and the gradient of a scalar potential.

At this stage, it is time to consider the "artificial" boundary condition $n \cdot (\mu_+ H_+^c) = f$ on Γ^c. We require the artificial boundary Γ^c to be "far away" from the conducting body D_- (without getting too precise what this means). Since the Biot-Savart field H_{BS} represents the true magnetic field obtained in the absence of the conductor D_-, and H_+^c is given by (9), the magnetic field H_+^c observed on Γ^c will be dominated by H_{BS} and the influence of the fields induced in D_- may be neglected. This allows us to set

$$n \cdot (\mu_+ H_+^c) = f = n \cdot (\mu_+ H_{BS}) \qquad \text{on } \Gamma^c, \tag{10}$$

so that $\partial_n \phi = 0$ on Γ^c.

Now we have of course to ask, whether this choice of f fulfills the solvability condition $\int_{\Gamma^c} f\, ds = 0$? Taking a closer look at Biot-Savarts law, the answer is obvious. Since H_{BS} is given by

$$H_{BS} = \mathrm{curl} A, \qquad A = J * \Phi, \qquad \Phi = \frac{1}{4\pi} \frac{1}{\|x\|_2}$$

and Γ^c is a closed surface, we get

$$\int_{\Gamma^c} f\, ds = \mu_+ \int_{\Gamma^c} n \cdot \mathrm{curl} A\, ds = 0$$

by using Stokes' Theorem.

Eliminating E_-, we finally have derived the following system of equations for the magnetic fields in the nonhomogeneous case:

$$\left.\begin{aligned} H_+^c &= H_{BS} + \nabla \phi \\ \triangle \phi &= 0 \end{aligned}\right\} \text{ in } D_+^c, \qquad \left.\begin{aligned} \mathrm{curl}(\rho_- \mathrm{curl}\, H_-) &= i\omega\mu_- H_- \\ \rho_- &= 1/\sigma_- \end{aligned}\right\} \text{ in } D_-,$$

$$\partial_n \phi = 0 \quad \text{on } \Gamma^c, \qquad \left.\begin{aligned} n \wedge H_+^c &= n \wedge H_- \\ n \cdot (\mu_+ H_+^c) &= n \cdot (\mu_- H_-) \end{aligned}\right\} \text{ on } \Gamma.$$

Considering the homogeneous problems ($J = 0,\ f = 0,\ h_i^c = \delta_{ij}$) we proceed in a similar way. From potential theory we know, that there exist exactly p linear independent fields N_+^i in D_+^c, $i = 1, \ldots, p$, the so-called *Neumann fields*, having the following properties [3]:

$$\mathrm{div} N_+^i = 0, \qquad \mathrm{curl} N_+^i = 0 \qquad \text{in } D_+^c,$$

$$n \cdot N_+^i = 0 \qquad \text{on } \partial D_+^c = \Gamma^c \cup \Gamma,$$

$$\int_{\gamma_+^j} \tau \cdot N_+^i \, dl = \delta_{ij}.$$

If we therefore consider the homogeneous problem with circulations $h_i^c = \delta_{ij}$, we get

$$\mathrm{div}(H_+^c - N_+^i) = 0, \qquad \mathrm{curl}(H_+^c - N_+^i) = 0, \qquad \text{in } D_+^c,$$

$$\int_{\gamma_+^i} \tau \cdot (H_+^c - N_+^i) \, dl = h_i^c - \delta_{ij} = 0,$$

so that we have again

$$H_+^c = N_+^i + \nabla\phi, \qquad \triangle\phi = 0, \qquad \text{in } D_+^c.$$

Moreover, $f = 0$ on Γ^c now results in

$$0 = f = n \cdot H_+^c = n \cdot (N_+^i + \nabla\phi) = \partial_n \phi \qquad \text{on } \Gamma^c.$$

Thus we have shown, that the inhomogeneous as well as the homogeneous case may be reduced to

$$\left.\begin{array}{l} H_+^c = H_0 + \nabla\phi \\ \triangle\phi = 0 \end{array}\right\} \text{in } D_+^c, \qquad \left.\begin{array}{r} \mathrm{curl}(\rho_- \,\mathrm{curl}\, H_-) = i\omega\mu_- H_- \\ \rho_- = 1/\sigma_- \end{array}\right\} \text{in } D_-,$$

$$\left.\begin{array}{r} n \wedge H_+^c = n \wedge H_- \\ n \cdot (\mu_+ H_+^c) = n \cdot (\mu_- H_-) \end{array}\right\} \text{on } \Gamma, \tag{11}$$

$$\partial_n \phi = 0 \qquad \text{on } \Gamma^c,$$

where H_0 is either the Biot-Savart field corresponding to J, or one of the p Neumann fields N_+^i.

Existence and uniqueness of $H_+^c = H_0 + \nabla\phi$ and H_- still follow from Theorem 2. Since the equations and boundary-conditions only depend on $\nabla\phi$, the potential ϕ itself is only determined up to a constant (since D_+^c is connected).

4.2 The discretization

According to the last section, all solutions of the bounded problem can be computed using (11). Therefore, we are going to discretize the equations of (11) instead of (3)–(5).

For the theoretical results presented in the last chapter, we assumed that the material coefficients $\omega, \mu_+, \mu_-, \sigma_-$ are posistive constants. To allow a greater generality for the problems we consider, we are going to develop a numerical scheme, considering μ_+, μ_-, σ_- as sufficiently smooth, positive functions in space. The uniqueness results stated in Theorem 2 are easily carried over to this new situation, but there are no straight forward extensions of the existence proofs.

The discretization procedure can be subdivided into two parts:

- Discretization of $\triangle\phi = 0$ in D_+^c.
- Discretization of $\mathrm{curl}(\rho_- \mathrm{curl} H_-) = i\omega\mu_- H_-$ in D_-.

In both cases we will use a finite-volume approach. Finite-volume methods have the advantage of being easily adapted to cylindrical grids, which play a central role in applications (most electric machines actually are of cylindric shape). For ease of exposition, we will restrict ourselves here to cartesian coordinate systems.

For the discretization of $\triangle\phi = 0$ we use a simple cartesian grid in D_+^c. Let $p_{ijk} = (x_i, y_j, z_k)^T$, - $i, j, k = 1, 2, \ldots$, be the locations of the nodes of the grid, assuming $x_i < x_j$, $y_i < y_j$, $z_i < z_j$ for $i < j$. For the discrete values of the approximation of ϕ in the node p_{ijk} we use the notation ϕ_{ijk}.

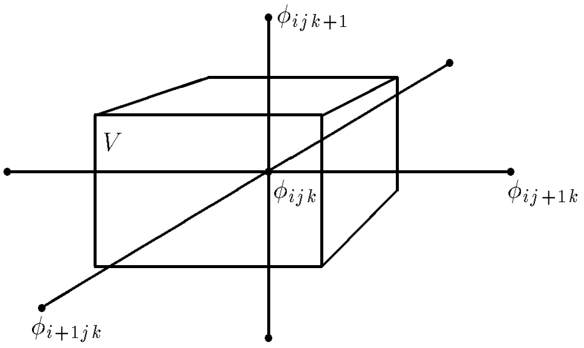

Fig. 3

Now let us consider $\triangle\phi = 0$. Integrating the equation over the volume V of Fig. 3 (a block whose surfaces are parallel to the x-y, x-z resp. y-z plane and intersect the vertices of the grid exactly in the midth between two nodes) and applying the theorem of Gauß, we get

$$0 = \int_V \triangle\phi \, dv = \int_{\partial V} \partial_n \phi \, ds.$$

We approximate the integral on the right hand side assuming $\partial_n\phi$ to be constant on each of the six parts of ∂V and expressing $\partial_n\phi$ in terms of the ϕ_{ijk}:

$$\begin{aligned}
\int_{\partial V} \partial_n \phi \, ds \quad \approx \quad & \frac{1}{4}(\triangle y_1 + \triangle y_2)(\triangle z_1 + \triangle z_2)(\frac{\phi_{i+1jk} - \phi_{ijk}}{\triangle x_2} - \frac{\phi_{ijk} - \phi_{i-1jk}}{\triangle x_1}) \\
+ \quad & \frac{1}{4}(\triangle x_1 + \triangle x_2)(\triangle z_1 + \triangle z_2)(\frac{\phi_{ij+1k} - \phi_{ijk}}{\triangle y_2} - \frac{\phi_{ijk} - \phi_{ij-1k}}{\triangle y_1}) \\
+ \quad & \frac{1}{4}(\triangle x_1 + \triangle x_2)(\triangle y_1 + \triangle y_2)(\frac{\phi_{ijk+1} - \phi_{ijk}}{\triangle z_2} - \frac{\phi_{ijk} - \phi_{ijk-1}}{\triangle z_1}),
\end{aligned}$$

where we used $\triangle x_1 = x_i - x_{i-1}$, $\triangle x_2 = x_{i+1} - x_i$, $\triangle y_1 = y_i - y_{i-1}$, $\triangle y_2 = y_{i+1} - y_i$, $\triangle y_1 = y_i - y_{i-1}$, $\triangle y_2 = y_{i+1} - y_i$. This delivers the usual seven–point formula for the Laplace operator in R^3.

The discretization of the operator $\mathrm{curl}(\rho \ \mathrm{curl} H) - i\omega\mu H$ is less straight forward and much more technical. Therefore, we only present a detailed derivation of the equation obtained for the first component H^x of $H = (H^x, H^y, H^z)^T$. The corresponding approximations for the other two components are obtained in exactly the same way. For the sake of completeness, the corresponding results are given below.

For the definition of the discrete approximates we use a certain kind of staggered grid in D_-. Let p_{ijk} denote the nodes of a cartesian grid in D_-. The value of H^x_{ijk} (H^y_{ijk}, H^z_{ijk}) is defined in the midth of the vertex connecting p_{ijk} and p_{i+1jk} (p_{ij+1k}, p_{ijk+1}). Moreover we assume the material coefficients ρ and μ to be constant in each block of the grid and define ρ_{ijk}, μ_{ijk} to be the values of the block whose corner with smallest indices is p_{ijk} (Fig. 4).

