A Matrix-Free Iterative Solution Procedure for Finite Element Problems

J. P. A. Bastos[1], R. T. Jacobs[2], N. Sadowski[1], and A. Kost[3]
[1]Universidade Federal de Santa Catarina, GRUCAD, Florianópolis, SC 88040-900, Brasil
[2]Technische Universität Dresden, 01062 Dresden, Germany
[3]Brandenburgische Technische Universität Cottbus, 03013 Cottbus, Germany
jpab@grucad.ufsc.br

***Abstract*—Recently, a procedure named N-Scheme has been introduced which enables the solution of finite element problems without explicitly assembling the system matrix when a stationary iterative solver is used. In this article, the matrix-free N-Scheme technique is generalised to the solution of finite element problems with non-stationary iterative solvers, which, compared to a stationary solver, enable a significant reduction of the required computation time.**

I. Introduction

The finite element procedure leads to a system of linear equations which is commonly denoted by $\mathbf{A}\cdot\mathbf{x}=\mathbf{b}$, where $\mathbf{A}$ represents the system matrix, $\mathbf{b}$ the source vector, and $\mathbf{x}$ the coefficient vector to be determined. Recently, a technique called N-Scheme has been introduced, which enables the solution of finite element problems without the explicit assembly of the system matrix [1]–[3]. This scheme is applicable in a stationary solver, such as Gauss-Seidel or the Successive Over-Relaxation (SOR) method, and rests on the construction of an array which provides the connectivity of a node to the elements that are associated with this particular node. The method delivers reliable results but requires noticeable more computation time in comparison to conventional finite element implementations. The required CPU time can be reduced by precomputing the element matrices [1], but this would then require more memory than actually assembling the system matrix which consequently is not a favourable solution. A significant improvement without increasing the memory requirement has been achieved by modifying the N-Scheme technique for the employment in a non-stationary iterative solver, the description of which represents the aim of this paper.

II. N-Scheme applied in non-stationary solvers

A slight modification of the N-Scheme enables its application in non-stationary solvers such as the Conjugate Gradient method (CG), the BiConjugate Gradient method (BiCG) and the stabilized version (BiCGstab), the Quasi-Minimal Residual method (QMR), or the Generalized Minimal Residual method (GMRES). Detailed algorithmic descriptions of these solvers are provided in [4]–[6]. In all the Krylov subspace methods mentioned, the system matrix is $\mathbf{A}$ is only referenced in the context of a matrix-vector product of the form

$$\mathbf{q}^{(k)}=\mathbf{A}\cdot\mathbf{p}^{(k)} \tag{1}$$

or of its transposed variant, where $\mathbf{p}$ represents a known vector whose product with $\mathbf{A}$ at iteration k is assigned to a vector $\mathbf{q}$. These vectors are of size n and are commonly referred to as search and update vectors, where n denotes the number of degrees of freedom in the system. The N-Scheme enables the matrix-free evaluation of (1) by performing the product on an element-by-element basis. This is achieved by first computing the local element matrix of a particular element, by then evaluating the product of this matrix with the relevant entries of $\mathbf{p}$, and by subsequently adding the result of this product to the relevant entries of $\mathbf{q}$. The computation of (1) is complete when all elements have been considered. This approach essentially requires more CPU time compared to the conventional assembling strategy, but it circumvents the explicit formation of $\mathbf{A}$ and, as a result of that, requires virtually no memory. An efficient implementation for the computation of the local element matrices is, of course, indispensable. The convergence behaviour of a Krylov subspace solver can be improved by employing an appropriate preconditioner, but the application of a preconditioner without violating the basic principle of a matrix-free computation, i.e. without explicitly computing and storing a preconditioner matrix, represents an intricate task. Good results have been achieved by employing a Jacobi preconditioner which is determined by the elements of the main diagonal of the system matrix. These n elements can be computed rapidly when needed or, as their storage would only require the same amount of memory as the search and update vectors, their assignment to an auxiliary vector can maybe be considered as admissible.

III. Examples and Initial Results

In order to validate the matrix-free procedure and to provide some initial timing results, a two-dimensional electrostatic problem and a three-dimensional magnetostatic problem are considered. The geometry of a shielded microstrip line with a substrate of permittivity ε is displayed in Fig. 1, where the actual problem domain has been halvened by exploiting the symmetry of the structure. The computation of the electric scalar potential $\phi_e(\vec{r})$ requires the solution of the Laplace equation

$$\nabla^2\phi_e(\vec{r})=0\,, \tag{2}$$

subject to the boundary conditions indicated in the figure, where ϕ_{e0} denotes the potential prescribed on the stripline. The domain is discretized with triangular elements. Table I shows number of elements n_{el}, the number of degrees of freedom n, the number of iterations n_{it} and the CPU time in seconds required to solve the resulting system of linear

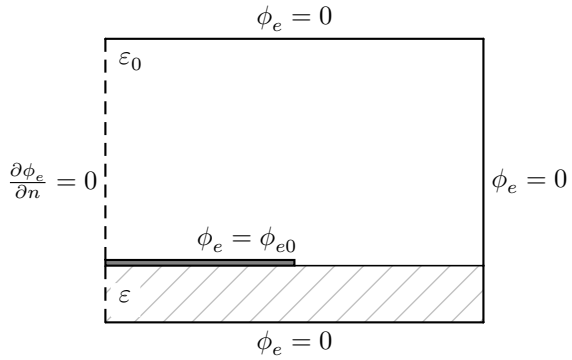

Fig. 1. Shielded microstrip line

equations with a tolerance of 10^{-5} by employing the non-preconditioned CG method with and without conventionally assembled system matrix, as well as the matrix-free SOR method. A relaxation factor of 1.95 has proven to be a good choice for the SOR method, but the CPU times reflect that the procedure is not as efficient as the CG algorithm. For

