Comparison of Quasi Minimal Residual and Bi-Conjugate Gradient Iterative Methods to solve Complex Symmetric Systems arising from Time-Harmonic Magnetic Simulations

H. De Gersem[①], D. Lahaye[②], S. Vandewalle[②] and K. Hameyer[①]

[①]Katholieke Universiteit Leuven, Dep. EE (ESAT), Div. ELEN, Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium
[②]Katholieke Universiteit Leuven, Dep. Computer Science, Celestijnenlaan 200A, B-3001 Leuven, Belgium

Abstract - Finite element discretizations of low-frequency time-harmonic magnetic problems lead to sparse, complex symmetric systems of linear equations. The question arises which Krylov subspace methods are appropriate to solve such systems. The Quasi Minimal Residual method combines a constant amount of work and storage per iteration step with a smooth convergence history. These advantages are obtained by building a quasi minimal residual approach on top of a Lanczos process to construct the search space. Solving the complex systems by transforming them to equivalent real ones of double dimension has to be avoided as such real systems have spectra that are less favourable for the convergence of Krylov based methods. Numerical experiments are performed on electromagnetic engineering problems to compare the Quasi Minimal Residual method to the Bi-Conjugate Gradient method and the Generalized Minimal Residual method.
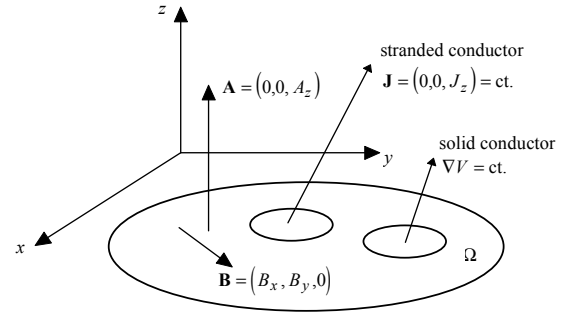
Fig. 1. 2D cartesian geometry.

Minimal Residual Method (GMRES). In this paper, the Quasi Minimal Residual (QMR) method is studied as an alternative for the latter two.

I. INTRODUCTION

Sinusoidally excited eddy current problems arise in e.g. the design and optimization of induction furnaces, transformers and alternating current machines. They are commonly treated as time-harmonic magnetic field problems [1]. Their finite element discretizations result in sparse, complex symmetric systems of equations. Solving those systems often absorbs more than 90% of the overall CPU-time of the numerical simulation [2]. Hence, a study of iterative methods for this specific kind of matrices is appropriate.

To solve large sparse systems, Krylov subspace iterative solvers have become competitive and often outperform direct solving techniques [3], [4], [5], [6], [7]. This investigation deals with the efficiency and robustness of different Krylov subspace solvers as solvers for complex symmetric systems. Such systems were also considered in [8] and [9]. In the electromagnetic computing community, one typically uses the Bi-Conjugate Gradient Method (BiCG) or the Generalized

II. TIME-HARMONIC MAGNETIC FORMULATIONS

The relevant equations are derived from the Maxwell equations by taking the following consideration into account. The dimensions of the models are small compared to the electromagnetic wave length corresponding to the frequency range of 0-1000 Hz. Therefore, the displacement-current term $\partial \mathbf{D}/\partial t$ in Maxwell's equations is neglected [1]. The divergence-free condition for the magnetic flux density $\mathbf{B}$ is ensured by expressing $\mathbf{B}$ in terms of the magnetic vector potential $\mathbf{A}$ as $\mathbf{B} = \nabla \times \mathbf{A}$. Faraday's law is reduced to

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t}, \qquad (1)$$

with $\mathbf{E}$ the electric field strength and V the electric scalar potential. The current density $\mathbf{J}$ is related to $\mathbf{E}$ by

$$\mathbf{J} = \sigma \mathbf{E} = -\sigma \nabla V - \sigma \frac{\partial \mathbf{A}}{\partial t} = \mathbf{J}_s + \mathbf{J}_e, \qquad (2)$$

where σ is the electric conductivity, $\mathbf{J}_s$ the source current density and $\mathbf{J}_e$ the eddy current density. Ampère's law in terms of $\mathbf{A}$ and V is

