Robust Parallel Newton–Multilevel Methods

Bodo Heise*
Institute for Mathematics, Johannes Kepler University Linz,
A - 4040 Linz, Austria.
e-mail: heise@miraculix.numa.uni-linz.ac.at

Michael Jung
Faculty for Mathematics, Technical University Chemnitz,
D - 09107 Chemnitz, Germany.
e-mail: michael.jung@mathematik.tu-chemnitz.de

Abstract

The present paper is devoted to the numerical solution of nonlinear boundary value problems arising in the magnetic field computation and in solid mechanics. These problems are discretized by using finite elements.

The nonlinearity is handled by a nested Newton solver, and the linear systems of algebraic equations within each Newton step are solved by means of various iterative solvers, namely multigrid methods and conjugate gradient methods with DD preconditioners as well as BPX preconditioners. All solvers are based on a non-overlapping domain decomposition data structure such that they are well-suited for implementations on parallel machines with MIMD architecture.

We compare by numerical examples the performance of the different iterative solvers which are applied within each Newton step.

KEY WORDS nonlinear partial differential equations, parallel computing, multigrid methods, domain decomposition, finite element methods, magnetic field calculations

AMS SUBJECT CLASSIFICATION 65N55, 65N22, 65N30, 78A30

1 Introduction

Recently, Multiple Instruction Multiple Data (MIMD) parallel computers with message-passing principle have found more and more interest. These parallel machines provide sufficient CPU power and sufficiently large storage capacity as it is necessary for the numerical simulation of complex processes. Therefore, efficient parallel solvers for large systems of algebraic equations resulting from, e.g., the finite element (FE) or boundary element (BE) discretization of partial differential equations, have been developed.

*Acknowledgment: This work is supported by the Austrian Science Fund – 'Fonds zur Förderung der wissenschaftlichen Forschung' – under project P 11215-TEC.

There are different approaches for the construction of efficient parallel solvers. The first one is the application of Domain Decomposition (DD) preconditioners within the preconditioned conjugate gradient (PCG) method. For example, the non-overlapping DD (NODD) methods, the main ideas of which are presented, e.g., in [3, 7, 8, 14, 13, 18, 28], have been applied successfully. Here, one has to define preconditioning matrices for the stiffness matrices of the corresponding problems in the subdomains, a preconditioner for the Schur complement, and a basis transformation (for more details see, e.g., [14, 13, 11, 8, 27, 28]). The appropriate definition of these preconditioners and of the basis transformation leads to (almost) optimal PCG methods, i.e. the number of iterations for getting an approximate solution with a relative accuracy ε is of the order $\mathcal{O}(\ln h_q^{-1} \ln \varepsilon^{-1})$ or $\mathcal{O}(\ln \varepsilon^{-1})$ (h_q denotes the dicretization parameter).

A second possibility for getting parallel solvers is the implementation of well-known global optimal iterative solvers, as e.g. multigrid methods or PCG methods with BPX preconditioners, on parallel machines. Using a NODD data structure, these solvers need, just as the DD-PCG method, a communication cost per iteration step which is one order lower than the FE problem itself.

In particular, a global multigrid process has been implemented in such a way that the multigrid interpolation and restriction operators can be performed without any communication [24]. Thus, communication is needed in the smoother and the coarse–grid solver only. The ideas presented in [24] enable us to implement the Gauss–Seidel smoother with the same communication effort as it is required by the Jacobi smoother. Further, a Schur complement solver with preconditioning, originating from the local DD methods, is employed as coarse–grid solver.

A comparison of the CG method with DD preconditioner (DD-PCG) and the global multigrid (GMG) method as linear problem solvers in a nested Newton framework (i.e. an inexact Newton method [5, 6], see [17, 19, 20, 21]) can be found in [22]. Therein, we present parallel efficiency results, too. We discovered that the Newton-GMG method can be faster than the Newton-DD-PCG method for model problems and a practical problem with low nonlinearity, whereas Newton-DD-PCG shows the better scalability. We observed a scaled efficiency of 0.9 for the Newton-DD-PCG method and of 0.7 for the Newton-GMG method.

In addition to the solvers mentioned above, we implemented a parallel CG algorithm with global BPX preconditioner (GBPX-PCG) [4] which uses the NODD data structure. Further, the GMG method can serve as a preconditioner (GMG-PCG) in the parallel CG algorithm (cf. [25]), too.

The aim of the present paper is to compare the performance and robustness of the different parallel algorithms for difficult nonlinear practical problems. This includes tests for the magnetic field computation in electric machines with complicated interior geometry and strong nonlinearity, i.e., high saturation of the iron parts. The latter causes a local anisotropy of the linear Jacobi operator. Further, we present test calculations for another nonlinear test problem originating from a shape design problem (see [9, 26]) that had been already examined in [17, 21].

The rest of the paper is organized as follows. In Section 2, we formulate the electromagnetic field problem as a nonlinear boundary value problem. We apply the nested Newton-DD method for linearization. We discuss some properties of the discrete Jacobi operators involved in the Newton method. In Section 3, we describe the different methods for solving the arising linear problems. Section 4 is devoted to the numerical results. We present performance results of the magnetostatic field simulation for a direct current motor (DC motor) and an induction machine as well as for the nonlinear shape design problem. Finally, we add some concluding remarks in Section 5.