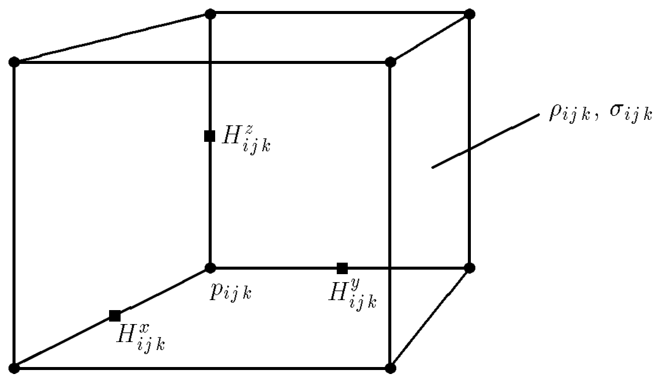

Fig. 4

Consider now the rectangle A, which is orthogonal to the vertex between p_{ijk} and p_{i+1jk} and whose center coincides with the mid point of this vertex (Fig. 5).

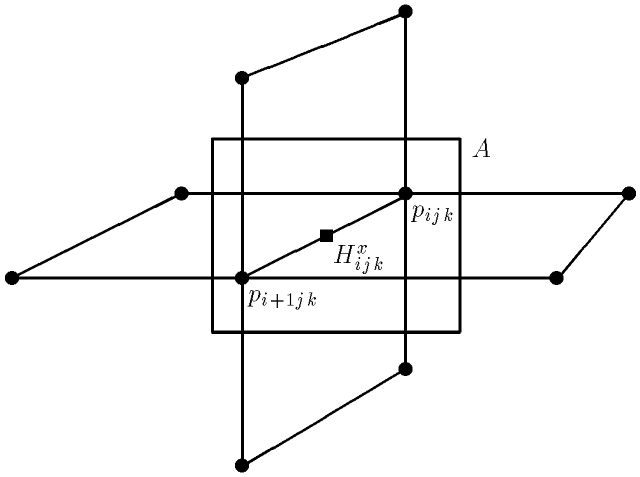

Fig. 5

We multiply $\mathrm{curl}(\rho\ \mathrm{curl} H) - i\omega\mu H = 0$ with the first unit vector $n = (1,0,0)^T$ and integrate over A. Applying Stokes' Theorem we deduce

$$i\omega \int_A \mu H\, ds = \int_A n \cdot \mathrm{curl}(\rho\ \mathrm{curl} H)\, ds = \int_{\partial A} \tau \cdot (\rho\ \mathrm{curl}\ H)\, dl. \tag{12}$$

Assuming $\mathrm{curl} H$ to be constant on each of the four vertices $V_1, \ldots, V_4$ of ∂A, we obtain the following approximation for the integral on the right hand side

$$\int_{\partial A} \tau \cdot (\rho\ \mathrm{curl}\ H)\, dl \quad \approx \quad \sum_{i=1}^{4} (\mathrm{curl} H)_i \cdot \int_{V_i} \tau\rho\, dl,$$

$$\sum_{i=1}^{4} (\mathrm{curl} H)_i \cdot \int_{V_i} \tau\rho \, dl = (\mathrm{curl} H)_1 \frac{\triangle y_1 \; \rho_{ij-1k-1} + \triangle y_2 \; \rho_{ijk-1}}{2}$$
$$- (\mathrm{curl} H)_3 \frac{\triangle y_1 \; \rho_{ij-1k} + \triangle y_2 \; \rho_{ijk}}{2}$$
$$+ (\mathrm{curl} H)_2 \frac{\triangle z_1 \; \rho_{ijk-1} + \triangle z_2 \; \rho_{ijk}}{2}$$
$$- (\mathrm{curl} H)_4 \frac{\triangle z_1 \; \rho_{ij-1k-1} + \triangle z_2 \; \rho_{ij-1k}}{2}.$$

Using a simple difference approximation yields

$$(\mathrm{curl} H)_1 \approx \frac{H^x_{ijk} - H^x_{ijk-1}}{\triangle z_1} - \frac{H^z_{i+1jk-1} - H^z_{ijk-1}}{\triangle x_2},$$
$$(\mathrm{curl} H)_3 \approx \frac{H^x_{ijk+1} - H^x_{ijk}}{\triangle z_2} - \frac{H^z_{i+1jk} - H^z_{ijk}}{\triangle x_2},$$
$$(\mathrm{curl} H)_2 \approx \frac{H^y_{i+1jk} - H^y_{ijk}}{\triangle x_2} - \frac{H^x_{ij+1k} - H^x_{ijk}}{\triangle y_2},$$
$$(\mathrm{curl} H)_4 \approx \frac{H^y_{i+1j-1k} - H^y_{ij-1k}}{\triangle x_2} - \frac{H^x_{ijk} - H^x_{ij-1k}}{\triangle y_1}.$$

For the left hand side of (12) we derive similarly

$$\int_A \mu H \, ds \approx \frac{\triangle y_2 \triangle z_2 \; \mu_{ijk} + \triangle y_1 \triangle z_2 \; \mu_{ij-1k} + \triangle y_1 \triangle z_1 \; \mu_{ij-1k-1} + \triangle y_2 \triangle z_1 \; \mu_{ijk-1}}{4} H^x_{ijk}.$$

Finally by introducing

$$r^x_{ijk} = \frac{\triangle x_1 \; \rho_{i-1jk} + \triangle x_2 \; \rho_{ijk}}{2},$$
$$r^y_{ijk} = \frac{\triangle y_1 \; \rho_{ij-1k} + \triangle y_2 \; \rho_{ijk}}{2},$$
$$r^z_{ijk} = \frac{\triangle z_1 \; \rho_{ijk-1} + \triangle z_2 \; \rho_{ijk}}{2},$$

we obtain for the x-component H^x_{ijk}:

$$0 = \Big(\frac{r^y_{ijk-1}}{\triangle z_1} + \frac{r^y_{ijk}}{\triangle z_2} + \frac{r^z_{ij-1k}}{\triangle y_1} + \frac{r^z_{ijk}}{\triangle y_2}$$
$$- i\omega \frac{\triangle y_2 \triangle z_2 \; \mu_{ijk} + \triangle y_1 \triangle z_2 \; \mu_{ij-1k} + \triangle y_1 \triangle z_1 \; \mu_{ij-1k-1} + \triangle y_2 \triangle z_1 \; \mu_{ijk-1}}{4} \Big) H^x_{ijk}$$
$$- \frac{r^z_{ij-1k}}{\triangle y_1} H^x_{ij-1k} - \frac{r^z_{ijk}}{\triangle y_2} H^x_{ij+1k} - \frac{r^y_{ijk-1}}{\triangle z_1} H^x_{ijk-1} - \frac{r^y_{ijk}}{\triangle z_2} H^x_{ijk+1}$$
$$+ \frac{r^z_{ijk}}{\triangle x_2} (H^y_{i+1jk} - H^y_{ijk}) - \frac{r^z_{ij-1k}}{\triangle x_2} (H^y_{i+1j-1k} - H^y_{ij-1k})$$
$$+ \frac{r^y_{ijk}}{\triangle x_2} (H^z_{i+1jk} - H^z_{ijk}) - \frac{r^y_{ijk-1}}{\triangle x_2} (H^z_{i+1jk-1} - H^z_{ijk-1}).$$

Using the same arguments as above, we get for the y-component H^y_{ijk}

$$0 = \Big(\frac{r^x_{ijk-1}}{\triangle z_1} + \frac{r^x_{ijk}}{\triangle z_2} + \frac{r^z_{i-1jk}}{\triangle x_1} + \frac{r^z_{ijk}}{\triangle x_2}$$

$$
\begin{aligned}
& -i\omega \frac{\triangle x_2 \triangle z_2\ \mu_{ijk} + \triangle x_1 \triangle z_2\ \mu_{i-1jk} + \triangle x_1 \triangle z_1\ \mu_{i-1jk-1} + \triangle x_2 \triangle z_1\ \mu_{ijk-1}}{4}) H^y_{ijk} \\
- \quad & \frac{r^z_{i-1jk}}{\triangle x_1} H^y_{i-1jk} - \frac{r^z_{ijk}}{\triangle x_2} H^y_{i+1jk} - \frac{r^x_{ijk-1}}{\triangle z_1} H^y_{ijk-1} - \frac{r^x_{ijk}}{\triangle z_2} H^y_{ijk+1} \\
+ \quad & \frac{r^z_{ijk}}{\triangle y_2}(H^x_{ij+1k} - H^x_{ijk}) - \frac{r^z_{i-1jk}}{\triangle y_2}(H^x_{i-1j+1k} - H^x_{i-1jk}) \\
+ \quad & \frac{r^x_{ijk}}{\triangle y_2}(H^z_{ij+1k} - H^z_{ijk}) - \frac{r^x_{ijk-1}}{\triangle y_2}(H^z_{ij+1k-1} - H^z_{ijk-1})
\end{aligned}
$$

and for the z-component H^z_{ijk}

$$
\begin{aligned}
0 \quad = \quad & (\frac{r^x_{ij-1k}}{\triangle y_1} + \frac{r^x_{ijk}}{\triangle y_2} + \frac{r^y_{i-1jk}}{\triangle x_1} + \frac{r^y_{ijk}}{\triangle x_2} \\
& -i\omega \frac{\triangle x_2 \triangle y_2\ \mu_{ijk} + \triangle x_1 \triangle y_2\ \mu_{i-1jk} + \triangle x_1 \triangle y_1\ \mu_{i-1j-1k} + \triangle x_2 \triangle y_1\ \mu_{ij-1k}}{4}) H^z_{ijk} \\
- \quad & \frac{r^y_{i-1jk}}{\triangle x_1} H^z_{i-1jk} - \frac{r^y_{ijk}}{\triangle x_2} H^z_{i+1jk} - \frac{r^x_{ij-1k}}{\triangle y_1} H^z_{ij-1k} - \frac{r^x_{ijk}}{\triangle y_2} H^z_{ij+1k} \\
+ \quad & \frac{r^y_{ijk}}{\triangle z_2}(H^x_{ijk+1} - H^x_{ijk}) - \frac{r^y_{i-1jk}}{\triangle z_2}(H^x_{i-1jk+1} - H^x_{i-1jk}) \\
+ \quad & \frac{r^x_{ijk}}{\triangle z_2}(H^y_{ijk+1} - H^y_{ijk}) - \frac{r^x_{ij-1k}}{\triangle z_2}(H^y_{ij-1k+1} - H^y_{ij-1k}).
\end{aligned}
$$

The coupling conditions on Γ and the boundary conditions on Γ^c are straight forwardly discretized.

