Speeding Up SSFEM Computation Using Kronecker Tensor Products

R Gaignaire, F Guyomarc'H, O Moreau, S Clénet, B Sudret

Speeding Up SSFEM Computation Using Kronecker Tensor Products

R. Gaignaire[1], F. Guyomarc'h[2], O. Moreau[3], S. Clénet[4], and B. Sudret[5]
[1]ACE, Montefiore Institute, Sart Tilman Campus, Grande Traverse, Liège B-4000, Belgiqum
[2]INRIA, Lille Nord-Europe, Villeneuve d'Ascq 59640, France
[3]Electricité de France, R&D Division, Clamart 92141, France
[4]L2EP, ENSAM, Lille 59046, France
[5]Electricité de France, R&D les Renardières, Moret-sur-Loing 77818, France

The spectral stochastic finite-element method makes it possible to convey some random aspects of input data to the output data. However, the system size dramatically increases with the number of input random variables. Using matrix Kronecker tensor products for system solving noticeably reduces the computation time and the storage requirements.

Index Terms—**Electrokinetics, finite-element method, hermite polynomial chaos, linear system resolution, random media.**

I. Introduction

DUE to aging, manufacturing process, or a lack of knowledge in state variables (pressure, temperature...), some uncertainties may appear in the classical input data of numerical electromagnetism modeling (such as material characteristics, loading, or geometrical dimensions). An approach to take these uncertainties into account consists in modeling input data as random variables and broadcasts this randomness to the output data. The methods, which we will deal with, are related to the finite-element method (FEM). The first method, a Monte Carlo simulation method (MCSM), builds a sample of R set of numerical values as input data and then solves the R-associated deterministic problems to obtain a sample of size R for the output data. Then, statistic treatment has to be applied to analyze the results. This method has remained as a reference for all of our work [1], [2]. Other methods, which have been proposed in mechanical engineering, are based on the so-called Hermite's polynomial chaos. One of them, called "nonintrusive" as an FEM code is used as a blackbox [3], will be mentioned in the last section. As in MCSM, numerous deterministic finite-element problems have to be solved. Another kind is called "intrusive" since it requires the finite-element code to be deeply modified, is the spectral stochastic finite-element method (SSFEM) [1], [2]. The underline of this method is to write input and output data as an expansion of Hermite chaos polynomials. Then, solving the random problem is equivalent to finding the deterministic values of an expansion in a Hilbertian base. Such resolution is performed by a Galerkin approach. Only one linear system has to be solved, meaning that the matrix coefficients are related to spatial and random meshes. This method features a general theoretical frame but unfortunately quickly leads to large systems. Standard numerical techniques lead to long computation time and huge random-access memory (RAM) storage requirements which, until now, prevent from tackling many industrial applications. The Kronecker tensor products method [4] has been tested to overcome this drawback on an actual electrokinetic case. First, the mathematical framework and discretization schemes applied to obtain the SSFEM discrete equations will be reminded. Second, the Kronecker tensor product approach will be detailed from the governing equations. The validations have been carried out by comparing SSFEM results involving conjugate gradient (CG) with either a standard stiffness matrix assembly or a Kronecker product technique. Finally, a discussion about numerical considerations will follow. To illustrate the efficiency of this approach, global current flux will be computed on an industrial case of a power-line joint using the tensor approach SSFEM and a nonintrusive method [3].

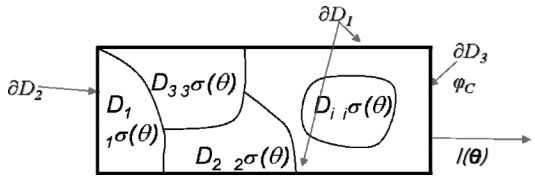

Fig. 1. Domain D divided in the M subdomain D_i.