2 The nonlinear problem and its linearization

A two–dimensional stationary magnetic field problem involving the saturation effects of ferromagnetic materials can be written as a nonlinear boundary value problem in its variational formulation as follows:

Find $u \in V = H_0^1(\Omega)$ *such that*

$$a(u,v) = \langle f, v\rangle \quad \forall v \in V, \tag{1}$$

where

$$a(u,v) = \int_\Omega \nu(x,|\nabla u|)\ \nabla^\top u\ \nabla v\ dx\ ,$$

and

$$\langle f, v\rangle = \int_\Omega (\mathrm{S}v - \mathrm{H}_{0\mathrm{y}}\frac{\partial v}{\partial \mathrm{x}} + \mathrm{H}_{0\mathrm{x}}\frac{\partial v}{\partial \mathrm{y}})\ dx\ .$$

Here, $\Omega \subset \mathbf{R}^2$ denotes a bounded domain. The physical model has been developed from Maxwell's equations, see [18] for details. We assume that Ω representing the cross–section of some electromagnetic device lies in the x–y–plane of $\mathbf{R}^3$. Then, the solution u is the z–component of some vector potential $\vec{\mathrm{A}}$. The z–component of the current density is represented by S, and the vector $\vec{\mathrm{H}}_0 = (\mathrm{H}_{0\mathrm{x}}, \mathrm{H}_{0\mathrm{y}}, 0)^\top$ describes the magnetization of permanent magnets. The nonlinearity of the problem is represented by the dependence of ν on the absolute value of the magnetic induction, $B = |\mathrm{rot}\vec{\mathrm{A}}| = |\nabla u|$.

We assume that $\bar{\Omega}$ consists of subdomains

$$\bar{\Omega} = \bigcup_{j=1}^{N_M} \bar{\hat{\Omega}}_j, \quad \text{with} \quad \hat{\Omega}_i \cap \hat{\Omega}_k = \emptyset \quad \forall i \neq k.$$

The $\hat{\Omega}_j$'s represent materials with different magnetic properties (iron, copper, air, permanentmagnetic materials) in the cross–section of an electromagnetic device. We assume that the function ν depends on the position $x \in \Omega$, but ν becomes independent of x inside each subdomain $\hat{\Omega}_j$, i.e.

$$\nu(x,B) = \nu^{(j)}(B) \quad \text{if} \quad x \in \hat{\Omega}_j, \quad j = 1,\ldots,N_M. \tag{2}$$

The function $\nu^{(j)}(B)$ is a constant, $\nu^{(j)}(B) \equiv \nu_1^{(j)}$, if the material in $\hat{\Omega}_j$ is not ferromagnetic (e.g., copper, air, vacuum). Assuming certain monotonicity and boundedness conditions on the functions $\nu^{(j)}(.)$ we can prove the existence and uniqueness of the solution of the variational problem (1) [17, 18]. Standard FE discretization with linear triangular elements has been discussed in [18]. Therein, the algorithm for monotonicity preserving spline interpolation of a pointwise given material function $\nu^{(j)}(.)$ is described, error estimates are given, too.

The parallel nested Newton (PNN) method for solving the discrete problems combining a Newton-like method with the nested iteration and a suitable parallel solver, is described in detail, e.g., in [21, 19, 20, 22]. Therefore, we recall the main definitions only.

The substructuring into non-overlapping subdomains $\Omega_i, i \in \mathcal{I}^\star := \{1,\ldots,p\}$, which are assigned to p processors of the MIMD computer can be defined as follows

$$\bar{\Omega} = \bigcup_{i\in\mathcal{I}^\star} \bar{\Omega}_i, \quad \text{where} \quad \bar{\hat{\Omega}}_j = \bigcup_{i\in\mathcal{I}_j} \bar{\Omega}_i \quad \forall j = 1,\ldots,N_M \tag{3}$$

with index sets fulfilling

$$\mathcal{I}_j \subset \mathcal{I}^\star := \{1,\ldots,p\}\ , \quad \bigcup_{j=1}^{N_M} \mathcal{I}_j = \mathcal{I}^\star, \quad \mathcal{I}_j \cap \mathcal{I}_k = \emptyset \ \ \forall j \neq k,$$

i.e., the subdomains $\bar{\Omega}_j$ determined by the materials may be decomposed further (cf. [13, 14]). We further assume that in each subdomain $\bar{\Omega}_i$ there is a multilevel sequence of linear finite element discretizations such that this discretization process results in conform triangulations $\mathcal{T}_q, q = 1, \ldots, l$, of Ω creating a nested sequence

$$V_1 \subset V_2 \subset \cdots \subset V_q \subset \cdots \subset V_l \subset V = H_0^1(\Omega) \tag{4}$$

of spaces of linear finite elements. The FE isomorphism is denoted by $\Phi_q : \mathbf{R}^{N_q} \longrightarrow V_q$.

We obtain a sequence of variational problems for $q = 1, \ldots, l$:
Find $u_q \in V_q \subset V$ such that

$$a_q(u_q, v_q) = \langle f_q, v_q \rangle \quad \forall v_q \in V_q, \tag{5}$$

and a sequence of equivalent nonlinear finite element equations

$$K_q \mathbf{u}_q = \mathbf{f}_q, \quad q = 1, \ldots, l, \tag{6}$$

with nonlinear operators

$$K_q : \mathbf{R}^{N_q} \longrightarrow \mathbf{R}^{N_q},$$

solution vectors $\mathbf{u}_q \in \mathbf{R}^{N_q}$ and vectors $\mathbf{f}_q \in \mathbf{R}^{N_q}$.