5 Krylov subspace methods

In this chapter, we give a detailed description of the four most popular Krylov subspace methods, BiCG, CGS, BiCGSTAB and GMRES. For the conjugate gradient like procedures we follow the lines of [8] and [9]. The derivation of GMRES goes back to [7]. In contrast to the original papers, we consider the case of complex linear systems, which is necessary in order to apply the algorithms to the systems arising from our transmission boundary-value problem.

5.1 Conjugate gradient like methods

5.1.1 The polynomial conjugate gradient method

We start our considerations with the simplest case, the original conjugate gradient algorithm, which was introduced by Hestenes and Stiefel for the solution of linear systems $Ax = b$, where A is a hermitian ($A = A^H = \bar{A}^T$), positive definite matrix.

If x_0 denotes an initial guess for the solution x, the iterates are computed as

$$
\begin{aligned}
r_0 &= b - Ax_0, \\
p_0 &= r_0, \\
\\
r_{i+1} &= r_i - \alpha_i A p_i, \\
p_{i+1} &= r_{i+1} + \beta_i p_i, \qquad (13) \\
x_{i+1} &= x_i + \alpha_i p_i, \\
\\
\alpha_i &= \frac{(r_i, r_i)}{(p_i, Ap_i)},
\end{aligned}
$$

$$\beta_i = \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)},$$

where $(u, v) = u^H v = \bar{u}^T v$.

Since $p_0 = r_0$, it is easy to see, that r_i and p_i can be written as polynomials ϕ_i and ψ_i of degree i , applied to the matrix A and multiplied by r_0 from the right:

$$r_i = \phi_i(A) r_0, \qquad p_i = \psi_i(A) r_0.$$

Substituting this representation into (13), we obtain recurrence relations for the polynomials ϕ_i, ψ_i, which are similar to those for r_i and p_i:

$$\begin{aligned}
\phi_0 &= 1, \\
\psi_0 &= 1, \\
\\
\phi_{i+1} &= \phi_i - \alpha_i x \psi_i, \\
\psi_{i+1} &= \phi_{i+1} + \beta_i \psi_i, \\
\\
\alpha_i &= \frac{< \phi_i, \phi_i >}{< \psi_i, x\psi_i >}, \\
\\
\beta_i &= \frac{< \phi_{i+1}, \phi_{i+1} >}{< \phi_i, \phi_i >}, \\
\\
< \phi, \psi > &= (\phi(A) r_0, \psi(A) r_0).
\end{aligned} \tag{14}$$

The relations (14) are usually called *polynomial CG algorithm.* Since $A = A^H$, the coefficients α_i, β_i are real. Therefore $\phi_i, \psi_i \in \Pi(R)$, the space of polynomials with real coefficients and $< \cdot, \cdot >$ defines a bilinear form on $\Pi(R)$.

5.1.2 Generalization

For the generalization of the standard Conjugate Gradient method, we reverse the order of argumentation used in the last section.

In 5.1.1, we started with the vectorial algorithm (13) and derived a corresponding polynomial form (14). Now we proceed exactly in the opposite direction. Starting with a modification of (14) we construct corresponding vectorial recurrence relations, which will differ from (13).

To get an idea, how to modify the polynomial method, we take again a look at (14). We observe, that all the information about the underlying linear system is contained in the bilinearform $< \cdot, \cdot >$. For A hermitian and positive definite, we get convergence of (13) (and therefore of (14)) and in the absence of rounding off errors, the solution of $Ax = b$ is reached in at most n steps, where n denotes the number of unknowns. In the convergence proof some special properties of $< \cdot, \cdot >$ are exploited, which do not hold for indefinite A. Moreover, it can be shown [1], that in this case it is not possible to construct an algorithm being of the same type as (13) (resp. (14)), having the same convergence properties.

Therefore, for the generalization of the standard methods to indefinite or non-hermitian matrices, we have to choose one of the following alternatives:

- We could try to keep all the convergence results unchanged. But then we are forced to alter the structure of the recurrence relations significantly, so that the number of operations and the amount of storage necessary in each step can not be compared with the corresponding values of (13).

- We preserve the easy structure of the recurrence relations, giving up a part of the theoretical results which hold for (13).

The methods we are going to derive here are of the second type. We modify the bilinear form $< \cdot, \cdot >$ in (14) and require the resulting method to fulfill the orthogonality relations of the Hestenes-Stiefel method

$$\begin{aligned} (r_i, r_j) &= c\,\delta_{ij}, \\ (p_i, Ap_j) &= c'\delta_{ij}, \end{aligned}$$

which may be written in polynomial representation as

$$\begin{aligned} < \phi_i, \phi_j > &= c\,\delta_{ij}, \\ < \psi_i, x\psi_j > &= c'\delta_{ij}. \end{aligned}$$

The essential observation in the convergence proof of (13) is, that due to the positive definiteness of A the constants c and c' are positive. For the reason just mentioned above, we are forced to give up this restriction.

Moreover, we allow complex coefficients for the polynomials ϕ_i and ψ_i and define the modified bilinear form over $\Pi(C)$, the space of polynomials with complex coefficients.

In the following theorem [8], we will see, that a large class of bilinear forms fulfills these requirements. The proof of this theorem is rather elementary and therefore omitted.

Theorem 3 *Let $[\cdot, \cdot]$ be an arbitrary, symmetric bilinearform on $\Pi(C)$, the space of polynomials with complex coefficients. If*

$$[\phi\chi, \psi] = [\phi, \chi\psi] \quad \forall \phi, \psi, \chi \in \Pi(C),$$

we get for the polynomials ϕ_i and ψ_i, which are computed from (14) by using $[\cdot, \cdot]$ instead of $< \cdot, \cdot >$:

$$[\phi_i, \phi_j] = c\,\delta_{ij}, \qquad [\psi_i, x\psi_j] = c'\delta_{ij} \qquad c, c' \in C.$$

In the next chapter we give an example of a bilinearform which fits into the framework of Theorem 3, but is different from the one given in (14). We then present three different ways how to translate the polynomial algorithm based on this new bilinearform into vectorial recurrence relations, thus obtaining the three most popular generalized conjugate gradient methods.

5.1.3 The bi-conjugate gradient method, BiCG

We consider the regular, complex matrix A and define a bilinear form $[\cdot, \cdot]$ on $\Pi(C)$ by

$$[\phi, \psi] = (\bar{\phi}(A^H)r_0', \psi(A)r_0), \qquad (u, v) = \bar{u}^H v, \tag{15}$$

where $\bar{\phi}(x) = \sum_{k=0}^{m} \bar{\lambda}_k x^k$ is the polynomial with the complex conjugate coefficients of $\phi(x) = \sum_{k=0}^{m} \lambda_k x^k$, $r_0 = b - Ax_0$ is the residual vector corresponding to the initial guess x_0 and $r_0' \in C^n$ is some nonzero vector.

Since

$$[\phi, \psi] = (\bar{\phi}(A^H)r_0', \psi(A)r_0) = (r_0')^H \phi(A)\psi(A)r_0$$

and

$$\phi(A)\psi(A) = \psi(A)\phi(A) \qquad \forall\ \phi, \psi \in \Pi(C),$$

we get

$$[\phi, \psi] = [\psi, \phi], \qquad [\phi\chi, \psi] = [\phi, \chi\psi],$$

so that the two assumptions of Theorem 3 are fulfilled. Replacing the bilinearform from (14) by (15), we want to construct the corresponding vectorial recurrence relations. We use as above

$$r_i = \phi_i(A)r_0, \qquad p_i = \psi_i(A)r_0$$

and conclude

$$r_{i+1} = r_i - \alpha_i A p_i, \qquad p_{i+1} = r_{i+1} + \beta_i p_i.$$

For the coefficients α_i, β_i we have

$$\alpha_i = \frac{(\bar{\phi}_i(A^H)r_0', \phi_i(A)r_0)}{(\bar{\psi}_i(A^H)r_0', A\psi_i(A)r_0)}, \qquad \beta_i = \frac{(\bar{\phi}_{i+1}(A^H)r_0', \phi_{i+1}(A)r_0)}{(\bar{\phi}_i(A^H)r_0', \phi_i(A)r_0)}.$$

The vectors $\bar{\phi}_i(A^H)r_0'$, $\bar{\psi}_i(A^H)r_0'$, occuring as factors on the left hand side of the scalar products, cannot be computed in terms of r_i and p_i. Therefore, we have to introduce additional vectors

$$r_i' = \bar{\phi}_i(A^H)r_0', \qquad p_i' = \bar{\psi}_i(A^H)r_0'.$$

Since $\bar{\phi}_i, \bar{\psi}_i$ are the polynomials with the complex conjugate coefficients of ϕ_i, ψ_i, we derive from

$$\phi_{i+1} = \phi_i - \alpha_i x \psi_i, \qquad \psi_{i+1} = \phi_{i+1} + \beta_i \psi_i,$$

the equations

$$\bar{\phi}_{i+1} = \bar{\phi}_i - \bar{\alpha}_i x \bar{\psi}_i, \qquad \bar{\psi}_{i+1} = \bar{\phi}_{i+1} + \bar{\beta}_i \bar{\psi}_i,$$

and thus

$$r_{i+1}' = r_i' - \bar{\alpha}_i A^H p_i', \qquad p_{i+1}' = r_{i+1}' + \bar{\beta}_i p_i'.$$