$$\nabla \times (\nu \nabla \times \mathbf{A}) + \sigma \frac{\partial \mathbf{A}}{\partial t} = -\sigma \nabla V, \qquad (3)$$

where ν is the magnetic reluctivity. In a lot of devices, the magnetic field varies sinusoidally in time. In that case $\mathbf{A}$ and V are represented by the complex phasors $\hat{\mathbf{A}}$ and $\hat{V}$ as

$$\mathbf{A} = \mathrm{Re}\left\{ \hat{\mathbf{A}} e^{j\omega t} \right\} \qquad (4)$$

and

$$V = \text{Re}\left\{\hat{V}e^{j\omega t}\right\}, \tag{5}$$

where $\omega = 2\pi f$ and f is the frequency. For a 2D cartesian coordinate system (Fig. 1), (3) becomes the complex Helmholtz equation

$$\frac{\partial}{\partial x}\left(\nu \frac{\partial \hat{A}_z}{\partial x}\right) + \frac{\partial}{\partial y}\left(\nu \frac{\partial \hat{A}_z}{\partial y}\right) - j\omega\sigma \hat{A}_z = \sigma \nabla \hat{V}. \tag{6}$$

The weak formulation of (6) is discretized by means of linear and triangular finite elements giving

$$\sum_k \left(K_{kl} + j\omega L_{kl}\right) \hat{A}_{zl} = T_k \tag{7}$$

where

$$K_{kl} = \int_\Omega \nu \nabla N_k \cdot \nabla N_l \, d\Omega \tag{8}$$

$$L_{kl} = \int_\Omega \sigma N_k N_l \, d\Omega \tag{9}$$

and

$$T_k = \int_\Omega \left(-\sigma\nabla\hat{V}\right) N_k \, d\Omega \tag{10}$$

N_k is the nodal form function associated with the mesh node k, $\hat{A}_{zk}$ is the coefficient in the discrete solution corresponding to form function N_k and Ω is the region of interest. The resulting system (7) will further be denoted by

$$Ax = b \tag{11}$$

with $A = K + jL$. It is sparse and complex symmetric: $A = A^T$.

External circuit connections are taken into account by adding dense algebraic equations to the system. The matrix symmetry is preserved by choosing appropriate extra unknown currents and voltages [10]. In the case of rotating electric devices, the motional electromagnetic force enters as the $\sigma\vec{\mathbf{v}}\times\mathbf{B}$-term in the differential equation (3). It is possible to incorporate this term in the $\sigma\,\partial\mathbf{A}/\partial t$ term under three conditions: the geometry is invariant under relative motion, the mechanical speed is linearly related to the frequency and the magnetic field is sinusoidally distributed along the airgap between the moving and the stand-still part of the model. Induction motors can be modelled in this way with a sufficient accuracy for the computation of stationary regimes [11].

III. KRYLOV SUBSPACE METHOD FOR COMPLEX SYMMETRIC SYSTEMS

The Krylov subspace generated by the matrix A and vector v is the subspace

$$K^m(A,v) = \text{span}\left(v, Av, A^2v, \dots, A^{m-1}v\right). \tag{12}$$

Given an initial guess x_0, Krylov subspace methods solve the linear system (11) iteratively by computing iterands

$x_m \in x_0 + K^m(A, r_0)$, with $r_0 = b - Ax_0$ the initial residual, converging to the exact solution for increasing iteration step m. Let the columns of $V_m = [v_1, \dots, v_m]$ form a basis for $K^m(A, r_0)$. Vector x_m is then expressed as $x_m = x_0 + V_m y_m$ where y_m is a vector of coefficients. Krylov subspace methods differ in the construction of the basis V_m and the criterium employed for determining y_m.

A. Krylov Subspace Basis Construction

Two methods are commonly used to construct a basis for $K^m(A, r_0)$: *Arnoldi*'s method and *Lanczos*'s method.