In the Newton method, we have to solve linear systems of the form

$$K_q'[\mathbf{v}_q]\mathbf{w}_q = \mathbf{d}_q \tag{7}$$

with the Fréchet derivative of K_q at a vector $\mathbf{v}_q$,

$$K_q'[\mathbf{v}_q] : \mathbf{R}^{N_q} \to \mathbf{R}^{N_q} \quad q = 1, \ldots, l,$$

which can be represented by the Jacobi matrix.

The analysis shows that this operator can be strongly anisotropic. We recall from [17] that the Jacobian can be rewritten as

$$(K_q'[\mathbf{v}_q]\mathbf{w}_q, \mathbf{z}_q)_q = \sum_{\mathcal{K} \in \mathcal{T}_q} \operatorname{meas}(\mathcal{K})\ \nabla^\top \Phi_q \mathbf{w}_q\ \ Q_{j(\mathcal{K})}(\nabla \Phi_q \mathbf{v}_q)\ \ \nabla \Phi_q \mathbf{z}_q \quad \forall \mathbf{w}_q, \mathbf{z}_q \in \mathbf{R}^{N_q}, \tag{8}$$

with

$$Q_j(\mathbf{t}) = \nu^{(j)}(|\mathbf{t}|) I_2 + \frac{\nu^{(j)\prime}(|\mathbf{t}|)}{|\mathbf{t}|} \mathbf{t}\, \mathbf{t}^\top, \tag{9}$$

where $\mathbf{t} = (t_1, t_2)^\top \in \mathbf{R}^2$, the matrix $I_2 \in \mathbf{R}^{2\times 2}$ denotes the identity matrix, and $j(\mathcal{K})$ is the corresponding material index, i.e., $j(\mathcal{K})$ is defined by $\mathcal{K} \subset \bar{\hat{\Omega}}_{j(\mathcal{K})} \quad \forall \mathcal{K} \in \mathcal{T}_q$, cf. (2).

The matrix $Q_j \in \mathbf{R}^{2\times 2}$ can be rewritten as

$$Q_j = \begin{pmatrix} \nu^{(j)}(|\mathbf{t}|) + \frac{\nu^{(j)\prime}(|\mathbf{t}|)}{|\mathbf{t}|} t_1^2 & \frac{\nu^{(j)\prime}(|\mathbf{t}|)}{|\mathbf{t}|} t_1 t_2 \\ \frac{\nu^{(j)\prime}(|\mathbf{t}|)}{|\mathbf{t}|} t_1 t_2 & \nu^{(j)}(|\mathbf{t}|) + \frac{\nu^{(j)\prime}(|\mathbf{t}|)}{|\mathbf{t}|} t_2^2 \end{pmatrix}. \tag{10}$$

Obviously, we have $Q_j = \operatorname{diag}(\nu_1^{(j)}, \nu_1^{(j)})$ for not ferromagnetic materials. For ferromagnetic materials, there are values of the magnetic induction $B = |\mathbf{t}|$ for which $\nu^{(j)\prime}$ is large, in particular for intermediate values between the linear part

$$\nu^{(j)}(B) \equiv \nu_1^{(j)} \quad \forall B \le B_1$$

and the asymptotic behaviour

$$\nu^{(j)}(B) \longrightarrow \nu_\infty = 1/\mu_0 \quad \text{for} \quad B \to \infty,$$

since $\nu^{(j)}(B)$ is a monotone increasing function [18]. Then the relation

$$|\mathbf{t}|\nu^{(j)\prime}(|\mathbf{t}|) \gg \nu^{(j)}(|\mathbf{t}|) \tag{11}$$

holds, and the Jacobi operator becomes the discretization of an anisotropic operator.

For the example $\mathbf{t} = (t_1, 0)^\top$, we get from (10) that $Q_j = \mathrm{diag}(q_{11}, q_{22})$ with $q_{22} = \nu^{(j)}(|\mathbf{t}|)$ and $q_{11} = q_{22} + |\mathbf{t}|\nu^{(j)\prime}(|\mathbf{t}|) \gg q_{22}$ holds, and the anisotropy is obvious. The anisotropy is, of course, not restricted to that direction of $\mathbf{t} = \nabla\Phi_q \mathbf{v}_q$.

3 Parallel solvers for linear problems

The standard NODD data structure and the parallelization strategy are presented, e.g., in [13]. The linear equations (7) can be rewritten in the standard block form [13]

$$J \begin{pmatrix} \mathbf{w}_C \\ \mathbf{w}_I \end{pmatrix} = \begin{pmatrix} J_C & J_{CI} \\ J_{IC} & J_I \end{pmatrix} \begin{pmatrix} \mathbf{w}_C \\ \mathbf{w}_I \end{pmatrix} = \begin{pmatrix} \mathbf{d}_C \\ \mathbf{d}_I \end{pmatrix} \tag{12}$$

where indices "I" and "C" correspond to the nodes belonging to the interior of subdomains Ω_i and to the coupling boundaries, respectively, and $\dim(\mathbf{w}_I) = N_{I,q}$, $\dim(\mathbf{w}_C) = N_{C,q}$. Note that $J_I = \mathrm{diag}\,(J_{I,i})_{i=1,2,\ldots,p}$ is a block-diagonal matrix.