Since $r_i = b - Ax_i$, $r_{i+1} = b - Ax_{i+1}$, the iterates x_i are defined by

$$x_{i+1} = x_i + \alpha_i p_i.$$

So we finally got from (14), using the bilinear form (15) the so called *Bi-Conjugate Gradient Method,* **BiCG** [2]:

$$\begin{aligned}
r_0 &= b - Ax_0, \\
p_0 &= r_0, \\
r_0' &\in C^n, \qquad (r_0', r_0) \neq 0, \\
p_0' &= r_0' \\[1em]
r_{i+1} &= r_i - \alpha_i A p_i, \\
p_{i+1} &= r_{i+1} + \beta_i p_i, \\[1em]
r_{i+1}' &= r_i' - \bar{\alpha}_i A^H p_i', \\
p_{i+1}' &= r_{i+1}' + \bar{\beta}_i p_i', \\[1em]
x_{i+1} &= x_i + \alpha_i p_i, \\[1em]
\alpha_i &= \frac{(r_i', r_i)}{(p_i', Ap_i)}, \\[1em]
\beta_i &= \frac{(r_{i+1}', r_{i+1})}{(r_i', r_i)}.
\end{aligned} \tag{16}$$

Remarks

- The usual choice for r'_0 is $r'_0 = r_0$.
- The BiCG method has the major disadvantage of involving matrix-vector products of the type Av and $A^H v$, so that the performance on vector–computers and especially massively parallel machines is very poor.
- As simple examples show, the algorithm might stop ($\alpha_i = 0$) without having reached the true solution, which means that $r_i \neq 0$. This is usually called a *break down*.
- If the algorithm does not break down, it is easily shown, that the true solution is reached after at most n steps, n being the dimension of the system.
- To control the accuracy of the actual iterate, one usually considers the euclidean norm $\|r_i\|_2$ of the residual vector r_i. If the BiCG method converges, $\|r_i\|_2$ does in general not decrease monotonically, but shows heavy oscillations when tending to zero.

5.1.4 The squared conjugate gradient method, CGS

Taking again a look at the assumption on (15) in Theorem 3, we see, that $[\phi, \psi] = [1, \phi\psi]$. Therefore we may write for α_i, β_i

$$\alpha_i = \frac{[1, \phi_i^2]}{[1, x\psi_i^2]}, \qquad \beta_i = \frac{[1, \phi_{i+1}^2]}{[1, \phi_i^2]},$$

so that only the squares of the polynomials ϕ_i, ψ_i are involved.

By simply taking the squares of (14), we derive the following recurrence relations for ϕ_i^2, ψ_i^2

$$\begin{aligned}
\phi_{i+1}^2 &= \phi_i^2 - \alpha_i x (2\phi_i - \alpha_i x \psi_i)\psi_i \\
&= \phi_i^2 - \alpha_i x (\phi_i + \phi_{i+1})\psi_i \\
&= \phi_i^2 - \alpha_i x (\phi_i^2 + \beta_{i-1}\phi_i\psi_{i-1} + \phi_{i+1}\psi_i),
\end{aligned}$$

$$\begin{aligned}
\psi_{i+1}^2 &= \phi_{i+1}^2 + 2\beta_i \phi_{i+1}\psi_i + \beta_i^2 \psi_i^2 \\
&= \phi_{i+1}^2 + \beta_i \phi_{i+1}\psi_i + \beta_i(\phi_{i+1}\psi_i + \beta_i \psi_i^2)
\end{aligned}$$

and for the mixed products $\phi_i \psi_{i-1}$

$$\phi_{i+1}\psi_i = (\phi_i - \alpha_i x \psi_i)\psi_i = \phi_i^2 + \beta_{i-1}\phi_i\psi_{i-1} - \alpha_i x \psi_i^2$$

Defining the polynomial χ_i as $\chi_i = \phi_i \psi_{i-1}$ we finally arrive at the definition of the *squared polynomial algorithm*

$$\begin{aligned}
\phi_0^2 &= 1, \\
\psi_0^2 &= 1, \\
\chi_0 &= 0,
\end{aligned}$$

$$\begin{aligned}
\chi_{i+1} &= \phi_i^2 + \beta_{i-1}\chi_i - \alpha_i x \psi_i^2, \\
\phi_{i+1}^2 &= \phi_i^2 - \alpha_i x (\phi_i^2 + \beta_{i-1}\chi_i + \chi_{i+1}), \\
\psi_{i+1}^2 &= \phi_{i+1}^2 + \beta_i \chi_{i+1} + \beta_i(\chi_{i+1} + \beta_i \psi_i^2),
\end{aligned}$$

or equivalently

$$\begin{aligned}
\psi_{-1}^2 &= 0, \\
\chi_0 &= 0,
\end{aligned}$$

$$\begin{aligned}
\phi_0^2 &= 1, \\
\beta_{-1} &= 0,
\end{aligned}$$

$$\begin{aligned}
\psi_i^2 &= \phi_i^2 + \beta_{i-1}\chi_i + \beta_{i-1}(\chi_i + \beta_{i-1}\psi_{i-1}^2), \\
\chi_{i+1} &= \phi_i^2 + \beta_{i-1}\chi_i - \alpha_i x \psi_i^2, \qquad (17) \\
\phi_{i+1}^2 &= \phi_i^2 - \alpha_i x(\phi_i^2 + \beta_{i-1}\chi_i + \chi_{i+1}),
\end{aligned}$$

$$\begin{aligned}
\alpha_i &= \frac{[1, \phi_i^2]}{[1, x\psi_i^2]}, \\
\beta_i &= \frac{[1, \phi_{i+1}^2]}{[1, \phi_i^2]}.
\end{aligned}$$

For the transformation to a vectorial algorithm, we use again the bilinear form (15), but in contrast to the last section, we define

$$r_i = \phi_i^2(A) r_0, \qquad p_i = \psi_i^2(A) r_0, \qquad q_i = \chi_i(A) r_0.$$

Substituting r_i, p_i, q_i in (17), we finally obtain the *squared Conjugate Gradient Method,* **CGS**

$$\begin{aligned}
p_{-1} &= 0, \\
q_0 &= 0, \\
r_0 &= b - Ax_0, \\
r'_0 &\in C^n, \qquad (r'_0, r_0) \neq 0, \\
\beta_{-1} &= 0,
\end{aligned}$$

$$\begin{aligned}
p_i &= r_i + \beta_{i-1} q_i + \beta_{i-1}(q_i + \beta_{i-1} p_{i-1}), \\
q_{i+1} &= r_i + \beta_{i-1} q_i - \alpha_i A p_i, \qquad (18) \\
r_{i+1} &= r_i - \alpha_i A(r_i + \beta_{i-1} q_i + q_{i+1}), \\
x_{i+1} &= x_i + \alpha_i(r_i + \beta_{i-1} q_i + q_{i+1}),
\end{aligned}$$

$$\begin{aligned}
\alpha_i &= \frac{(r'_0, r_i)}{(r'_0, Ap_i)}, \\
\beta_i &= \frac{(r'_0, r_{i+1})}{(r'_0, r_i)}.
\end{aligned}$$

Remarks

- Usually $r'_0 = r_0$, as for BiCG.
- For CGS, we might also have break downs.
- Without break down, the solution is obtained after at most n steps.
- There exists the same problem of oscillating $\|r_i\|_2$ as in the last section.
- For CGS, we don't have to compute products involving A^H.

5.1.5 The BiCGSTAB method

Comparing the two methods we derived up to now, we observe that

$$r_i^{CGS} = \phi_i^2(A) r_0 = \phi_i(A) r_i^{BiCG},$$

$$p_i^{CGS} = \psi_i^2(A) r_0 = \psi_i(A) p_i^{BiCG}.$$

Numerical experiments show, that

- CGS usually converges faster than BiCG.
- Oscillations of $\|r_i^{CGS}\|_2$ are larger than oscillations of $\|r_i^{BiCG}\|_2$.

These two effects are essentially caused by the additional multiplication of r_i^{BiCG} by $\phi_i(A)$ in each step. Therefore, we have to ask, whether ϕ_i is a good choice for this additional multiplication or do there exist other polynomials, delivering better results ?

For the following investigations, we consider polynomials $\lambda_i \in \Pi(C)$ of the type

$$\begin{aligned} \lambda_0 &= 1, \\ \lambda_{i+1} &= (1 - \omega_i x)\, \lambda_i, \end{aligned} \tag{19}$$

where $\omega_i \in \Pi(C)$ are free parameters, which will be determined below. The degree of λ_i is less or equal to i.

Using the polynomials $\phi_i,\ \psi_i$ obtained from (14), (15), we define

$$r_i = (\lambda_i \phi_i)(A) r_0, \qquad p_i = (\lambda_i \psi_i)(A) r_0.$$

Note that in contrast to CGS, the additional multiplication is done using the *same* polynomial in both cases.