A.1 Arnoldi's method

Assume that a set of m orthonormal vectors $\{v_1, \dots, v_m\}$ has already been computed. The Arnoldi method expands this set to a orthonormal basis for $K^{m+1}(A, r_0)$ by orthogonalizing $t = Av_m$ with respect to this set and normalizing the resulting vector to give v_{m+1}. The matrix V_m and the vector v_{m+1} then satisfy

$$AV_m = V_{m+1}\widetilde{H}_{m+1,m} \text{ where } V_{m+1}^H V_{m+1} = I_{m+1}. \tag{13}$$

Here, $\widetilde{H}_{m+1,m}$ is an upper $(m+1)\times m$ Hessenberg matrix with elements defined by the Arnoldi algorithm. The $(m+1)$-th basis vector v_{m+1} is constructed using all previous vectors $\{v_1, \dots, v_m\}$, resulting in *long* recurrences. These long recurrences are expensive: both the memory requirements (storing all vectors v_i) and the CPU-time (orthogonalizing with respect to all v_i) grow linearly with m.

A.2 Lanczos method

The cost of long term recurrences can be avoided by resorting to a *bi*-orthogonalization procedure. Then the basis V_m is orthogonalized with respect to a set of basis vectors $\{w_1, \dots, w_m\}$ of a second Krylov subspace L_m. In this case, we have that

$$W_m^H A V_m = D_m T_{m,m} \text{ where } W_m^H V_m = D_m \tag{14}$$

and where D_m and $T_{m,m}$ are diagonal and tridiagonal respectively. For complex symmetric matrices, a convenient choice for the space L_m is

$$L_m = \overline{K^m}(A, r_0) = \left\{\overline{v}\,\middle|\,v \in K^m(A, r_0)\right\}. \tag{15}$$

Then W_m can be taken as $\overline{V}_m$ and the V_m-sequence is

$$AV_m = V_{m+1}\widetilde{T}_{m+1,m} \text{ where } V_m^T V_m = D_m. \tag{16}$$

$\widetilde{T}_{m+1,m}$ is the $(m+1) \times m$ tridiagonal matrix formed as an expansion of $T_{m,m}$. In this way the need of constructing the space L_m explicitly is avoided. As the matrix $T_{m,m}$ in (14) is tridiagonal, the basis V_m can be computed by *short recurrences*: computing v_{m+1} requires as basis vectors only v_m and v_{m-1}. The Lanczos algorithm (14) corresponds to the Arnoldi algorithm where the bilinear form $< v, w >_T = w^T v$ is used instead of the inner product $< v, w >_H = w^H v$. This Lanczos algorithm may break down in the event that a generated basis vector is a quasi-null vector, i.e. a vector for which $< v, v >_T = 0$ even if $v \neq 0$.

B. Projection and Minimal Residual Criteria

Once the basis V_m has been constructed, the coefficient vector y_m can be computed by either a *Ritz-Galerkin*, *Petrov-Galerkin* or a *(quasi-) minimal residual* approach.

B.1 Ritz-Galerkin approach

The Ritz-Galerkin approach requires that the residual vector $r_m = b - Ax_m$ is orthogonal to the Krylov subspace with respect to the proper inner product:

$$V_m^H r_m = 0. \tag{17}$$

B.2 Petrov-Galerkin approach

In the Petrov-Galerkin approach, the residual r_m is made orthogonal to the space L_m. With the choice of L_m in (15), this requirement is equivalent to

$$V_m^T r_m = 0. \tag{18}$$

B.3 (Quasi) Minimal Residual approach

In the minimal residual approach, the Euclidean norm $\|r_m\|_2$ is minimized over the space $K^m(A, r_0)$ at each iteration step.

The basis vectors generated by the Lanczos algorithm are $< , >_T$-orthogonal, rather then $< , >_H$-orthogonal. Hence, the approach

determine y_m such that

$$
\begin{aligned}
\|r_m\|_2 &= \left\| V_{m+1}\left(\|r_0\|_2 e_1 - \widetilde{T}_{m+1,m} y_m \right) \right\|_2 \\
&= \min_{y \in C^m} \left\| V_{m+1}\left(\|r_0\|_2 e_1 - \widetilde{T}_{m+1,m} y \right) \right\|_2
\end{aligned} \tag{19}
$$

requires solving a least-squares problems that leads to an algorithm for which work and storage per iteration step grow

