Analysis of Finite Element Method Equation Solvers

Alius Noreika, Paulius Tarvydas

Department of Electronics Engineering, Kaunas University of Technology, Studentu str. 50, LT-51368, Kaunas, Lithuania. E-mail: alius.noreika@stud.ktu.lt

Abstract. Direct and iterative solution techniques of equation systems used in finite element method are analyzed. Comparison of these techniques, their advantages and drawbacks are characterized and recommendations for their applications are presented. Methods are analyzed by performing the modeling of the symmetrical electronic optical system using software package ANSYS/Emag. Electronic optical system modeling performance evaluation algorithm and technique was designed. After the modeling the performances of equation system solution methods were compared against each other.

Keywords. Finite element method, equation solvers, electronic optical system.

1. Introduction

The behaviour of a phenomenon in an engineering system depends upon the geometry or domain of the system, the property of the material or medium, and the boundary, initial and loading conditions. For an engineering system, the geometry or domain can be very complex. Further, the boundary and initial conditions can also be complicated. It is therefore, in general, very difficult to solve the governing differential equation via analytical means. In practice, most of the problems are solved using numerical methods. Among these, the methods of domain discretization championed by the finite element method are the most popular, due to its practicality and versatility.

The finite element method leads to a system of equations that must be solved simultaneously [1,2]. A complex model can generate a very large system of equations. Each equation represents an unknown quantity which is required to find. Each unknown quantity is also referred to as a degree of freedom. Solving a large system of simultaneous equations requires a long time and large computer resources.

Software package ANSYS (ANSYS Inc., www.ansys.com) has leading solver technology to support robust and comprehensive simulation capability. Two types of direct solvers (Sparse Direct Solver, Frontal (Wavefront) Solver) are implemented in it and also three iterative solvers which are based on the conjugate gradient method (Jacobi Conjugate Gradient solver, Preconditioned Conjugate Gradient solver, Cholesky Conjugate Gradient solver) [3–6].

2. Modeling algorithm

In order to determine the performance of mentioned solvers the spatial model of colour TV-tube symmetrical electronic optical system is used (Fig. 1) [7,8].
The decision to use this system was assumed because electronic optical system modeling technique is already highly elaborated [7-10].

Electronic optical system construction model consists of several separate electrodes. In this case, electronic optical system is composed of seven electrodes – cathode, modulator, first accelerating electrode, first focusing electrode, second accelerating electrode, second focusing electrode and anode [7,8]. Such electronic optical system is shown in Fig. 1. Model is created only for side presumable electron trajectory.

Presumptible electron trajectory zones were selected in order to reduce the calculation span. It is obvious, that electron trajectories exist only in electrode aperture regions. Thus presumptive electron trajectory zones (central and side) are approximated using eight connected cylinders (Fig. 1) [7,8].

Completed finite element mesh is shown in Fig. 2. In order to determine the performance (solution duration) of solvers the electronic optical system model containing 394288 tetrahedral finite elements SOLID123 (546352 nodes) was used.

EOS modeling consists of four main stages: construction model creation, finite element mesh generation (tetrahedral finite element SOLID123 is used), setting of electrode potentials and solution of equation system (Fig. 3). Differences of this algorithm compared to technique presented in [7,8] are the following:

1) algorithm permits the analysis of influence of solver type and finite element mesh density (finite element number) on the modeling performance,
2) there are no limitations when selecting the required solver;
3) there are no limitations when selecting the size of finite elements (tetrahedral finite element SOLID123 is used) in all areas of electronic optical system (in eight trajectory cylinders and in remaining space).
4) only values of electric potential at the nodes of finite element mesh are calculated (values of electric potential are not calculated at the selected points of trajectory cylinders).

The performance of Sparse Direct, Distributed Sparse Direct, Jacobi Conjugate Gradient, Preconditioned Conjugate Gradient, Cholesky Conjugate Gradient solvers is analyzed during modeling.

3. System of linear equations

The system of linear equations generated by the finite element technique is solved using a direct elimination or an iterative method [3]. A direct elimination method is based on a Gaussian elimination method, according to which the values of the unknown vector of variables \( \{u\} \) are found in the following equation

\[
[K]\{u\} = \{F\},
\]  

(1)
where: \( K \) – conductivity matrix; \( \{ u \} \) – vector of nodal unknown values; \( \{ F \} \) – applied load vector.

The direct elimination method [3] consists of decomposition of the matrix \( K \) into lower and upper triangular matrixes \( [L][U] \). Then forward and backward substitutions using matrixes \( L \) and \( U \) are made in order to compute the vector \( \{ u \} \), containing the values of solution results.

In case of iterative method [3] the initial guess is made for the value \( u_1 \) of the solution vector \( \{ u \} \) and then a iterative steps are performed resulting in a sequence of vectors \( u_2, u_3, ..., u_n \), such that, in the limit, \( u_n = u \) as \( n \) tends to infinity. During the calculation of vector \( u_n \) matrix \( K \), vectors \( \{ F \} \) and \( \{ u \} \) are used from one of the previous iterations. Typically the solution converges to within a specified tolerance after a finite number of iterations. Further we will present a short overview of these methods [3].