DD preconditioned conjugate gradient method:
The parallel CG algorithm with DD preconditioning for solving the systems (12) can be implemented in a standard way, see [13, 23]. It runs completely in parallel with the exception of the two scalar products, and the preconditioning. The DD preconditioner for J, i.e. the matrix C with

$$C = \begin{pmatrix} I_C & J_{CI}B_I^{-T} \\ O & I_I \end{pmatrix} \begin{pmatrix} C_C & O \\ O & C_I \end{pmatrix} \begin{pmatrix} I_C & O \\ B_I^{-1}J_{IC} & I_I \end{pmatrix} \tag{13}$$

contains three components, i.e., the preconditioners C_C and $C_I = \mathrm{diag}\,(C_{I,i})_{i=1,2,\ldots,p}$, and the regular matrix $B_I = \mathrm{diag}\,(B_{I,i})_{i=1,2,\ldots,p}$ defining the basis transformation, which can be adapted to the matrix J in a suitable way [13].

Here, we choose a multigrid V-cycle with one pre- and one postsmoothing step of Gauss-Seidel type in the symmetric Multiplicative Schwarz Method [13] for C_I, and B_I is implicitly defined by hierarchical extension (formally $E_{IC} = -B_I^{-1}J_{IC}$) [15]. We apply a Schur complement preconditioner C_C following Bramble/Pasciak/Schatz [3], which uses the idea of Dryja [7] on the coupling boundaries and a global crosspoint system (BPS-D), or a Bramble/Pasciak/Xu [29, 4] type Schur complement preconditioner together with a global crosspoint system (S-BPX).

Spectral equivalence between J and C has been proved in [13]. Together with the results of [3, 4, 15, 17, 29] we can prove that the numerical effort spent for one Newton step on grid q is at most of order $\mathcal{O}(N_q \ln\ln h_q^{-1} \ln\varepsilon_{\mathrm{lin}}^{-1})$ in the (S-BPX) case, i.e. almost optimal. In the (BPS-D) case we have to add a factor $\ln h_q^{-1}$. Here h_q denotes the discretization parameter, such that $N_q = \mathcal{O}(h_q^{-2})$. We refer to [23, 21] for details.

Global multigrid method:
In the NODD data structure, the interpolation and restriction procedures do not need any communication. Further, we developed parallel Gauss-Seidel type smoothers and Jacobi

smoothers. Both types of smoothers require the same communication effort, i.e., one so-called vector type conversion per iteration step. This procedure requires a data exchange of the order $\mathcal{O}(N_q^{0.5})$ between the processors [1, 24]. The Gauss-Seidel smoother executes the smoothing first for the coupling nodes ("C"), and then for the inner nodes ("I"), i.e., "forward", or in the reverse order, "backward" (see [24]).

We utilize parallelized preconditioned conjugate gradient methods with a (BPS-D) preconditioner applied to the corresponding Schur complement system as coarse grid solvers. Here, communication is required in the two scalar products and in the preconditioner, whereas all other operations are completely parallel. The parallel multigrid algorithm is described in detail in [24, 22].

Obviously, the GMG method can serve as a preconditioner in the CG algorithm, too [25].

Parallel global BPX method:
We realize the parallel CG method with BPX preconditioner [4] in a similar manner. In our implementation, the data of different levels is exchanged together, i.e., the number of data exchange steps (the startup time) is independent of the number of levels. Additionally, we need a coarse-grid solver which coincides completely with that of the GMG method.

4 Numerical results

4.1 Implementation

The complete algorithms are implemented in the parallel code FEM$\otimes$BEM [11, 24]. Thus, all parts of the parallel nested Newton algorithm, such as the grid generation, the matrix generation and representation, and the defect computation, are identical for all solving methods. We tested the algorithms on the parallel system GC-Power Plus (with maximal 64 processors Power PC 601) with the operating system Parix. In the following we describe our choice of components and parameters in the solvers for linear problems.

Initially, the linear problems arising in the Newton method are solved with a relative accuracy of $\varepsilon_{\text{lin}} = 0.01$. This parameter ε_{lin} can be adapted to the quadratic convergence speed of the Newton method in later Newton iterations [17].

In the (GMG) solver, we used a V-cycle with 2 pre- and 2 postsmoothing steps of the parallelized Gauss-Seidel type, and we restrict the maximal number of multigrid iterations initially to 2. This number is doubled if the multigrid convergence rate affects the Newton rate essentially, see [17]. With respect to the coarse grid solver, a preconditioned Schur complement CG solver turned out to be sufficient where best results have been obtained with a relative accuracy of 0.1 in the multilevel case ($q \geq 2$).

In the GMG-PCG method, we apply one V-cycle as a preconditioner. In the GBPX-PCG method, we scale the restricted right-hand side by the diagonal elements of the Jacobi matrix.

To be sure, we demand an error reduction by the relative accuracy $\varepsilon = 10^{-6}$ on the finest grid, therefore, we apply 4 or 5 Newton iterations on the finest grid. From the nested iteration we can expect that for obtaining an approximate solution which differs from the exact solution in the order of the discretization error, two Newton iterations are sufficient, i.e., we can nearly halve the processing time presented in the tables.