Before we can start the construction of the iteration procedure for $r_i,\ p_i$, we have to derive the recurrence relations for the two polynomials $\lambda_i\phi_i,\ \lambda_i\psi_i$. This can be done without restricting ourselves to a special bilinear form. Instead of (15) we can also use any other bilinear form on $\Pi(C)$, which fulfills the assumption of Theorem 3. Taking a look at (14) and keeping in mind the definition of the polynomials λ_i, we have

$$\lambda_{i+1}\phi_{i+1} = \lambda_{i+1}(\phi_i - \alpha_i x \psi_i) = (1 - \omega_i x)(\lambda_i\phi_i - \alpha_i x \lambda_i \psi_i),$$

$$\lambda_{i+1}\psi_{i+1} = \lambda_{i+1}(\phi_{i+1} + \beta_i \psi_i) = \lambda_{i+1}\phi_{i+1} + \beta_i (1 - \omega_i x)\lambda_i \psi_i$$

with the usual coefficients

$$\alpha_i = \frac{[\phi_i, \phi_i]}{[\psi_i, x\psi_i]}, \qquad \beta_i = \frac{[\phi_{i+1}, \phi_{i+1}]}{[\phi_i, \phi_i]}.$$

To ensure a simple structure of the corresponding vectorial algorithm, we try to express the coefficients $\alpha_i,\ \beta_i$ in terms of $\lambda_i\phi_i,\ \lambda_i\psi_i$. For this modification, we have to investigate the behaviour of the polynomials $\phi_i(x) = \sum_{j=0}^{i} \phi_{ij} x^j$, $\psi_i(x) = \sum_{j=0}^{i} \psi_{ij} x^j$ and $\lambda_i(x) = \sum_{j=0}^{i} \lambda_{ij} x^j$:

- We know, that $\phi_i,\ \psi_i$ are polynomials of degree i and

  $$\phi_{i+1} = \phi_i - \alpha_i x \psi_i, \qquad \psi_{i+1} = \phi_{i+1} + \beta_i \psi_i.$$

  The leading coefficients $\phi_{ii},\ \psi_{ii}$ are therefore computed by

  $$\phi_{i+1i+1} = -\alpha_i \psi_{ii}, \qquad \psi_{i+1i+1} = \phi_{i+1i+1}.$$

  Since $\phi_{00} = \psi_{00} = 1$, we get the representation

  $$\phi_{ii} = \psi_{ii} = (-1)^i \prod_{j=0}^{i-1} \alpha_i. \tag{20}$$

- As long as no break down ($\alpha_i = 0$) occurs, the leading coefficients $\phi_{ii},\ \psi_{ii}$ of $\phi_i,\ \psi_i$ are nonzero (20). Therefore, the polynomials ϕ_i, $i = 0, \ldots, m$, resp. ψ_i, $i = 0, \ldots, m$ are linear independent and form a basis of $\Pi_m(C)$, the space of polynomials of degree less or equal m with complex coefficients.

- The leading coefficient of λ_i is

$$\lambda_{ii} = (-1)^i \prod_{j=0}^{i-1} \omega_i. \tag{21}$$

Now we consider $[\phi_i, \phi_i]$. Using Theorem 3, we know that $[\phi_j, \phi_i] = c\,\delta_{ij}$ and obtain

$$[\phi_j, \phi_i] = 0, \qquad j = 0, \ldots, i-1.$$

But according to the above remark, $\{\phi_0, \ldots, \phi_{i-1}\}$ is a basis of $\Pi_{i-1}(C)$, so that

$$[\mu, \phi_i] = 0, \qquad \forall\ \mu \in \Pi_{i-1}(C).$$

Since $\phi_i(x) = \sum_{j=0}^{i} \phi_{ij} x^j = \tilde{\phi}_i + \phi_{ii} x^i,\ \tilde{\phi}_i \in \Pi_{i-1}(C)$ and $\lambda_i(x) = \sum_{j=0}^{i} \lambda_{ij} x^j = \tilde{\lambda}_i + \lambda_{ii} x^i,\ \tilde{\lambda}_i \in \Pi_{i-1}(C)$, we obtain

$$[\phi_i, \phi_i] = [\tilde{\phi}_i + \phi_{ii} x^i, \phi_i] = \phi_{ii}\,[x^i, \phi_i],$$
$$[\lambda_i, \phi_i] = [\tilde{\lambda}_i + \lambda_{ii} x^i, \phi_i] = \lambda_{ii}\,[x^i, \phi_i]$$

and therefore

$$[\phi_i, \phi_i] = \frac{\phi_{ii}}{\lambda_{ii}}\,[1, \lambda_i \phi_i].$$

Proceeding in the same way, we also derive

$$[\psi_i, x\psi_i] = \frac{\psi_{ii}}{\lambda_{ii}}\,[1, x\lambda_i\psi_i] = \frac{\phi_{ii}}{\lambda_{ii}}\,[1, x\lambda_i\psi_i],$$

and

$$\alpha_i = \frac{[\phi_i, \phi_i]}{[\psi_i, x\psi_i]} = \frac{[1, \lambda_i\phi_i]}{[1, x\lambda_i\psi_i]},$$
$$\beta_i = \frac{[\phi_{i+1}, \phi_{i+1}]}{[\phi_i, \phi_i]} = \frac{\phi_{i+1i+1}\lambda_{ii}}{\lambda_{i+1i+1}\phi_{ii}}\,\frac{[1, \lambda_{i+1}\phi_{i+1}]}{[1, \lambda_i\phi_i]}.$$

Using the representations (20) and (21), we get

$$\alpha_i = \frac{[1, \lambda_i\phi_i]}{[1, x\lambda_i\psi_i]}, \qquad \beta_i = \frac{\alpha_i}{\omega_i}\,\frac{[1, \lambda_{i+1}\phi_{i+1}]}{[1, \lambda_i\phi_i]}.$$

With the above analysis, we now obtain the *stabilized polynomial algorithm*

$$\begin{aligned}
\psi_{-1} &= 0,\\
\phi_0 &= 1,\\
\lambda_0 &= 1,\\
\beta_{-1} &= 0,\\
\omega_{-1} &= 0,\\
\\
\lambda_i\psi_i &= \lambda_i\phi_i + \beta_{i-1}(\lambda_{i-1}\psi_{i-1} - \omega_{i-1}x\lambda i - 1\psi_{i-1}),\\
\lambda_{i+1}\phi_{i+1} &= \lambda_i\phi_i - \alpha_i x\lambda_i\psi_i - \omega_i x(\lambda_i\phi_i - \alpha_i x\lambda_i\psi_i),\\
\\
\alpha_i &= \frac{[1, \lambda_i\phi_i]}{[1, x\lambda_i\psi_i]},\\
\beta_i &= \frac{\alpha_i}{\omega_i}\,\frac{[1, \lambda_{i+1}\phi_{i+1}]}{[1, \lambda_i\phi_i]}.
\end{aligned} \tag{22}$$

Using again the bilinear form (15) and $r_i = (\lambda_i\phi_i)(A)r_0,\ p_i = (\lambda_i\psi_i)(A)r_0$, we easily compute the corresponding vectorial recurrence relations:

$$\begin{aligned} p_i &= r_i + \beta_{i-1}(p_{i-1} - \omega_{i-1}Ap_{i-1}), \\ r_{i+1} &= r_i - \alpha_i Ap_i - \omega_i A(r_i - \alpha_i Ap_i), \end{aligned}$$

$$\alpha_i = \frac{(r'_0, r_i)}{(r'_0, Ap_i)}, \qquad \beta_i = \frac{\alpha_i}{\omega_i}\frac{(r'_0, r_{i+1})}{(r'_0, r_i)}.$$

To determine the parameters ω_i, we consider the vector r_{i+1} as a function depending on ω_i and try to minimize $\|r_{i+1}\|_2$. Using $s = r_i - \alpha_i Ap_i$ and $t = As$, we may write r_{i+1} as

$$r_{i+1} = s - \omega_i t,$$

so that we have to minimize

$$\|t\omega_i - s\|_2.$$

The solution of this least square problem is

$$\omega_i = \frac{(t,s)}{(t,t)}, \qquad (u,v) = u^H v.$$

Finally, for the iterates x_i we get, using $r_i = b - Ax_i$

$$x_{i+1} = x_i + \alpha_i p_i + \omega_i(r_i - \alpha_i Ap_i),$$

which completes the derivation of the *stabilized Bi-Conjugate Gradient Method,* **BiCGSTAB**:

$$\begin{aligned} r_0 &= b - Ax_0, \\ r'_0 &\in C^n, \qquad (r'_0, r_0) \neq 0, \\ p_{-1} &= 0, \\ \beta_{-1} &= 0, \\ \omega_{-1} &= 0, \end{aligned}$$

$$\begin{aligned} p_i &= r_i + \beta_{i-1}(p_{i-1} - \omega_{i-1}Ap_{i-1}), \\ r_{i+1} &= r_i - \alpha_i Ap_i - \omega_i A(r_i - \alpha_i Ap_i), \\ x_{i+1} &= x_i + \alpha_i p_i + \omega_i(r_i - \alpha_i Ap_i), \end{aligned} \tag{23}$$

$$\begin{aligned} \alpha_i &= \frac{(r'_0, r_i)}{(r'_0, Ap_i)}, \\ \beta_i &= \frac{\alpha_i}{\omega_i}\frac{(r'_0, r_{i+1})}{(r'_0, r_i)}, \\ \omega_i &= \frac{(A(r_i - \alpha_i Ap_i), r_i - \alpha_i Ap_i)}{(A(r_i - \alpha_i Ap_i), A(r_i - \alpha_i Ap_i))}. \end{aligned} \tag{24}$$

Remarks

- The usual choice for r'_0 is $r'_0 = r_0$.
- As for the above algorithms, there may occur break downs.
- Without break downs, the true solution is reached after at most n steps.
- The oscillations of $\|r_i\|_2$ for BiCGSTAB can be neglected, compared to those of BiCG and CGS.