4. Sparse Direct Solver

The linear matrix equation (1) is solved by triangular decomposition of matrix \( K \) to yield the following equation:

\[
[L][U]\{u\} = \{F\} \tag{2}
\]


By substituting

\[
\{ w \} = [U]\{ u \} \tag{3}
\]

it is possible to obtain \( \{ u \} \) by first solving the triangular matrix system for \( \{ w \} \) by using the operation of direct insertion of (3) into the (2) equation

\[
[L]\{ w \} = \{ F \} \tag{4}
\]

and then vector of unknown quantities \( \{ u \} \) is calculated using the back substitution operation on a triangular matrix given by:

\[
[U]\{ u \} = \{ w \} \tag{5}
\]

When the matrix \( K \) is symmetric, the description given above could use the substitution:

\[
[K] = [L][L]^T \tag{6}
\]

However, it is modified as:

\[
[K] = [L][D][L]^T \tag{7}
\]

where: \( D \) – a diagonal matrix.

The diagonal terms of matrix \( D \) may be negative in the case of some nonlinear finite element analysis. This allows the generation of matrix \( [L] \) without the consideration of a square root of negative number. Thus, equations (2–5) become:

\[
[L][D][L]^T \{ u \} = \{ F \} \tag{8}
\]

\[
\{ w \} = [D][L]^T \{ u \} \tag{9}
\]

\[
[L][w] = \{ F \} \tag{10}
\]

\[
[D][L]^T \{ u \} = \{ F \} \tag{11}
\]

Since the matrix \( K \) is normally sparsely populated with coefficients dominantly located around the main diagonal, the Sparse Direct Solver is intended to handle only the non-zero entries in the matrix \( K \). In general, during the Cholesky decomposition of the matrix \( K \) shown in equation (2) or (8), non-zero coefficients appear in matrixes \( [L] \) or \( [L] \) at coefficient locations where \( K \) matrix had zero entries. By using the Sparse Direct Solver method this fill-in of the matrix is minimized by reordering the equation numbers in the matrix \( K \). This procedure is extensively described in [3].

5. Frontal Solver

The computer time required in order to solve the problem is proportional to the square of the mean wavefront magnitude. The wavefront magnitude is determined by the order in which the elements are arranged. The node numbers of all elements are verified in order to determine which element is the last to use for each node. Since the overall system of equations is formed from the element matrixes, the equations for a
node which occurs for the last time are algebraically solved in terms of the remaining unknown values and eliminated from the formed matrix by using the Gauss elimination technique. The active equations are represented by:

$$\sum_{j=1}^{L} \mathbf{K}_{kj} \mathbf{u}_j = \mathbf{F}_k,$$  \hspace{1cm} (12)

where: \( \mathbf{K}_{kj} \) – stiffness coefficient relating the force at the degree of freedom \( k \) to the displacement at the degree of freedom \( j \); \( \mathbf{u}_j \) – nodal displacement of the degree of freedom \( j \); \( \mathbf{F}_k \) – nodal force of the degree of freedom \( k \); \( k \) – row number; \( j \) – column number; \( L \) – number of equations (rows).

To eliminate an equation, for which \( i = k \), the equation is normalized in the following way:

$$\sum_{j=1}^{L} \frac{K_{ij}}{K_{ii}} \mathbf{u}_j = \frac{F_i}{K_{ii}}.$$  \hspace{1cm} (13)

Equation (13) can be written as:

$$\sum_{j=1}^{L} K^*_j \mathbf{u}_j = \mathbf{F}_i^*,$$  \hspace{1cm} (14)

where: \( K^*_j = \frac{K_{ij}}{K_{ii}} \), \( F^*_i = \frac{F_i}{K_{ii}} \), \( K_{ii} \) – is known as the “pivot” [3]. If the absolute value of \( K_{ii} \) is sufficiently small, it is mathematically equal to zero. This usually means the structure is insufficiently constrained.

Equation (14) is written to a file for later back-substitution. The remaining equations are modified as:

$$K^*_j = K_{jj} - K_{kj}K^*_k,$$  \hspace{1cm} (15)

$$F^*_k = F_k - K_{kj}F^*_j,$$  \hspace{1cm} (16)

where: \( k \neq i \) so that

$$\sum_{j=1}^{L-1} K^*_j \mathbf{u}_j = \mathbf{F}_k^*,$$  \hspace{1cm} (17)

where \( k \) varies from 1 to \( L-1 \). When the row \( j \) is eliminated from the equation (17), this procedure is repeated for all other equations suitable for elimination.

The equations for a node which occurs for the first time are added to the formed matrix over the course of the solution process. So, the formed matrix expands and contracts as node make their first and last appearance in the element definitions. The varying size of the active matrix is considered as the momentary wavefront size.

When several elements are connected to the same node, the degrees of freedom related to these elements remain active until the wavefront “passes” all elements connected to the node. Degrees of freedom related by constraint equations or coupled nodes remain active until the wavefront “passes” all elements connected to the related degrees of freedom. Master degrees of freedom remain active in memory and are not deleted from the wavefront. This procedure is extensively described in [3]. The ideal model size when using Frontal Solver is under fifty thousand degrees of freedom, therefore it is not suitable for three-dimensional modeling of electronic optical system.