	Construction basis	
	Lanczos $< , >_T$	Arnoldi $< , >_H$
Ritz-Galerkin	-	FOM
(Quasi) Minimal Residual	QMR	GMRES
Petrov-Galerkin	COCG	-

linearly with m. Therefore, Freund [8] suggested to replace the true minimization by the following quasi-minimization:

determine y_m such that

$$
\begin{aligned}
\|r_m\|_2 &= \left\| \omega_1 \|r_0\|_2 e_1 - \Omega_{m+1} \widetilde{T}_{m+1,m} y_m \right\|_2 \\
&= \min_{y \in C^m} \left\| \omega_1 \|r_0\|_2 e_1 - \Omega_{m+1} \widetilde{T}_{m+1,m} y \right\|_2
\end{aligned} \tag{20}
$$

where $\Omega_{m+1} = \text{diag}(\omega_1, ..., \omega_{m+1})$ is a scaling matrix with weights

$$\omega_k = \|v_k\|_2. \tag{21}$$

The Lanczos and Arnoldi algorithms are combined with one of the above criteria to obtain different Krylov subspace methods for solving (11) (Table I). For real matrices, the Conjugate Orthogonal Conjugate Gradient (COCG) and Quasi Minimal Residual (QMR) algorithms reduce to the Conjugate Gradient (CG) and Minimal Residual (MINRES) algorithms respectively. Both the Full Orthogonalization Method (FOM) and the Generalised Minimal Residual method (GMRES) use long recurrences. GMRES minimizes the residual at each iteration. FOM minimizes the error.

Thanks to the structure of the matrix $\widetilde{T}_{m+1,m}$, the least-squares problem on the right-hand side of (20) can be solved appropriately using a Givens QR factorization. Numerically stable implementations of Givens rotations for complex variables are available in recent BLAS I implementations. Numerical stability of the algorithm further requires the three-terms recurrences for generating the basis to be implemented as coupled two-terms recurrences [12], [13].

Both the COCG and coupled two-terms QMR algorithms are susceptible to breakdown at two stages: in the Lanczos algorithm due to quasi-null basis vectors (breakdown of the first kind) and the linear system that defines the next iterate becoming singular (breakdown of the second kind). Both breakdowns have been observed in our numerical experiments. To overcome these breakdowns, a *look-ahead* strategy [14], [15], [16] needs to be incorporated into the algorithm.

IV. PRECONDITIONING QMR

To derive a preconditioned version of the QMR algorithm, we apply the QMR algorithm to the transformed system $\hat{A}\hat{x} = \hat{b}$ where $\hat{A} = C^{-T}AC^{-1}$, $\hat{x} = Cx$, $\hat{b} = C^{-T}b$ and C is such that the preconditioning matrix M can be written as
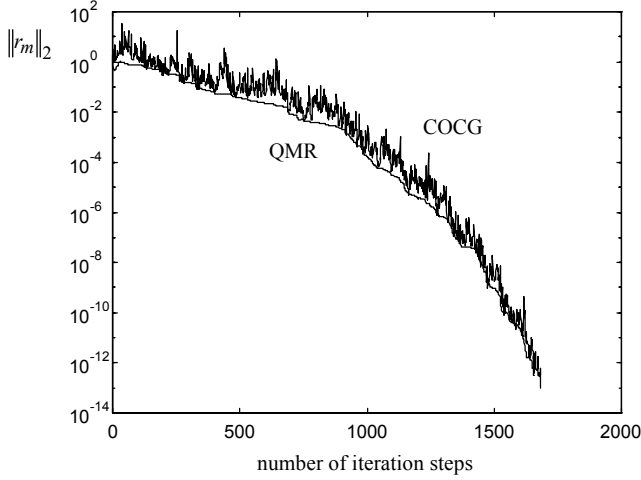
$$M = C^T C. \tag{22}$$

Fig. 2. Convergence histories of COCG (upper) and QMR (lower).
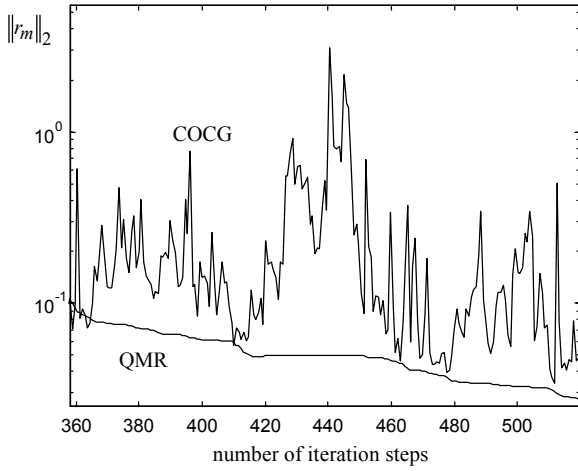

Fig. 3. Detail of the convergence histories of COCG (upper) and QMR (lower).