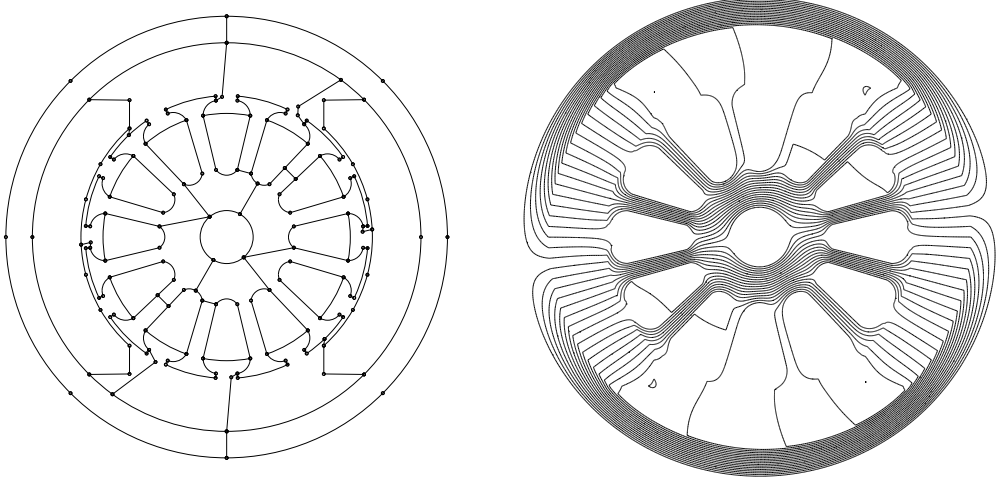

Figure 1: Domain decomposition and level lines for the DC machine

4.2 Direct current motor

A technical direct current (DC) motor which is excited by permanent magnets serves as an example of practical interest. The machine has a diameter of 50 mm, it is described in detail in [17]. The motor contains two different ferromagnetic materials; the rotor consists of dynamo sheet, the case is made from rolled steel. In Figure 1, we present the decomposition of the domain Ω into 32 subdomains and the level lines of the magnetic vector potential. This decomposition is generated automatically by means of the preprocessing tool ADDPRE (Adaptive DD Preprozessor, see [10, 12]). The performance of the different algorithms is compared in Table 1.

Table 1: Performance for the DC machine

Solver	DD-PCG (S-BPX)	GMG	GMG -PCG	GBPX -PCG
Newton iterations 1st grid	6	6	6	6
Newton iterations 2nd grid	2	2	2	2
CG/MG iterations 2nd grid	19,21	2,2	2,3	12,11
Newton iterations 3rd grid	2	2	2	2
CG/MG iterations 3rd grid	16,17	2,2	2,3	12,12
Newton iterations 4th grid	2	2	2	2
CG/MG iterations 4th grid	16,17	2,2	2,2	11,12
Newton iterations 5th grid	4	4	4	4
CG/MG iterations 5th grid	16,22,16,22	2,2,2,2	2,3,2,3	11,17,16,20
Time (system generation)	28.0	27.2	27.2	27.2
Time (solver)	137.9	36.4	49.8	87.4
Total time	165.9	63.6	77.0	114.6

Time in seconds, GC-Power Plus, 32 processors (subdomains); 5 grids, 409 948 unknowns, relative accuracy $\varepsilon = 10^{-6}$

We find that the GMG method has the best performance. This is due to the excellent convergence properties of the multigrid iteration that keep the number of iterations small, whereas DD-PCG and GBPX-PCG need more iterations to achieve the relative accuracy $\varepsilon_{\text{lin}} = 0.01$.

Although the ratio of anisotropy, i.e. the quotient of the eigenvalues of the matrix Q_j (cf. (10)), is about 10 in some finite elements, the anisotropy does not affect the convergence of the GMG method in this example.

4.3 Induction machine

The second example of technical interest, an induction machine (asynchronous motor), is a more challenging one due to its very complicated interior geometry and the stronger influence of saturation (i.e., a stronger nonlinearity). Indeed, we have an air gap of 0.2 mm where the coefficient has a jump by a factor of more than 1000, whereas the machine has a diameter of 45 mm. The electro-magnetic field is simulated for a state with currents in the stator, but no rotor currents. We present the automatically generated decomposition of the cross-section of the machine into 64 subdomains [10, 12] and level lines in Figure 2, performance results in Table 2.

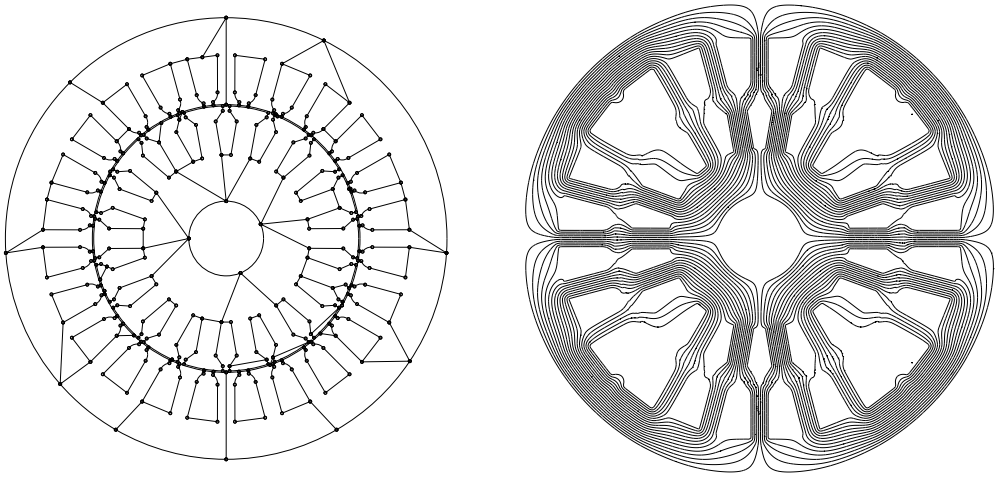

Figure 2: Domain decomposition and level lines for the induction machine

Note that in Table 2 for the (GMG) with 2 Gauss-Seidel "backward" pre-smoothing and 2 "backward" post-smoothing steps, (2b,2b), we achieved the best performance of all solvers. But, for the same number of "forward" Gauss-Seidel steps, (2f,2f), the multigrid solver on the levels $q = 3, 4, 5$ even did not converge.