6. Iterative Solvers

The iterative solvers are based on the conjugate gradient method. In this case the system of linear equations (1) is also solved. In the conjugate gradient method, the solution is found as a series of vectors \( \{\mathbf{p}_i\} \):

$$\{\mathbf{u}\} = \alpha_1 \{\mathbf{p}_1\} + \alpha_2 \{\mathbf{p}_2\} + \ldots + \alpha_m \{\mathbf{p}_m\},$$  \hspace{1cm} (18)

where \( m \) is no larger than the matrix size \( n \).

The rate of convergence of the conjugate gradient algorithm is proportional to the square root of the conditioning number of the matrix \( [K] \) where the condition number of matrix \( [K] \) is equal to the ratio of the maximum eigenvalue of \( [K] \) to the minimum eigenvalue of \( [K] \). A preconditioning procedure is used to reduce the condition number of linear equations (1). In the Jacobi Conjugate Gradient algorithm, the diagonal elements of the matrix \( [K] \) are used as the preconditioner matrix \( [Q] \), while in the Cholesky Conjugate Gradient and Preconditioned Conjugate Gradient algorithms, a more sophisticated preconditioner matrix \( [Q] \) is used.
Convergence is achieved when:

\[ \left\{ R_i \right\}^T \left\{ R_i \right\} \leq \varepsilon^2, \quad (19) \]

where: \( \varepsilon \) – user supplied tolerance; \( \left\{ R_i \right\} = \left\{ F \right\} - \left\{ K [u_i] \right\}; \left\{ u_i \right\} – solution vector at iteration \( i \). It is assumed that the initial starting vector \( \left\{ u_0 \right\} \) is a zero vector. This procedure is extensively described in [3].

7. Analysis of solver performance

Solver performance was analyzed using two personal computers with different configuration. Configuration of the first computer: mainboard – Biostar M7VIT Pro; CPU – AMD Athlon XP 2500+ (1.83 GHz); RAM – Corsair DDR-400 3xCMX512-3200 1.5 GB (latency 3-3-3-8); HDD – Western Digital WD600JB.

Configuration of the second computer: mainboard – ASUS P5B Deluxe; CPU – Intel Core 2 Duo E6600 (2.40 GHz); RAM – Corsair DDR2-800 Twin2x2048-6400C4 2.0 GB (latency 4-4-4-12); HDD – Western Digital WD5000KS. Windows XP Professional (Win32 x86) 5.01.2600 (Service Pack 2) operational system was used in both cases.

![Figure 3. Solver performance](image-url)

Solver performance was analyzed in two different cases. The received results are presented in Fig. 3 and Fig. 4.

In the first case the number of finite element mesh nodes remains constant, but different solvers are selected (Fig. 3). Finite element mesh generated in the area of electronic optical system consists of 546352 nodes.

In the second case the Sparse Direct Solver is selected and the number of finite element mesh nodes is varied, i.e. it is made denser or sparser (Fig. 4). In this case the number of finite element mesh nodes is gradually changed from 158322 to 546352.

After analyzing solver performance it was found, that it is most purposeful to use Preconditioned Conjugate Gradient solver to perform the spatial modeling of electronic optical system.

When using two computers with earlier specified configuration, it was determined that Preconditioned Conjugate Gradient solver is approximately six times faster than Sparse Direct Solver, approximately from 3% to 11% faster than Jacobi Conjugate Gradient and...
approximately from 8% to 10% Cholesky Conjugate Gradient solvers. Using any of the mentioned solvers, the error of the modeling results does not exceed the defined limit [7,8].

The defined tolerance has little impact on the performance of Jacobi Conjugate Gradient, Cholesky Conjugate Gradient and Preconditioned Conjugate Gradient solvers. The difference of performances when tolerances 10^{-6} and 10^{-10} are set when using Jacobi Conjugate Gradient solver are 19%, Cholesky Conjugate Gradient solver – 11%, Preconditioned Conjugate Gradient solver – 12%.

During analysis of finite element mesh density influence on the modeling performance, it was determined that the calculation duration exponentially increases with increase of number of finite element mesh nodes. Results of this investigation could be used when selecting the optimal number of finite element mesh nodes.

8. Conclusions

Longest time to solve the system of equations is taken when using Sparse Direct and Distributed Sparse Direct solvers (approximately six times longer than using iterative solvers).

Preconditioned Conjugate Gradient solver is approximately 7% faster than Jacobi Conjugate Gradient and approximately 9% Cholesky Conjugate Gradient solvers. In this case the selection of solver is mainly determined the amount of available memory.

Sparse Direct Solver is most effective to use when the number of degrees of freedom is approximately half million, and for Jacobi Conjugate Gradient, Preconditioned Conjugate Gradient, Cholesky Conjugate Gradient solvers this number is approximately ten million degrees of freedom.

The modeling duration is related with the number of finite element mesh nodes by exponential law.

9. References