5.2 The GMRES method

Taking a look at the algorithms developed above, we recognize, that the residuals in the i^{th} step are computed as

$$\begin{aligned} r_i^{BiCG} &= \phi_i(A)r_0, \\ r_i^{CGS} &= \phi_i^2(A)r_0, \\ r_i^{BiCGSTAB} &= \lambda_i\phi_i(A)r_0, \end{aligned}$$

$$\phi_i \in \Pi_i(C), \qquad \phi_i^2, \lambda_i\phi_i \in \Pi_{2i}(C),$$

$\Pi_k(C)$ denoting the space of polynomials of degree k with complex coefficients. Since $\lambda_0 = \phi_0 = 1$ and $\phi_{i+1} = \phi_i - \alpha_i x\psi_i$, $\lambda_{i+1} = (1 - \omega_i x)\lambda_i$, we see, that $\lambda_i(0) = \phi_i(0) = 1$ for all i. Therefore we have $\phi_i = 1 + x\tilde{\phi}_i$, $\lambda_i = 1 + x\tilde{\lambda}_i$, $\tilde{\phi}_i, \tilde{\lambda}_i \in \Pi_{i-1}(C)$ and

$$r_i^{BiCG} = (1 + x\tilde{\phi}_i)(A)\, r_0 = r_0 + (x\tilde{\phi}_i)(A)\, r_0,$$

$$r_i^{CGS} = (1 + x\tilde{\phi}_i)^2(A)\, r_0 = r_0 + \left(x\tilde{\phi}_i(1 + x\tilde{\phi}_i)\right)(A)\, r_0,$$

$$r_i^{BiCGSTAB} = (1 + x\tilde{\lambda}_i)(1 + x\tilde{\phi}_i)(A)\, r_0 = r_0 + \left(x\tilde{\lambda}_i(1 + x\tilde{\phi}_i)\right)(A)\, r_0.$$

Since $x_i = A^{-1}(b - r_i)$, the iterates are in general given by

$$x_i - x_0 = A^{-1}(r_i - r_0) = \mu(A)r_0, \qquad \mu \in \Pi_{m-1}(C), \tag{25}$$

where the degree $m - 1$ of μ is $i - 1$ for BiCG resp. $2i - 1$ for CGS and BiCGSTAB.

So the nature of BiCG, CGS resp. BiCGSTAB is to modify the initial guess x_0 by an element $y = \mu(A)r_0$ of the Krylov space $K_{m-1}^A(r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}$, which explains the title of this chapter.

Of course, we have to ask, whether the above choice of y is optimal in some sense. Using $||r_i||_2$ as a citerion for the quality of the approximation x_i, easy examples show, that there exist other elements in $K_{m-1}^A(r_0)$, producing smaller norms of the residuals than those obtained from BiCG, CGS or BiCGSTAB.

For GMRES, we now want to find the best possible vector in $K_{m-1}^A(r_0)$. In other words, we set

$$x_m = x_0 + \sum_{j=0}^{m-1} \alpha_j A^j r_0 \tag{26}$$

and determine the parameters α_j (which are the coefficients of the polynomial μ) in order to minimize

$$||r_m||_2 = ||b - Ax_m||_2 = ||b - \sum_{j=0}^{m-1} \alpha_j A^{j+1} r_0||_2. \tag{27}$$

This problem could be attacked using a standard least square method. But as we will see in the remaining part of this section, it is very useful to exploit the special structure of (26) in order to obtain an efficient method for the solution of this problem.

To reformulate (26), we first consider the generation of a Krylov space $K_l^A(r_0)$ of dimension $l+1$. Instead of simply computing the vectors $r_0, Ar_0, \ldots, A^l r_0$, we try to construct an orthonormal basis $\{v_1, \ldots, v_{l+1}\}$ of $K_l^A(r_0)$ using *Arnoldi's method*, which is nothing else than the usual Gram-Schmidt procedure combined with a normalization of the vectors:

$$v_1 = \frac{r_0}{||r_0||_2},$$

$$\text{for } j = 1, \dots, l: \quad \begin{aligned} w_j &= Av_j, \\ \hat{v}_{j+1} &= w_j - \sum_{i=1}^{j} (v_i, w_j) v_i, \\ v_{j+1} &= \frac{\hat{v}_{j+1}}{\|\hat{v}_{j+1}\|_2}. \end{aligned} \tag{28}$$

If for one k the vector $A^k r_0$ depends linearly on $r_0, Ar_0, \dots, A^{k-1} r_0$, this process is going to break down.

Now assume, that k is the first index, where a break down occurs. Thus

$$A^k r_0 = \sum_{i=0}^{k-1} \gamma_i A^i r_0. \tag{29}$$

The coefficient γ_0 does not vanish. Otherwise $A^k r_0 = A \sum_{i=1}^{k-1} \gamma_i A^{i-1} r_0$, so that $A^{k-1} r_0$ would be a linear combination of $r_0, Ar_0, \dots, A^{k-2} r_0$, which contradicts the minimality of k. Using (25) and (28) we get

$$\begin{aligned} r_k &= b - Ax_k \\ &= b - A(x_0 + \sum_{i=0}^{k-1} \alpha_i A^i r_0) \\ &= r_0 - \sum_{i=0}^{k-1} \alpha_i A^{i+1} r_0 \\ &= r_0 - \sum_{i=0}^{k-2} \alpha_i A^{i+1} r_0 - \alpha_{k-1} \sum_{i=0}^{k-1} \gamma_i A^i r_0 \\ &= (1 - \alpha_{k-1}\gamma_0) r_0 - \sum_{i=1}^{k-1} (\alpha_{i-1} + \alpha_{k-1}\gamma_i) A^i r_0, \end{aligned} \tag{30}$$

where the coefficients γ_i are given by (28) and the coefficients α_i are used to minimize $\|r_k\|_2$. Since $\gamma_0 \neq 0$, we may set $\alpha_{k-1} = 1/\gamma_0$. Using $\alpha_{i-1} = -\alpha_{k-1}\gamma_i = -\gamma_i/\gamma_0,\ i = 1, \dots, k-1$ in (29), we obtain $r_k = 0$. Therefore, if Arnoldi's method breaks down, the last iterate x_k corresponding to r_k is already the exact solution, so that there is no need to continue the construction of the Krylov space.

Moreover, since there are at most n linear independent vectors, n denoting the number of unknowns, Arnoldi's process breaks down after at most n steps. Thus we have shown, that in the absence of rounding off errors, GMRES needs less than $n + 1$ iterations to reach the true solution x of $Ax = b$. Due to the minimization condition (26), $\|r_m\|_2$ is a monotone nonincreasing sequence of real numbers.

Following the above considerations, we may assume that (27) does not break down during the first m steps and produces an orthonormal basis $\{v_1, \dots, v_{m+1}\}$ of $K_m^A(r_0)$. Therefore we may write x_m as

$$x_m = x_0 + \sum_{j=0}^{m-1} \alpha_j A^j r_0 = x_0 + \sum_{j=1}^{m} \beta_j v_j$$

and

$$r_m = b - Ax_m = r_0 - \sum_{j=1}^{m} \beta_j A v_j = r_0 - \sum_{j=1}^{m} \beta_j w_j.$$

From (27) we conclude

$$w_j = \|\hat{v}_{j+1}\|_2 \, v_{j+1} + \sum_{i=1}^{j} (v_i, w_j) v_i. \tag{31}$$

Now we define the $(m+1) \times m$ matrix H through

$$\begin{array}{rcll} h_{ij} & = & (v_i, w_j), & i \leq j \leq m, \\ h_{j+1j} & = & \|\hat{v}_{j+1}\|_2, & 1 \leq j \leq m, \\ h_{ij} & = & 0, & 1 \leq j+1 < i \leq m+1 \end{array} \tag{32}$$

and V as the matrix consisting of the columns $v_1, \ldots, v_{m+1}$. Moreover, let $\beta = (\beta_1, \ldots, \beta_m)^T$ be the vector of the optimization parameters. Now (30) can be written as

$$w_j = h_{j+1j} v_{j+1} + \sum_{i=1}^{j} h_{ij} v_i = \sum_{i=1}^{m+1} h_{ij} v_i$$

and thus

$$\begin{aligned} r_m &= r_0 - \sum_{j=1}^{m} \beta_j \sum_{i=1}^{m+1} h_{ij} v_i \\ &= r_0 - \sum_{i=1}^{m+1} v_i \sum_{j=1}^{m} h_{ij} \beta_j \\ &= r_0 - V H \beta. \end{aligned}$$

Using $r_0 = \|r_0\|_2 \, v_1$, we get

$$r_0 = Ve, \qquad e = (\|r_0\|_2, 0, \ldots, 0)^T \in R^{m+1}.$$

Since the columns of V are orthonormal, we finally arrive at

$$\|r_m\|_2 = \|r_0 - VH\beta\|_2 = \|V(e - H\beta)\|_2 = \|e - H\beta\|_2. \tag{33}$$

Therefore the original minimization problem, going back to the (dense) $n \times m$ matrix VH, was substituted by a minimization problem involving the $(m+1) \times m$ upper Hessenberg matrix H. This may easily be solved using for example Givens rotations.

Summarizing the above considerations, GMRES is characterized as follows:

- Let x_0 be an initial guess for the solution of $Ax = b$.
- For $m = 1, 2, \ldots$, the iterates x_m are determined through the following steps:
  - Compute an orthonormal basis $v_1, \ldots, v_{m+1}$ of $K_m^A(r_0)$ using Arnoldis method

$$v_1 = \frac{r_0}{\|r_0\|_2},$$

for $j = 1, \ldots, m$:

$$\begin{aligned} w_j &= A v_j, \\ \hat{v}_{j+1} &= w_j - \sum_{i=1}^{j} (v_i, w_j) v_i, \\ v_{j+1} &= \frac{\hat{v}_{j+1}}{\|\hat{v}_{j+1}\|_2}. \end{aligned}$$

– Define x_m as

$$x_m = x_0 + \sum_{j=1}^{m} \beta_j v_j .$$

– Determine the coefficients β_j by minimizing

$$\|r_m\|_2 = \|e - H\beta\|_2, \qquad e = (\|r_0\|_2, 0, \ldots, 0)^T, \qquad \beta = (\beta_1, \ldots, \beta_m)^T,$$

using Givens rotations. The $(m+1) \times m$ upper Hessenberg matrix H is given by

$$\begin{array}{rcll} h_{ij} & = & (v_i, w_j), & i \leq j \leq m, \\ h_{j+1j} & = & \|\hat{v}_{j+1}\|_2, & 1 \leq j \leq m, \\ h_{ij} & = & 0, & 1 \leq j+1 < i \leq m+1 \end{array}$$

Remarks

- The sequence $\|r_m\|_2$ is monotone nonincreasing.
- After at most n steps, n being the number of unknowns, the true solution is obtained.
- The main drawback of the above algorithm is the large amount of memory which is necessary to store all the basis vectors $v_1, \ldots, v_m$ of the Krylov space $K^A_{m-1}(r_0)$. This is the reason why sometimes the following restarted version GMRES(k) is used:

  (i) we start with the same initial guess x_0

  (ii) we apply the GMRES algorithm described above but restrict the maximal dimension of the Krylov space to k, thus getting a new approximation x_k

  (iii) if the accuracy is not high enough, we go back to (i), but now use x_k as initial guess

  It is obvious, that GMRES(k) only requires a limited amount of memory depending on k. But the convergence results obtained for GMRES do not carry over.