We discovered that in large areas of the machine the ratio of anisotropy, i.e. the quotient of the eigenvalues of the matrix Q_j (cf. (10)), is between 17 and 18. Therefore, the Gauss-Seidel iteration may be a poor smoother [16]. We conclude from (10) that the dominance direction of the anisotropy is determined by the direction of $\mathbf{t}$. Thus, it depends on the direction of ∇u_q^j for the current approximate solution u_q^j. The direction of ∇u_q^j is perpendicular to the direction of the magnetic induction $\vec{\mathrm{B}}_q^j$. It differs for different parts of a machine, and it cannot be predicted a priori for other than model problems.

For Table 3, the current density is reduced by a factor 0.1. Consequently, the magnetic induction is lower. Various combinations of Gauss-Seidel smoothers perform well. Here, we have 4 Newton iterations on level 1, and 2 Newton iterations with 2 multigrid cycles on the levels 2, 3, and 4 for all 4 columns. The abbreviation "fb" denotes a "forward" followed by a "backward" step.

The application of the MG algorithm for defining a preconditioner in the PCG method gives a more robust solver. We observed that (GMG-PCG) *does* converge for several variants of Gauss-Seidel smoothers, see Table 4.

Table 2: Performance for the induction machine

Solver	DD-PCG (BPS-D)	GMG (2b,2b)	GMG-PCG (2b,2b)	GBPX -PCG
Newton iterations 1st grid	11	11	11	11
Newton iterations 2nd grid	2	2	2	2
CG/MG iterations 2nd grid	43,55	2,2	2,2	13,11
Newton iterations 3rd grid	2	2	2	2
CG/MG iterations 3rd grid	42,30	2,2	3,3	15,18
Newton iterations 4th grid	2	2	2	2
CG/MG iterations 4th grid	13,81	2,2	3,3	16,21
Newton iterations 5th grid	5	5	5	5
CG/MG iterations 5th grid	36,61,60,74,61	2,2,4,8,8	3,4,3,4,5	17,29,35,39,34
Time (system generation)	49.5	48.2	48.3	50.2
Time (solver)	550.2	295.9	347.2	848.8
Total time	599.7	344.1	395.5	899.0

Time in seconds, GC-Power Plus, 64 processors (subdomains); 5 grids, 549 091 unknowns, relative accuracy $\varepsilon = 10^{-6}$, current density J (cf. Table 3)

Table 3: Performance for the induction machine with reduced current density (GMG)

Smoother	(2f,2f)	(2b,2b)	(fb,fb)	(2f,2b)
Newton iterations 5th grid	4	4	5	5
MG iterations 5th grid	2,2,4,8	2,2,4,8	2,2,4,8,8	2,2,4,8,8

GC-Power Plus, 64 processors (subdomains); 5 grids, 549 091 unknowns, relative accuracy $\varepsilon = 10^{-6}$, current density $0.1 * J$

Table 4: Performance for the induction machine (GMG-PCG)

Smoother	(2f,2b)	(fb,fb)	(2b,2b)	(2f,2f)
Newton iterations 1st grid	11	11	11	11
Newton iterations 2nd grid	2	2	2	2
CG iterations 2nd grid	3,3	3,3	2,2	2,2
Newton iterations 3rd grid	2	2	2	2
CG iterations 3rd grid	3,3	3,3	3,3	3,3
Newton iterations 4th grid	2	2	2	2
CG iterations 4th grid	3,3	3,4	3,3	3,3
Newton iterations 5th grid	5	5	5	5
CG iterations 5th grid	3,4,4,5,3	3,4,5,5,5	3,4,3,4,5	3,4,3,4,4

GC-Power Plus, 64 processors (subdomains); 5 grids, 549 091 unknowns, relative accuracy $\varepsilon = 10^{-6}$, current density J (cf. Table 3)

4.4 Another nonlinear test problem

This example demonstrates the application of the algorithms to a quite different nonlinear problem.

Consider the problem in $\Omega = (0,1) \times (0,1) \subset \mathbf{R}^2$:
Find $u \in V = H_0^1(\Omega)$ *such that*

$$\int_\Omega \psi(|\nabla u|)\ \nabla^\top u\ \nabla v\ dx = \int_\Omega v\ dx \quad \forall v \in V \tag{14}$$

where

$$\psi(z) = \begin{cases} 2 & \text{if } z^2 \le 0.008 \\ 1 & \text{if } z^2 \ge 0.032 \\ \sqrt{0.032}/z & \text{else.} \end{cases} \tag{15}$$

The problem arises in the theory of elasticity (see [26] and [9]). We want to construct an infinitely long elastic bar of given cross–section Ω with maximal torsional rigidity from two different linearly elastic materials of given shear moduli. The proportions of these materials are prescribed. The problem has nearly the same form as (1) but it is not strongly monotone. Further, ψ' is not continuous.

The problem served as a test example for the Full Multigrid Newton Technique using the program FEMGP on a personal computer, see [17], and for the nested Newton-DD-PCG method in [21]. Our first aim is to demonstrate the performance of the algorithms. Further, we want to check whether in the "homogenized region", i.e., the region where $0.008 \le |\nabla u_l|^2 \le 0.032$ (here the two materials have to be "mixed"), the level lines are circular sectors with identical radius, as it has been predicted theoretically for the solution u in [26].

We present the performance in Table 5, the "homogenized region" on the left-hand side of Figure 3 and the level lines on the right-hand side.