The preconditioner M is thus complex and symmetric. We proceed in a way similar to Section 10.3 in [17]. Due to (21), the resulting algorithm requires in the m-th iteration step the normalization of the vector $C^{-1}\tilde{v}_m$, obtained from the Lanczos process by $<,>_T$-orthogonalizing $\hat{A}C^{-1}v_{m-1}$ with respect to $\left\{C^{-1}v_{m-2}, C^{-1}v_{m-1}\right\}$ and thus the computation of

$$\left\|C^{-1}\tilde{v}_m\right\|_2. \tag{23}$$

For *real* matrices (23) can be rewritten in such a way that the matrix C disappears, as follows

$$\left\|C^{-1}\tilde{v}_m\right\|_2 = <\tilde{v}_m, C^{-T}C^{-1}\tilde{v}_m>_T = <\tilde{v}_m, M^{-1}\tilde{v}_m>_T \tag{24}$$

with M defined in (22). For *complex* matrices however, we have that

$$\left\|C^{-1}\tilde{v}_m\right\|_2 = <\tilde{v}_m, C^{-H}C^{-1}\tilde{v}_m>_H \neq <\tilde{v}_m, M^{-1}\tilde{v}_m>_H. \tag{25}$$
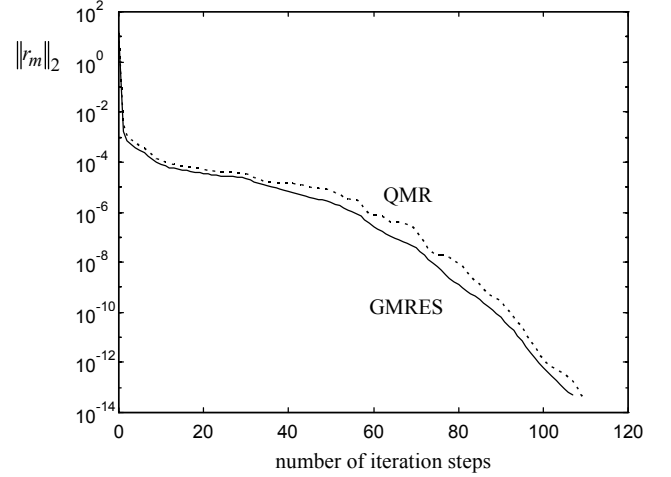

Fig. 4: Convergence histories of SSOR-QMR (upper) and SSOR-GMRES (lower).

For the Symmetric Successive Over-Relaxation (SSOR) preconditioner, the factor C can be explicitly constructed. For other preconditioners, the need for explicitly constructing the factor C, can be avoided by defining other normalization weights in (21).

V. COMPARISON OF QMR WITH COCG

As COCG does not have a minimal residual property, its convergence behaviour can be irregular (Fig. 2 and Fig. 3). This behaviour prevents COCG to be numerically robust. One can prove the following estimate for the difference at step m between the iteratively updated residual $\|r_m\|_2$ and the true residual $\|b - Ax_m\|_2$ (Section 5.3 in [6]):

$$\left| \|r_m\|_2 - \|b - Ax_m\|_2 \right| \leq m\overline{\xi}f(A) \max_{j=1,\ldots,m} \|r_j\|_2 \tag{26}$$

where $\overline{\xi}$ is the relative machine precision and $f(A)$ is a function of matrix A. QMR on the other hand has a smoother convergence history.

The overall number of iterations needed by the COCG and QMR algorithms to reach a prescribed tolerance is about the same (Fig. 2). This can be explained by taking a closer look to the linear systems that the COCG and QMR algorithms solve at each step m to determine y_m. For the COCG and QMR algorithms, these linear systems only differ by an additional row in the QMR case. Details can be found in e.g. in Section 6.5.5 in [3].