II. Global Framework

A. Problem Definition and Notations

Let us consider a spatial domain D which represents the geometry of the device (Fig. 1) and Θ as the random domain. The boundary of D is divided into three complementary parts: 1) ∂D_1; 2) ∂D_2; and 3) ∂D_3. x and θ will denote the spatial and random dependence and $\boldsymbol{n}$ is an outward normal vector to D. $\mathbf{grad}_{\boldsymbol{x}}(.)$, $\mathrm{div}_x(.)$, and $\mathbf{curl}_{\boldsymbol{x}}(.)$ stand for the standard gradient, divergence, and curl operator through the spatial dimension. The random conductivity $\sigma(x,\theta)$ represents input data whereas the current density $J(x,\theta)$, the electrical field $E(x,\theta)$, and the global current through the surface $\partial D_1 I(\theta)$ are the output data.

D is assumed to be divided into M subdomains D_i, where each conductivity ${}_i\sigma(\theta)$ is supposed to be a uniform and independent random variable with finite variance. Let us denote $1_{Di}(x)$ as the indicator function of the subdomain D_i which is equal to 1 if x belongs to D_i and 0 elsewhere. Conductivity may then be written as

$$\sigma(x,\theta) = \sum_{i=1}^{M} {}_i\sigma(\theta) 1_{D_i}(x). \tag{1}$$

Manuscript received October 07, 2008. Current version published February 19, 2009. Corresponding author: O. Moreau (e-mail: Olivier.moreau@edf.fr).
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/TMAG.2009.2012662

The constitutive law linking the current density to the electrical field through the conductivity is written as

$$J(x,\theta) = \sigma(x,\theta)\boldsymbol{E}(x,\theta). \tag{2}$$

On ∂D_3, the normal component of J is supposed to be zero, whereas ∂D_2 and ∂D_3 are supposed to be equipotential surfaces, respectively, imposed to 0 and φ_C. Let us introduce a function $\alpha(x)$ such as

$$\alpha(x) = \begin{cases} 0, & \text{on } \partial D_2 \\ 1, & \text{on } \partial D_3. \end{cases} \tag{3}$$

Since the electrical field is curl free, a scalar potential exists, denoted by $\varphi(x,\theta)$ so that

$$E(x,\theta) = \mathbf{grad}_x\left(\varphi(x,\theta) + \varphi_C.\alpha(x)\right).$$

The resulting electrokinetics problem can then be written as

$$\begin{cases} div_x\left(J(x,\theta)\right) = 0 \text{ on } D \\ \boldsymbol{E}(x,\theta) = \mathbf{grad}_x\left(\varphi(x,\theta) + \varphi_C\alpha(x)\right), \text{ on } D \\ J(x,\theta) = \sigma(x,\theta)\boldsymbol{E}(x,\theta) \\ J(x,\theta).\boldsymbol{n} = 0 \text{ on } \partial D_1 \otimes \Theta \\ \varphi(x,\theta) = 0 \text{ on } \partial D_2 \otimes \Theta \\ \varphi(x,\theta) = \varphi_C \text{ on } \partial D_3 \otimes \Theta. \end{cases} \tag{4}$$

B. Discretization Scheme SSFEM

To numerically solve (4), a spatial mesh is needed. Let us consider a finite-element mesh. In the deterministic case, the scalar potential is spanned in the space of the nodal function related to the node i $\lambda_i(x)$. We assume that there are n spatial unknowns (the scalar potential) among the number N of nodes. They are the degrees of freedom related to the spatial dimension (SDoF).