Table 5: Performance for the shape design problem

Solver	DD-PCG (BPS-D)	DD-PCG (S-BPX)	GMG-PCG (fb,fb)	GBPX -PCG
Newton iterations 1st grid	44	44	38	38
CG iterations 1st grid	5 ··· 23	5 ··· 23	5 ··· 23	5 ··· 23
Newton iterations 2nd grid	2	2	2	2
CG iterations 2nd grid	11,16	18,11	5,5	15,16
Newton iterations 3rd grid	2	2	2	2
CG iterations 3rd grid	14,11	26,23	8,6	36,20
Newton iterations 4th grid	2	2	2	2
CG iterations 4th grid	14,23	28,17	9,9	41,29
Newton iterations 5th grid	7	6	5	5
CG iterations 5th grid	18,29,57, 70,24,40,74	42,53,63, 61,93,51	13,9, 26,23,27	55,29, 125,91,97
Time (system generation)	32.5	27.0	24.9	24.9
Time (solver)	391.3	483.1	219.9	348.3
Total time	423.8	510.1	244.8	373.2

Time in seconds, GC-Power Plus, 64 processors (subdomains); 5 grids, 1 530 657 unknowns, relative accuracy $\varepsilon = 10^{-6}$

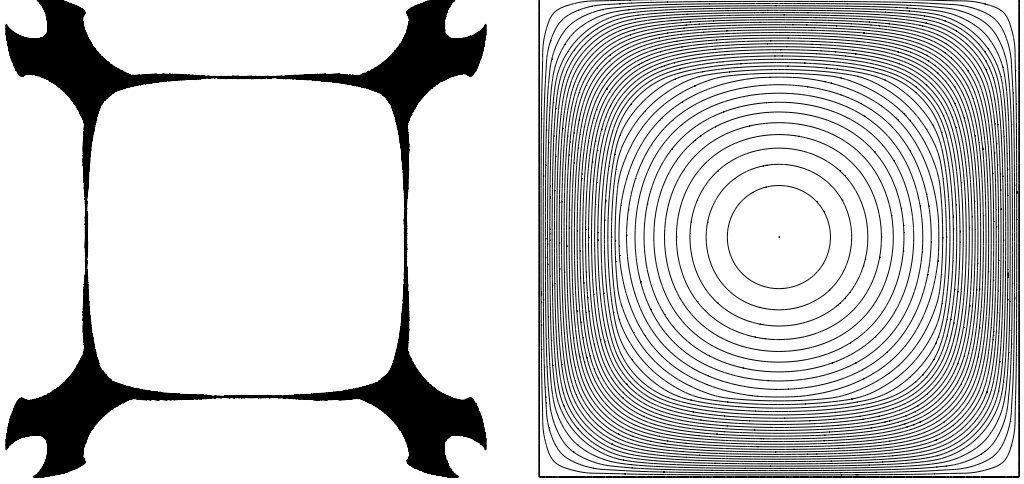

Figure 3: Homogenized region and level lines for the shape design problem

We note that for this example the Newton-GMG method does not converge, nor for Gauss-Seidel smoothers, nor for Jacobi smoothers with up to 8 smoothing steps. On the other hand, the combination of multigrid with a CG method leads to a quite robust solver. We observe that the "homogenized region" has the typical shape, and the level lines in that region can be recognized as circular sectors.

5 Conclusions

We have tested several parallel iterative solvers for the linearized problems arising in a nested Newton method for real-life nonlinear problems. We have found that the DD-PCG method is able to solve the problems, but the convergence can be slow.

The GMG method can converge faster, but its components have to be adapted well to the actual problem. Indeed, for anisotropic linear problems, we have to apply other smoothers than Gauss-Seidel ones, e.g. ILU smoothers [16, 2], but it seems to be not yet clear how to construct them for the complicated interior geometry of an electric machine. For the shape-design problem, a multigrid method had been successfully applied in a quite coarse discretization [17]. For a fine discretization, the difficulties with multigrid algorithms could be overcome in [26] by nonstandard "tricks" only. So we are not surprised that our standard GMG method does not converge.

In [25, 17], we had already found that the use of multigrid as a preconditioner essentially improves the convergence, in particular if the multigrid method converges slowly. Our parallel computations demonstrate that the combination with the PCG algorithm makes multigrid much more robust, even in that cases in which the pure multigrid method does not converge.

The global BPX preconditioner yields a robust method. It may be a good choice if a black-box solver is required, i.e., the components of the solver cannot be adapted well to the problem.

All the algorithms mentioned above are parallelized using the NODD data structure, they apply a parallel preconditioned Schur complement CG method as coarse grid solver. The communication overhead may differ between the solver types, but it is in the same order of magnitude for all of them, namely one order lower than the problem itself. The latter makes the NODD data structure attractive for implementing robust multilevel solvers in three dimensions, too.

Acknowledgement

This work is supported by the Austrian Science Fund – 'Fonds zur Förderung der wissenschaftlichen Forschung' – under project P 11215-TEC.

References

[1] T. Apel, G. Haase, A. Meyer and M. Pester, Parallel solution of finite element equation systems: efficient inter-processor communication, Preprint SPC 95_5, Technical University Chemnitz–Zwickau, Faculty for Mathematics, 1995.

[2] P. Bastian and G. Horton, Parallelization of robust multigrid methods: ILU factorization and frequency decomposition method, *SIAM J. Sci. Stat. Comput.*, 12 (1991), 1457–1470.