VI. COMPARISON OF QMR WITH GMRES

A natural question is by how much the quasi-minimization computed by QMR differs from the true minimization computed by GMRES. In Fig. 4 the convergence histories of SSOR-QMR and SSOR-GMRES are plotted for a representative test case, small enough to run a non-restarted
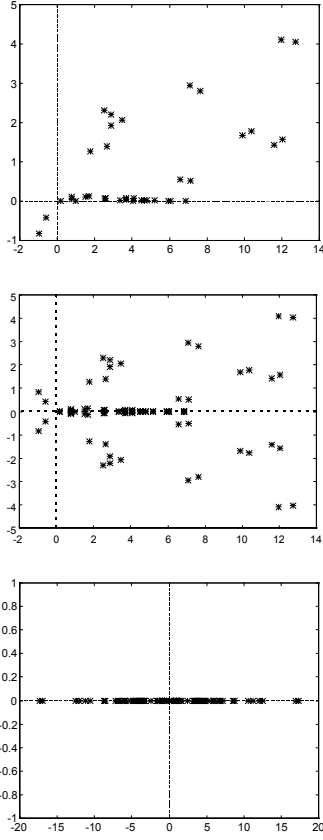
$$Ax = b$$

$$A_* \begin{bmatrix} \mathrm{Re}\, x \\ \mathrm{Im}\, x \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\, b \\ \mathrm{Im}\, b \end{bmatrix},$$

$$A_* = \begin{bmatrix} \mathrm{Re}\, A & -\mathrm{Im}\, A \\ \mathrm{Im}\, A & \mathrm{Re}\, A \end{bmatrix}$$

$$A_{**} \begin{bmatrix} \mathrm{Re}\, x \\ -\mathrm{Im}\, x \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\, b \\ \mathrm{Im}\, b \end{bmatrix},$$

$$A_{**} = \begin{bmatrix} \mathrm{Re}\, A & \mathrm{Im}\, A \\ \mathrm{Im}\, A & -\mathrm{Re}\, A \end{bmatrix}$$

Fig. 5. Comparison of the spectrum of the complex symmetric system matrix to the spectra of two equivalent real systems of double dimension.

version of GMRES. From this figure it is concluded that, for the examples considered here, the quasi-minimization approximates the true minimization rather well. Here, QMR behaves like the optimal method. This means that efforts in speeding up computations should be directed towards finding more efficient (possibly multilevel type) preconditioners [18].

VII. COMPLEX VERSUS EQUIVALENT REAL SYSTEMS

Splitting the real and imaginary parts of the complex system matrix leads to two possible equivalent real systems of double dimension [8] (Fig. 5). For the spectra of the matrices A_* and A_{**} the following properties hold:

$$\sigma(A_*) = \sigma(A) \cup \overline{\sigma(A)} \tag{27}$$

and

$$\sigma(A_{**}) = \left\{ \lambda \in R \middle| \lambda^2 \in \sigma(\overline{A}A) \right\} \tag{28}$$

respectively (Fig. 5). Both the spectra for A_* and A_{**} are less favourable for the convergence of Krylov subspace iterative methods than the spectrum of the original complex matrix A.
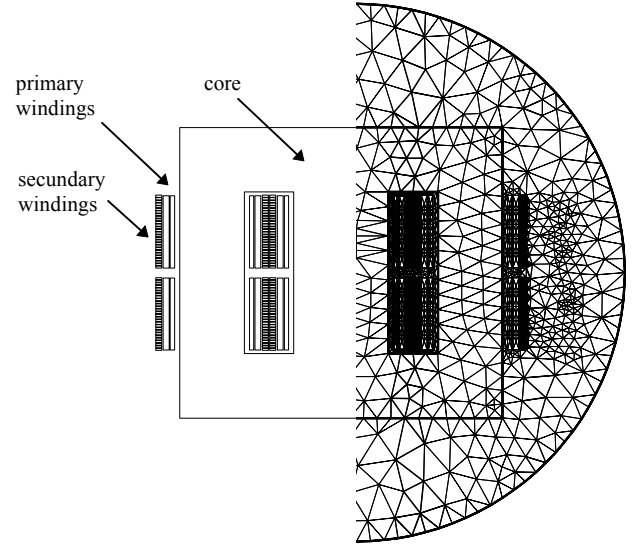