TABLE I
CPU TIME, NON-PRECONDITIONED CG AND SOR METHOD

		convent.		matrix-free			
		CG		CG		SOR	
n_{el}	n	n_{it}	t/s	n_{it}	t/s	n_{it}	t/s
6055	2877	124	0.05	124	0.29	208	1.03
13947	6748	179	0.18	179	0.98	203	2.43
57053	28075	360	2.87	358	8.58	203	12.85
231039	114616	669	24.92	666	74.36	792	241.88

the highly refined discretization in this example, the matrix-free CG solution requires about three times as long as the conventional procedure, but virtually without needing memory. This can potentially be an advantage for the simulation of large practical problems. The convergence behaviour of a Krylov subspace solver can be significantly improved by employing an appropriate preconditioner, but this is not an easy task in a matrix-free environment, as mentioned in section II. Table II displays the CPU time the CG, the BiCGstab, and the QMR method require to solve the microstrip problem to the same accuracy as before, but with the application of a Jacobi preconditioner. The number of iterations required by

TABLE II
CPU TIME, MATRIX-FREE JACOBI-PRECONDITIONED METHOD

	CG		BiCGstab		QMR	
n	n_{it}	t/s	n_{it}	t/s	n_{it}	t/s
2877	89	0.22	57	0.28	81	0.42
6748	134	0.75	81	0.94	117	1.47
28075	257	6.52	150	7.82	207	11.63
114616	487	55.67	283	68.09	363	104.70

the Jacobi-preconditioned CG solver reduces to about 70 % of the non-preconditioned version, whereas the BiCGstab and QMR procedure primarily confirm the applicability of the N-Scheme technique to more advance solvers which enable the solution of non-symmetric systems. The CPU times displayed are of course machine and implementation dependent, but their relation to the conventional CG solution shown in Table I provides a good first indication of the performance of the matrix-free algorithms. The second problem considered is a magnetic circuit consisting of a coil and an iron core, as shown in figure 2. The magnetic field in the circuit is modelled using

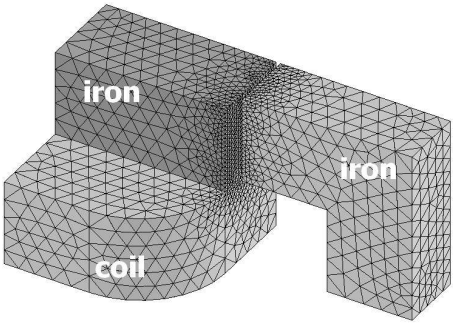

Fig. 2. Magnetic circuit with coil and iron core

the Source-Field technique [7], [8], for which

$$\nabla \cdot \left[\mu(\vec{r}\,)\left(\vec{H}_s(\vec{r}\,) - \nabla\phi_m(\vec{r}\,)\right)\right] = 0 \qquad (3)$$

is solved, where $\vec{H}_s(\vec{r}\,)$ denotes the field generated by a source current density and $\phi_m(\vec{r}\,)$ the magnetic scalar potential. The domain is discretized with tetrahedral elements and the computations confirm the basic CPU time relations between the matrix-free CG solver and matrix-free SOR method of the first problem considered.

IV. DISCUSSION AND OUTLOOK

The N-Scheme technique introduced in [1]–[3] has been modified and applied in non-stationary iterative solvers. Timing results of an electrostatic test problem have provided a good first indication of the performance of the matrix-free algorithms. The parallelisation of the matrix-free solvers as well as the computation of non-linear problems are currently under investigation and will be reported in due course.

REFERENCES

[1] J. P. A. Bastos, "Is it possible to solve a FEM static case without assembling and storing an $Ax = b$ matrix system?," *International Compumag Society Newsletter*, vol. 16, no. 1, pp. 4–10, 2009.

[2] J. P. A. Bastos and N. Sadowski, "A method to solve FEM statics cases without assembling a matrix system: application to 3D edge elements," in *Proceedings of the 8th International Symposium on Electric and Magnetic Fields, EMF 2009*, Mondovi, Italy, May 2009, pp. 157–158.

[3] J. P. A. Bastos and N. Sadowski, "A new method to solve 3-D magnetodynamic problems without assembling an $Ax = b$ system," *IEEE Transactions on Magnetics*, vol. 46, no. 8, pp. 3365–3368, August 2010.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, London, 3 edition, 1996.

[5] Anne Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[6] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[7] Y. Le Ménach, S. Clénet, and F. Piriou, "Determination and utilization of the source field in 3D magnetostatic problems," *IEEE Transactions on Magnetics*, vol. 34, no. 5, pp. 2509–2512, September 1998.

[8] P. Dular, F. Robert, J. F. Remacle, M. Umé, W. Legros, and U. Pahner, "Computation of the source current density in inductors of any shape using a mixed formulation," in *Proceedings of the 3rd International Workshop on Electric and Magnetic Fields, EMF'96*, 1996, pp. 107–112.