[3] J. H. Bramble, J. E. Pasciak and A. H. Schatz, The construction of preconditioners for elliptic problems by substructuring I – IV, *Mathematics of Computation*, 47 (1986), 103–134; 49 (1987), 1–16; 51 (1988), 415–430; 53 (1989), 1–24.

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

[5] R. S. Dembo, S. C. Eisenstat and T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.*, 19 (1982), 400–408.

[6] P. Deuflhard, Global inexact Newton methods for very large scale nonlinear problems, *IMPACT of Computing in Science and Engineering*, 3 (1991), 366–393.

[7] M. Dryja, A capacitance matrix method for Dirichlet problems on polygonal regions, *Numerische Mathematik*, 39 (1982), 51–64.

[8] M. Dryja and O. B. Widlund, Towards a unified theory of domain decomposition algorithms for elliptic problems. In T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund (eds.) *Domain decomposition methods for partial differential equations*, SIAM Philadelphia, 1990, 3–21. Proc. of the 3rd International Symposium, Houston, 1989.

[9] J. Goodman, V. R. Kohn and L. Reyna, Numerical study of a relaxed variational problem from optimal design, *Computer Methods in Applied Math. and Engineering*, 57 (1986), 107–127.

[10] M. Goppold, G. Haase, B. Heise and M. Kuhn, Preprocessing in BE/FE domain decomposition methods, Technical report no. 96-2, Institute for Mathematics, Johannes Kepler University Linz, 1996.

[11] G. Haase, B. Heise, M. Jung and M. Kuhn, FEM⊗BEM - a parallel solver for linear and nonlinear coupled FE/BE-equations, Report 94-16, DFG-Schwerpunkt "Randelementmethoden", University Stuttgart, 1994.

[12] G. Haase, B. Heise, M. Kuhn and U. Langer, Adaptive domain decomposition methods for finite and boundary element equations, Technical Report no. 95-2, Institute for Mathematics, Johannes Kepler University Linz, 1995.

[13] G. Haase, U. Langer and A. Meyer, The approximate Dirichlet domain decomposition method. Part I: An algebraic approach. Part II: Applications to 2nd-order elliptic boundary value problems, *Computing*, 47 (1991), 137–151 (Part I), 153–167 (Part II).

[14] G. Haase, U. Langer and A. Meyer, Domain decomposition preconditioners with inexact subdomain solvers, *J. of Num. Lin. Alg. with Appl.*, 1 (1992), 27–42.

[15] G. Haase, U. Langer, A. Meyer and S. V. Nepomnyaschikh, Hierarchical extension operators and local multigrid methods in domain decomposition preconditioners, *East-West J. Numer. Math.*, 2 (1994), 173–193.

[16] W. Hackbusch, *Multi–Grid methods and applications*, Springer, Berlin, 1985.

[17] B. Heise, Nonlinear field calculations with multigrid-Newton methods, *IMPACT of Computing in Science and Engineering*, 5 (1993), 75–110.

[18] B. Heise, Analysis of a fully discrete finite element method for a nonlinear magnetic field problem, *SIAM J. Numer. Anal.*, 31 (1994), 745–759.

[19] B. Heise, Nonlinear simulation of electromagnetic fields with domain decomposition methods on MIMD parallel computers, *Journal of Computational and Applied Mathematics*, 63 (1995), 373–381.

[20] B. Heise, Parallel solvers for coupled FEM-BEM equations with applications to nonlinear magnetic field problems. In W. Hackbusch and G. Wittum (eds.) *Numerical Treatment of Coupled Systems*, volume 51 of *Notes on Numerical Fluid Mechanics*, Vieweg, Braunschweig, Wiesbaden, 1995, 73–85.

[21] B. Heise, Nonlinear field simulation with FE domain decomposition methods on massively parallel computers, *Surveys on Mathematics for Industry*, 1996, to appear.

[22] B. Heise and M. Jung, Comparison of parallel solvers for nonlinear elliptic problems based on domain decomposition ideas, Report no. 494, Institute for Mathematics, Johannes Kepler University Linz, 1995. Submitted for publication.

[23] B. Heise and M. Kuhn, Parallel solvers for linear and nonlinear exterior magnetic field problems based upon coupled FE/BE formulations, *Computing*, 56 (1996), 237–258.

[24] M. Jung, On the parallelization of multi-grid methods using a non-overlapping domain decomposition data structure, *Appl. Numer. Math.*, 1997. To appear. Also Preprint SPC 95_27, Technical University Chemnitz–Zwickau, Faculty for Mathematics, 1995.

[25] M. Jung, U. Langer, A. Meyer, W. Queck and M. Schneider, Multigrid preconditioners and their applications. In G. Telschow (ed.) *Third Multigrid Seminar, Biesenthal 1988*, Report R–MATH–03/89, Karl–Weierstrass–Institute, Berlin, 1989, 11–52.

[26] B. Kawohl, J. Stara and G. Wittum, Analysis and numerical studies of a problem of shape design, *Archive of Rational Mechanics and Analysis*, 114 (1991), 349–363.

[27] S. V. Nepomnyaschikh, Domain decomposition and multilevel techniques for preconditioning operators, Preprint SPC 95_30, Technical University Chemnitz–Zwickau, Faculty for Mathematics, 1995.

[28] B. F. Smith, Domain decomposition algorithms for the partial differential equations of linear elasticity, Technical Report 517, Dep. of Computer Science, Courant Institute, New York, 1990.

[29] C. H. Tong, T. F. Chan and C. J. Kuo. A domain decomposition preconditioner based on a change to a multilevel nodal basis, *SIAM J. Sci. Stat. Comput.*, 12 (1991), 1486–1495.