6 Numerical results

In this chapter we apply the discretization method of section 4 to several test problems. We first present some results reflecting the difficulties which are dew to the nonuniqueness of our problem in the case of multiply connected domains. In the second part we couple the iterative methods from the last chapter with different preconditioning techniques and compare their rate of convergence.

6.1 Test problems

We consider the following three problems:

(A) D_- is a square massive plate of conducting material of the size 8 cm × 16 cm × 16 cm. The inducing current is flowing in a square concentric loop where each vertex has a length of 80 cm (Fig. 6).

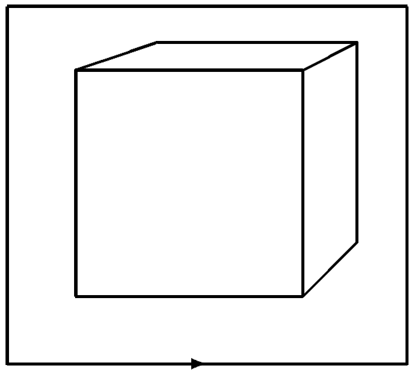

Fig. 6

(B) The situation in the second case is shown in Fig. 7. This configuration (measuring $4\,\mathrm{cm} \times 8\,\mathrm{cm} \times 4\,\mathrm{cm}$) is a little closer to reality, since it reveals a part of the complicated geometry which usually occurs in connection with electric machines.

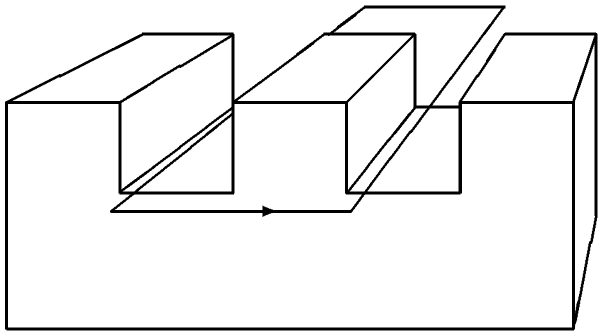

Fig. 7

(C) We consider (A) and concentrically cut out a block of size $4\,\mathrm{cm} \times 4\,\mathrm{cm} \times 8\,\mathrm{cm}$, so that D_- is no longer simply connected (Fig. 8).

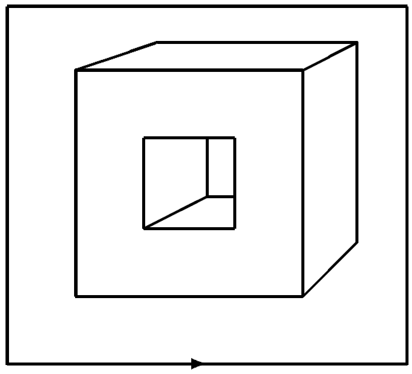

Fig. 8

For (A), (B) and (C) we assume the conducting material in D_- to be aluminium ($\mu_- = 4\pi * 10^{-7}\frac{Vs}{Am}$, $\sigma_- = 2.7 * 10^7 \frac{A}{Vm}$) and the inducing current to be $1\,A$.

The computed current densities in D_- are shown in Fig. 9–12. The crossections are taken in the plane of the inducing loop. The length of the arrows is proportional to the modulus of the current density.

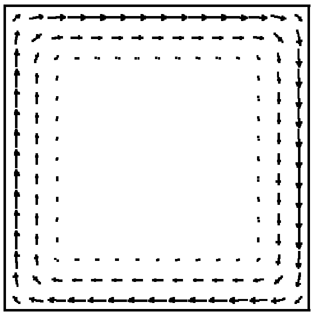

Fig. 9: Computed current densities for (A)

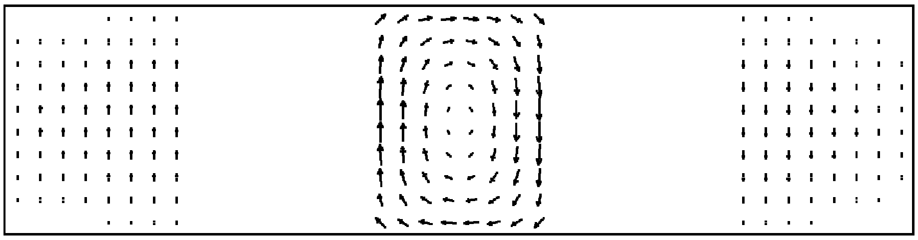

Fig. 10: Computed current densities for (B)

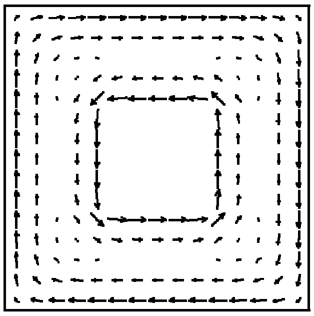

Fig. 11: Solution of the inhomogeneous problem (C)

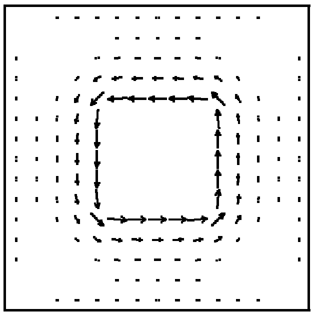

Fig. 12: Nontrivial solution of the homogeneous problem (C)

The densities corresponding to (A) and (B) show the expexted behaviour. This is not the case for (C). In Fig. 11 we see, that close to the hole in D_- the current flow changes its direction, which is physically wrong. This strange behaviour is due to the nonuniqueness of (3)–(5) mentioned in Chapter 3. In the case of (C), where the topological genus p of D_- is one, we get according to Theorem 2 exactly one nontrivial solution of the homogeneous problem (3)–(5), which is shown in Fig. 12.

Using some resently obtained theoretical results [6], it seems to be possible to motivate an algorithm how to construct out of this two solutions the physically correct one.

6.2 Comparison of the preconditioned iterative solvers

After having given a detailed description of BiCG, CGS, BiCGSTAB and GMRES we now want to check their performance by applying them to the linear systems $Ax = b$ we get for the test problems of the last section by using the finite volume discretization of Chapter 4. To accelerate the convergence, we combine the above algorithms with one of the following two preconditioners:

- **Incomplete LU Decomposition, ILU**
  We perform the usual Gaussian decomposition on A, using only those entries which are nonzero in the original matrix A. Thus we obtain lower (upper) triangular matrices L (U) being nonsingular and having the same sparsity pattern as the lower (upper) triangular part of A.

- **Matrix Splitting, SPLIT(ω)**
  We split A into $A = D + E + F$, where D is the diagonal of A and E (F) denotes the lower (upper) triangular part A. We define $L = D + \omega E$, $U = D + \omega F$ where ω is a complex parameter.

The matrices L and U obtained by the above procedures are easily inverted. Instead of the original equation $Ax = b$ we now consider one of the following three equivalent systems

$$\begin{aligned} (L^{-1}AU^{-1})\ (Ux)\ &= \quad L^{-1}b && (LR) \\ (U^{-1}L^{-1}A) \quad x \quad &= U^{-1}L^{-1}b && (L) \\ (AU^{-1}L^{-1})(LUx) &= \qquad b && (R) \end{aligned}$$

As initial guess we use for

$$\begin{aligned} (Ux)_0 \ &= \ U(U^{-1}L^{-1}b) \ = L^{-1}b && \text{for} \quad (LR), \\ x_0 \quad &= \quad U^{-1}L^{-1}b && \text{for} \quad (L), \\ (LUx)_0 &= LU(U^{-1}L^{-1}b) = \quad b && \text{for} \quad (R). \end{aligned}$$

The different algorithms are then applied to model problem (A). In the diagrams below the ratio $\|r_i\|_2/\|r_0\|_2$ is shown as a function of the number of matrix–vector products. Since these products are the characteristic time–consuming parts of both the conjugate–gradient like algorithms and GMRES, this scaling of the x–axes allows good comparison of the different routines, without refering to the CPU time.

If not explicitely stated, we used the ILU preconditioner and the modified system (LR).