A random mesh is also necessary to characterize those n unknown random variables ${}_i\varphi(\theta)$. Since there are M different independent random variables ${}_i\sigma(\theta)$ as input, an M-multidimensional Hermite polynomial will be used [1]–[3]. Let us denote ${}^g\Psi(\xi(\theta))$ as the gth multidimensional Hermite polynomial with variable ξ which is a random normal vector of size M. We will use the so-called polynomial chaos of M dimensions and of order p, which is the subspace of random variables with the finite variance spanned by the M dimension Hermite polynomials with order up to p. This space consists of dimension $P = C^p_{M+p}$. Since input random variables are of finite variance and independent, we may expand them in the space of dimension P_{in}

$$\sigma(x,\theta) \approx \sum_{i=1}^{M}\sum_{j=1}^{P_{\text{in}}} {}^j_i\sigma^j\Psi\left(\left(\xi_k(\theta)\right)_{k=1}^{M}\right)1_{D_i}(x). \tag{5}$$

The scalar potential may now be expanded in the spatial (SDoF) and random dimension with P_{out} degrees of freedom (RDoF) as

$$\varphi(x,\theta) + \varphi_C\alpha(x) = \sum_{i=1}^{n}\sum_{j=1}^{P_{\text{out}}} {}^j_i\varphi^j\Psi\left(\left(\xi_k\right)_1^M\right)\lambda_i(x) + \varphi_C\alpha(x) \tag{6}$$

where ${}^j_i\varphi$ will be the scalar unknowns of our problem.

C. Linear System

Whereas in the deterministic case a node is related to one unknown, in our case, a node is related to P_{out} unknowns which correspond to the random discretization. Let us store in a list of P_{in} matrices E^j of size P^2_{out} the mathematical expectation $E(.)$ of the product of the j, g, and m Hermite polynomials

$$E^j(g,m) = E\left({}^g\Psi\left(\xi(\theta)\right)^j\Psi\left(\xi(\theta)\right)^m\Psi\left(\xi(\theta)\right)\right). \tag{7}$$

Using the weak formulation of (4) from the Galerkin method and after simple algebra, we have to find ${}^j_i\varphi$ so that [1]

$$\begin{aligned}\sum_{l=1}^{n}\sum_{m=1}^{P_{\text{out}}} {}^m_l\varphi\sum_{i=1}^{M}\sum_{j=1}^{P_{\text{in}}}\Big\{ {}^j_i\sigma E^j(g,m) \\ \times\int_{D_i}\mathbf{grad}_x\left(\lambda_l(x)\right)\mathbf{grad}_x\left(\lambda_f(x)\right)dD\Big\} \\ = -\varphi_C\sum_{i=1}^{M}\sum_{j=1}^{P_{\text{in}}}\Big\{ {}^j_i\sigma E^j(g,1)\int_{D_i}\mathbf{grad}_x\left(\alpha(x)\right)\mathbf{grad}_x\left(\lambda_f(x)\right)dD\Big\}.\end{aligned} \tag{8}$$

From a continuous problem, a discrete problem of nP_{out} equations with nP_{out} unknowns has been defined—it is a linear system of the shape $A\varphi = B$. A is the stiffness matrix, φ is the vector of the nP_{out} unknowns, and B is the load vector which contains only boundary conditions in our case. Building the stiffness matrix and solving the linear problem with an ICCG algorithm will be denoted as method A1.

III. SSFEM Kronecker Product Approach

A. Computational Issue

Let us consider a spatial domain with a mesh leading to 6949 spatial scalar unknowns where three subdomains have random conductivities. The unknown scalar potential will be searched in a polynomial chaos of three dimensions with an order of less than 6, that leads to a problem with 84 RDoF related to the random dimension. Then, the total number of unknowns is $6,949 * 84 = 583\,717$. We can overestimate the number of nonzero terms (nnz) in the following way. In a 3-D tetrahedral mesh, each node is connected to about 30 other nodes. Then, on each line of the stiffness matrix, we should have about $\text{CN} * \text{RDOF} = 30 * 84$ nonzero terms (CN: connectivity). The RAM storage requirement will be about 10.9GO. This prevents tackling many industrial applications. The Kronecker product approach [4] enables avoiding the assembly of the whole stiffness matrix.