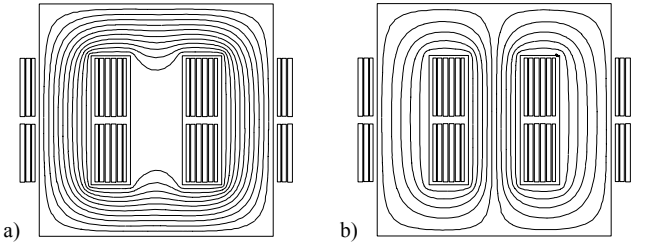Fig. 6. Outline and mesh of a three-phase transformer.

a) b)

Fig. 7. Flux line plots of the three-phase transformer at two different time instants.
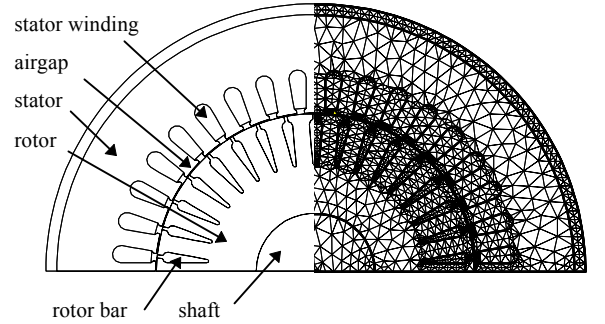

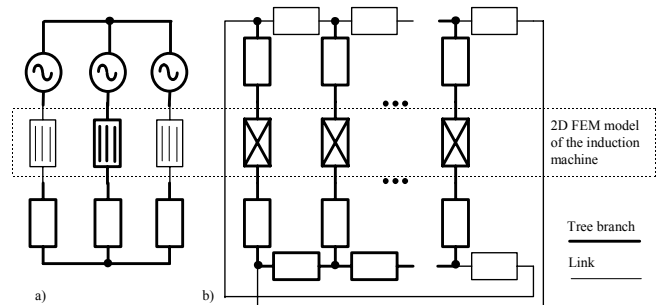Fig. 8. Outline and mesh of an Induction motor.

a) b)

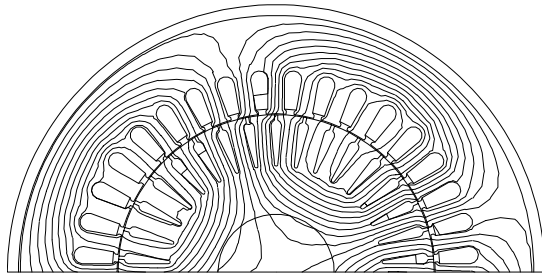Fig. 9. External circuit connection of the induction motor.

Fig. 10. Flux line plot of the induction motor.

TABLE II
SIZE OF THE CONSIDERED MODELS

	Model	Mesh nodes	Circuit unknowns	Number of non-zeros in the matrix
1	Transformer	5565	154	28882
2	Transformer	17807	154	83817
3	Induction motor	2857	69	12760
4	Induction motor	10931	69	48614

TABLE III
NUMBER OF ITERATION STEPS OF THE PRECONDITIONED AND NOT-PRECONDITIONED VERSION OF COCG AND QMR

Model	COCG	SSOR-COCG	QMR	SSOR-QMR
1	2985	1548	2975	1538
2	3384	1666	3325	1638
3	1188	1048	1184	1045
4	2099	1657	2093	1652

VIII. PRACTICAL EXAMPLES

A three-phase power transformer (Fig. 6) consists of high voltage windings (primary) and low voltage windings (secondary) Eddy current effects are neglected in the high voltage windings. The secondary windings consist of hollow copper conductors in which skin effect is not negligible. The primary is voltage driven. The secondary is connected to a resistive load. Fig. 7 shows the flux line plots of the magnetic field at two instants of time.

A three-phase induction motor (Fig. 8) with 4 poles, 36 stator slots and 34 rotor slots is computed under loaded condition. The stator end-windings and rotor end-rings are modelled by external electric impedances (Fig. 9). The stator windings are voltage-driven. A magnetic flux line plot is shown in Fig. 10.