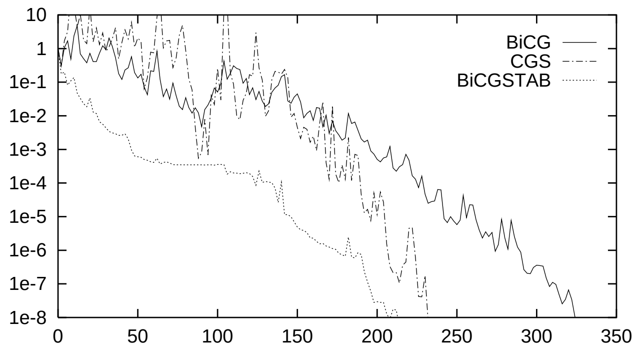

Fig. 13: Comparison of the different conjugate gradient methods

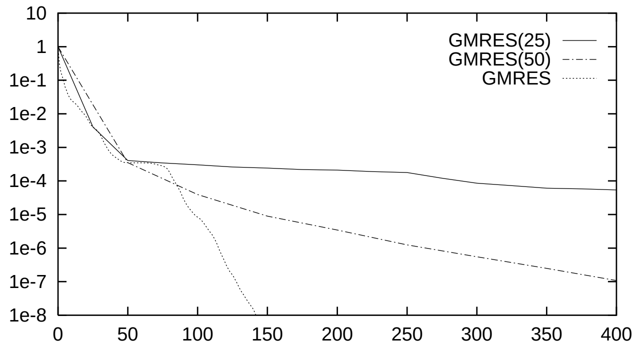

Fig. 14: Comparison of GMRES and GMRES(k)

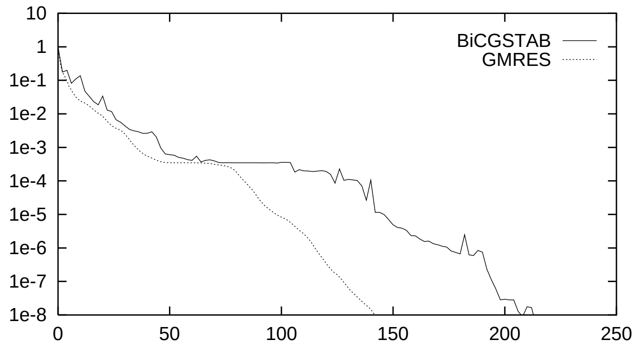

Fig. 15: Comparison of BiCGSTAB and GMRES

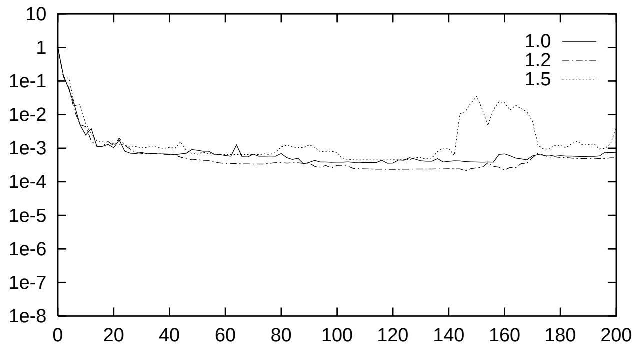

Fig. 16: Behaviour of BiCGSTAB-SPLIT for different parameters ω

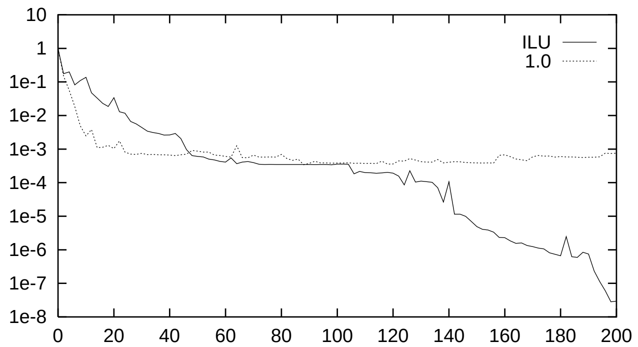

Fig. 17: Comparison of BiCGSTAB-ILU and BiCGSTAB-SPLIT

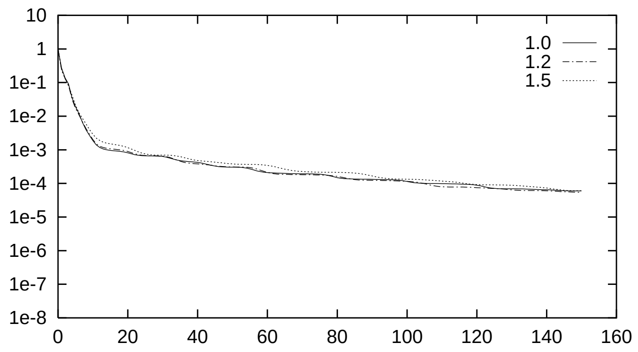

Fig. 18: Behaviour of GMRES-SPLIT for different parameters ω

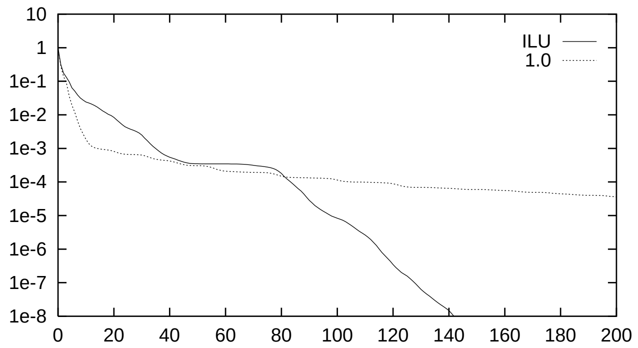

Fig. 19: Comparison of GMRES-ILU and GMRES-SPLIT

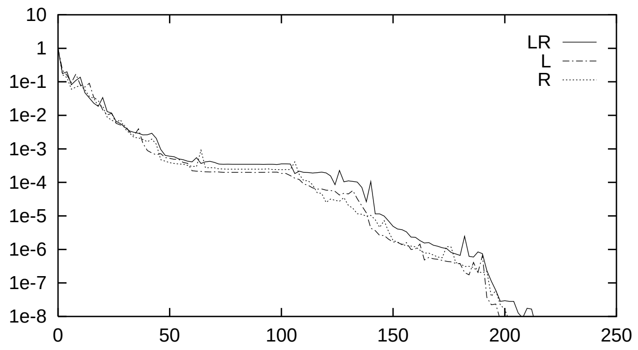

Fig. 20: BiCGSTAB-ILU applied to different modified systems

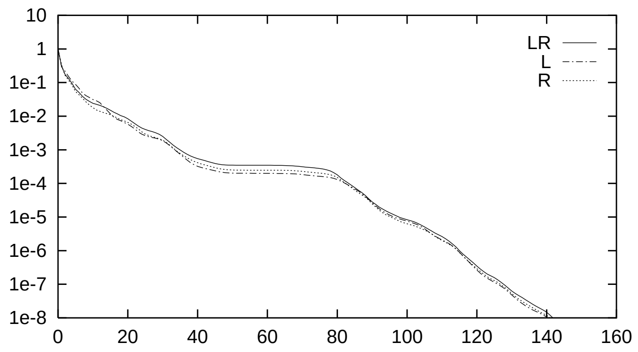

Fig. 21: GMRES-ILU applied to different modified systems

Fig. 13 shows the very strong oscillations of BiCG and CGS opposed to the smooth behaviour of BiCGSTAB. Moreover, the residuals of CGS are decaying much faster than those of BiCG. The speed up of BiGSTAB relative to CGS is not that large.

In the next figure, the original and restarted versions of GMRES are compared. The results show, that for our problem the restarted algorithms are of no use, since the residuals hardly decrease after the first restart.

Fig. 15 clearly displays the optimality of GMRES. What seems to be remarkable is the fact, that BiCGSTAB is very close to GMRES up to the first 70 matrix-vector products, where both slow down. But in contrast to GMRES, BiCGSTAB does not reduce the residual significantly for almost 45 steps.

The effect of the matrix splitting preconditioner is shown in Fig. 16–19 (using again the (LR) system). Instead of the rapid decrease of the residuals in the first few steps, the overall convergence behaviour is very poor compared with the ILU preconditioning strategy.

Finally we see (Fig. 20, Fig. 21) that the choice of the modified systems (LR), (L) resp., (R) does not significantly influence the speed of convergence of BiCGSTAB and GMRES (both coupled with ILU). This also holds for BiCG and CGS.

7 Conclusions and outlook

We presented an easy to handle and easy to implement discretization procedure for the solution of 3D eddy–current problems in simply connected domains. For multiply connected domains there occur some problems, since the model equations admit more than one solutions. With the above algorithm it is possible to approximate the whole set of solutions (based on this observation and some recently obtained theoretical results [6], it seems to be possible to modify this algorithm in order to produce the physically correct solutions independently of the topological genus of the domains under consideration).

The linear systems arising from the discretization are very large, sparse, complex, nonsymmetric and do not possess any special structure, so that standard iterative procedures in general do not converge.

In contrast to this, the Krylov subspace methods introduced in Chapter 5 together with a suitably chosen preconditioning method were successfully applied to our model problems.

GMRES showed the best convergence behaviour, having the disadvantage that the complete Krylov subspace has to be stored. For this reason it is preferable to use the slower conjugate gradient methods in quite a number of applications (especially when the systems become very large), thus keeping the amount of memory small and inreasing the overall efficiency. Taking into account the results of the last section, we highly recommend the use of BiCGSTAB in this case, because of the rather smooth decay of the residuals and the good rate of convergence.

A future field of interest would be a detailed investigation of the different preconditioning strategies. This seems to be a very challenging problem, since simple modifications of the underlying linear system (like a rescaling or a reordering of the equations or unknowns) may have a very strong influence on preconditioners like ILU. For general systems, these effects are not at all understood.

References

[1] Faber V., Manteuffel T., *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21, 1984, pp. 352-362

[2] Jacobs D. A. H., *The exploitation of Sparsity by Iterative Methods* pp. 191-221 in Duff I. S., *Sparse Matrix and their Uses*, Academic Press 1981

[3] Martensen E., *Potentialtheorie*, Teubner 1968

[4] Müller C., *Grundprobleme der mathematischen Theorie elektromagnetischer Schwingungen*, Springer 1957

[5] Reissel M., *Theorie und Numerik eines Wirbelstromproblems*, PhD Thesis, Kaiserslautern 1992

[6] Quell P., *Über das Grenzverhalten eines Wirbelstromproblems für verschwindende Verschiebungsströme*, Thesis, Kaiserslautern 1995

[7] Saad Y., Schultz M. H., *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., Vol. 7, No. 3, 1986, pp. 856-869

[8] Sonneveld P., *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp. 10, No. 1, 1989, pp. 36-52

[9] Van der Vorst H. A., *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems* SIAM J. Sci. Stat. Comp. 13, No. 2, 1992, pp. 631-644

[10] Wilde P., *Über Transmissionsprobleme bei der vektoriellen Helmholtzgleichung*, PhD Thesis, Göttingen 1985