Comparison of Quasi Minimal Residual and Bi-Conjugate Gradient Iterative Methods solving the Complex Symmetric Systems of Time-Harmonic Magnetic Problems

H. De Gersem\textsuperscript{a}, D. Lahaye\textsuperscript{b}, S. Vandewalle\textsuperscript{a} and K. Hameyer\textsuperscript{a}
\textsuperscript{a}Katholieke Universiteit Leuven, Dep. EE (ESAT), Div. ELEN, Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium
\textsuperscript{b}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.

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

II. TIME-HARMONIC MAGNETIC FORMULATIONS

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 $\mathbf{B} = \mathbf{V} \times \mathbf{A}$. Faraday's law is reduced to

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t},$$

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

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

where $\sigma$ 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

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

where $\nu$ 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

$$\mathbf{A} = \text{Re}\{\hat{\mathbf{A}} e^{j\omega t}\}$$

and

$$V = \text{Re}\{\hat{V} e^{j\omega t}\},$$
where \( \omega = 2\pi f \) and \( f \) is the frequency. For a 2D cartesian coordinate system (Fig. 1), (3) becomes the complex Helmholtz equation

\[
\nabla \cdot \left( \epsilon \nabla \phi \right) - \omega^2 \phi = v \rho, 
\]

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

\[
\sum_k \left( K_{kl} + j \alpha L_{kl} \right) \Delta_{cl} = T_k
\]

(7)
where

\[
K_{kl} = \int \nabla N_k \nabla N_l \, d\Omega \\
L_{kl} = \int \sigma N_k \nabla N_l \, d\Omega
\]

and

\[
T_k = \left[ \left( -j \sigma \nabla \hat{\phi} \right) \right] N_k \, d\Omega
\]

(8)
(9)
(10)

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

\[
Ax = b
\]

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 \( \sigma \mathbf{V} \times \mathbf{B} \)-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

\[
K^m(A,v) = \text{span} \left\{ v, Av, A^2 v, \ldots, A^{m-1} v \right\}.
\]

(12)
Given an initial guess \( x_0 \), Krylov subspace methods solve the linear system (11) iteratively by computing iterands \( x_m \) with \( r_0 = b - Ax_0 \) the initial residual, converging to the exact solution for increasing iteration step \( m \). The columns of \( V_m = [v_1, \ldots, v_m] \) form a basis for \( K^m(A,r_0) \). \( x_m \) is 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 a set of \( m \) orthonormal vectors \( \{v_1, \ldots, v_m\} \) has already been computed. The Arnoldi method expands this set to an 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

\[
AV_m = V_{m+1} \tilde{H}_{m+1,m} \quad \text{where} \quad V_m^H V_{m+1} = I_{m+1}.
\]

(13)

Here, \( \tilde{H}_{m+1,m} \) 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 \( \{v_1, \ldots, v_m\} \), resulting in long recurrences. These long recurrences are expensive: both the memory requirements (storing all vectors \( v_j \)) and the CPU-time (orthogonalizing with respect to all \( v_j \)) 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 \( \{w_1, \ldots, w_m\} \) of a second Krylov subspace \( L_m \). In this case, we have that

\[
AV_m = D_m \tilde{L}_{m+1,m} \quad \text{where} \quad V_m^H V_{m+1} = I_{m+1},
\]

(14)
and where \( D_m \) and \( \tilde{L}_{m+1,m} \) are diagonal and \( (m+1) \times m \) tridiagonal matrices respectively. The submatrix of \( \tilde{L}_{m+1,m} \) formed by its first \( m \) rows will be denoted by \( L_m \). For complex symmetric matrices, a convenient choice for the space \( L_m \) is

\[
L_m = \overline{K^m(A,r_0)} = \left\{ v \in K^m(A,r_0) \right\}.
\]

(15)
Condition (14) reduces to

\[
AV_m = D_m \tilde{T}_{m+1,m} \quad \text{where} \quad V_m^H V_{m+1} = D_m.
\]