Table II compares the numbers of nodes, extra circuit equations and non-zero matrix elements for different discretizations of both models. Table III presents the number of iterationsteps of COCG, QMR and their preconditioned variants for the studied models.

IX. CONCLUSIONS

A Quasi Minimal Residual approach is used as Krylov subspace projection method to solve the sparse and complex symmetric matrices arising from low-frequency time-harmonic magnetic finite element discretizations. QMR converges much smoother compared to COCG. The convergence rate of QMR approximates the one of GMRES, indicating that QMR is a valuable low-cost iterative method to solve the complex symmetric matrices of these discretizations. Further improvement should concentrate on the problem of preconditioning and on incorporating look-ahead strategies to cope with breakdown. The methods are applied to electromagnetic problems of technical importance.

REFERENCES

[1] P.P. Silvester and R.L. Ferrari, *Finite Elements for Electrical Engineers*, 3rd ed., Cambridge University Press, New York, 1996.
[2] U.Pahner, R.Mertens, H. De Gersem, K. Hameyer and R.Belmans, "A parametric finite element environment tuned for numerical optimization", *Proc. Compumag*, Rio de Janeiro, Brazil, Nov. 2-6, 1997, pp. 505-506.
[3] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
[4] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Vol. 17 of *Frontiers in Applied Mathematics*, SIAM, Philadelphia, PA, 1997.
[5] R.W. Freund, G.H. Golub and N.M. Nachtigal, "Iterative solution of linear systems", *Acta Numerica*, pp. 1-44, 1992.
[6] G.H. Golub and H.A. van der Vorst, "Closer to the solution: iterative linear solvers", in *The State of the Art in Numerical Analysis*, I.S. Duff and G.A. Watson, eds, Claredon Press, 1997, pp. 63-92.
[7] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, Ch. Romine, H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed., SIAM, Philadelphia, PA, 1994.
[8] R.W. Freund, "Conjugate Gradient-type methods for linear systems with complex symmetric coefficient matrices", *SIAM J. Sci. Comput.*, Vol. 13, pp. 425-448, January 1992.
[9] H.A. van der Vorst, J.B.M. Melissen, "A Petrov-Galerkin type method for solving Ax=b, where A is symmetric complex", *IEEE Trans. on Magn.*, Vol. 26, pp. 706-708, March 1990.
[10] H. De Gersem, R. Mertens, U. Pahner, R. Belmans and K. Hameyer, "A topological method used for field-circuit coupling", *IEEE Trans. on Magn.*, Vol. 34, September 1998, pp. 3190-3191.
[11] R. De Weerdt, E. Tuinman, K. Hameyer and R. Belmans, "Finite element analysis of steady state behaviour of squirrel cage induction motors: a comparison with measurements", *Proc. International Conference on Electrical Machines (ICEM)*, Vol. 3, Vigo, Spain, September 10-12, 1996, pp. 63-67.
[12] R.W. Freund, N.M. Nachtigal, "An implementation of the QMR method based on coupled two-term recurrences", *SIAM J. Sci. Comput.*, Vol. 15, pp. 313-337, March 1994.
[13] M.H. Gutknecht and Z. Strakos, "Accuracy of three-term and two-term recurrences for Krylov space solvers", *SIAM J. Matrix Anal. Applics.*, 1999, in press.
[14] R.W. Freund and N.M. Nachtigal, "An implementation of the look-ahead Lanczos algorithm, part ii", RIACS Technical Report 90.46, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffett Field, California, December 1990.
[15] R.W. Freund, M.H. Gutknecht and N.M. Nachtigal, "An implementation of the look-ahead Lanczos algorithm for non-hermitian matrices", *SIAM J. Sci. Comput.*, Vol. 14, pp. 137-158, 1993.
[16] C. Brezinski, M. Redivo Zaglia and H. Sadok, "New look-ahead Lanczos-type algorithms for linear systems", *Numer. Math.*, 1999, in press.
[17] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 2nd ed., 1992.
[18] R. Mertens, H. De Gersem, R. Belmans, K. Hameyer, D. Lahaye, S. Vandewalle and D. Roose, "An algebraic multigrid method for solving very large electromagnetic systems", *IEEE Trans. on Magn.*, Vol. 34, Sept. 1998, pp. 3327-3330.