(16)
In this way the need of constructing the space $L_m$ explicitly is avoided. As the matrix $\tilde{T}_{m+1,m}$ 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 $\langle v, w \rangle_T = w^T v$ is used instead of the inner product $\langle v, w \rangle_H = w^H v$. As a consequence, 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 $\langle v, v \rangle_T = 0$ 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 $r_m = b - Ax_m$ satisfies

$$ r_m \perp \perp K^m(A, r_0) \Leftrightarrow V_m^H r_m = 0. \quad (17) $$

#### 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

$$ r_m \perp \perp K^m(A, r_0) \Leftrightarrow V_m^T r_m = 0. \quad (18) $$

#### 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 $\langle , \rangle_T$-orthogonal, rather then $\langle , \rangle_H$-orthogonal. Hence, the approach

$$ r_m = b - Ax_m $$. 

determine $y_m$ such that

$$ \|r_m\|_2 = \|r_m^\perp\|_2 + \|r_m\|_2 $$

represents the solving of a least-squares problem that leads to an algorithm for which work and storage per iteration step grows linearly with $m$. Therefore, Freund [8] suggested to replace the true minimization by the following quasi-minimization:

$$ \min_{y \in C^m} \|r_0\|_2 + \|r_0\|_2 $$. 

such that all terms in the minimization (20) have equal weight.

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, but whereas GMRES minimizes the residual at each iteration, FOM does not.

Due to the structure of the matrix $\tilde{T}_{m+1,m}$, 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

\[
\begin{array}{c|c|c|c}
\text{TABLE I} & \text{A TAXONOMY OF KSP METHODS FOR COMPLEX SYMMETRIC SYSTEMS} \\
\hline
\text{Construction basis} & \text{Ritz-Galerkin} & \text{Petrov-Galerkin} & \text{FOM} & \text{GMRES} \\
\text{Lanczos} & - & \text{QMR} & \text{GMRES} & - \\
\text{Arnoldi} & - & \text{GMRES} & - & - \\
\end{array}
\]

To derive a preconditioned version of the QMR algorithm, we apply the QMR algorithm to the transformed system
\[ A \times x = b \]

where

\[ A = A^T \]
\[ C = M^{-1} \]
\[ b = C^T b \]

and \( C \) is such that the preconditioning matrix \( M \) can be written as

\[ M = C^T C. \]  (22)

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 \( \tilde{C}v_m \), obtained from the Lanczos process by \( \langle \cdot, \cdot \rangle_T \)-orthogonalizing \( AC^{-1}v_m \) with respect to \( \left\{ C^{-1}v_{m-2}, C^{-1}v_{m-1} \right\} \) and thus the computation of

\[ \left\| C^{-1}v_m \right\|_2. \]  (23)

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

\[ \left\| C^{-1}v_m \right\|_2 = \langle \tilde{v}_m, C^{-1}v_m \rangle_T = \langle \tilde{v}_m, \tilde{v}_m \rangle_T = M^{-1}v_m > T \]  (24)

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

\[ \left\| C^{-1}v_m \right\|_2 = \langle \widetilde{v}_m, C^{-1}v_m \rangle_H = \langle \widetilde{v}_m, \tilde{v}_m \rangle_H \neq \langle \widetilde{v}_m, M^{-1}\tilde{v}_m \rangle_H. \]  (25)

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, as one can prove that the following estimate for the difference at step \( m \) between the iteratively updated residual \( \left\| r_m \right\|_2 \) and the true residual \( \| b - Ax_m \|_2 \) holds (Section 5.3 in [6]):

\[ \left\| r_m \right\|_2 - \| b - Ax_m \|_2 \leq m\xi f(A) \max_{j=1,...,m} \left\| r_j \right\|_2, \]  (26)

where \( \xi \) 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 to ask 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 be able to run a non-restarted version of GMRES. From this figure it is concluded that, for the practical examples considered here, the quasi-minimization approximates the true minimization rather well. Hence, for the studied examples, 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:

\[
\sigma(A) = \sigma(A) \cup \sigma(A)
\]

(27)

and

\[
\sigma(A^{**}) = \{ \lambda \in \mathbb{R} | \lambda^2 \in \sigma(\bar{A}A) \}
\]

(28)

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

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

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

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

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

**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 iteration steps of COCG, QMR and their preconditioned variants for the studied models.

---

**REFERENCES**