B. Mathematical Framework

The Kronecker product of a matrix H (n lines, m columns) with a matrix K (p lines, m columns) gives a matrix of C with $n \times p$ lines and $m \times p$ columns so that

$$C = H \otimes K = \begin{bmatrix} H_{11}K & \cdots & H_{1m}K \\ \vdots & \ddots & \vdots \\ H_{n1}K & \cdots & H_{nm}K \end{bmatrix}. \tag{9}$$

Let us now define a list of M deterministic matrices $(A_i)_{1\le i\le M}$ related to each subdomain D_i of D so that

$$A_i(l,f) = \int_{D_i}\mathbf{grad}_x\left(\lambda_l(x)\right)\mathbf{grad}_x\left(\lambda_f(x)\right)dD. \tag{10}$$

Each matrix A_i has less than SDoF $*$ CN nonzero terms. From this list of matrices, we may define another list of P_{in} matrices $(B^j)_{1\leq j\leq P\text{in}}$ which mix spatial discretization and the value of conductivity expansion

$$B^j = \sum_{i=1}^{M} {}_i^j\sigma A_i. \tag{11}$$

Matrix B^j is related to the expansion of the conductivity on the jth Hermite polynomial through each domain. Each matrix B^j has less than SDoF $*$ CN nonzero terms. Moreover, as the conductivities are modeled by independent random variables, some j exists such that for all i, ${}_i^j\sigma$ is equal to zero. For such j, the matrix B^j is empty.

Let us consider a line li defined by the index f and g so that $li = (g-1)n + f$ and a column co are defined by the index l and m so that $co = (m-1)n + l$. Simple algebra from (8) shows that A may be written as

$$A(li, co) = \sum_{j=1}^{P_{\text{in}}} E^j(m,g) \sum_{i=1}^{M} {}_i^j\sigma \times \int_{D_i} \mathbf{grad}_x\left(\lambda_l(x)\right)\mathbf{grad}_x\left(\lambda_f(x)\right) dD. \tag{12}$$

By using (11), (12) and the definition (9), one can notice A may be written as a sum of the Kronecker product

$$A = \sum_{j=1}^{P_{in}} E^j \otimes B^j. \tag{13}$$

C. CG Involving Kronecker Products

In the CG algorithm (like in most of iterative solvers), assembling the stiffness matrix is not necessary: we just need to know how to compute a matrix vector product. Taking into account that the operator $\text{vect}(.)$ converts an $n \times m$ matrix into a transposed vector of $n * m$ lines and considering a matrix X with n lines and P_{out} columns, the Kronecker product features the following interesting property:

$$\left(\sum_{j=1}^{P_{\text{in}}} E^j \otimes B^j\right) \text{vect}(X) = \sum_{j=1}^{P_{\text{in}}} \text{vect}(B^j X E^j). \tag{14}$$

By using (14), P_{in} matrix computations are required, which consist of a first product of a matrix B^j (order n^2, $n * CN$ nonzero terms) by a matrix X of size $n * P_{\text{out}}$, and a second one involving the resulting matrix by a matrix E^j of order P_{out}^2. Some optimized algorithms have been developed to perform this type of operation, owing to special structures of B^j and E^j.

D. Advantages and Difficulties of the Kronecker Product

By using the Kronecker product approach, we just have to store the list of P_{in} matrices B^j and E^j. Then, the total storage is of the order $P_{\text{in}}(P_{\text{out}}^2 + n * CN)$, whereas the sequential approach is needed to store $n*CN*P_{\text{out}}^2$ terms. By considering the same example than in Section III-A, one can show that we just need 0.7GO of RAM storage by using the Kronecker approach. In fact, the gain by using the Kronecker product is obvious (due to $P_{in} * P_{\text{out}}^2$ being negligible compared to $n*CN*P_{\text{in}}$). In addition, tests carried out with the 1-D matrix vector by using the Kronecker approach have been quicker than the direct stiffness matrix vector product.

The main issue is the preconditioning method for the CG algorithm. As far as the standard CG algorithm is concerned, it is quite simple to use Cholesky preconditioners (method A1). It becomes unfortunately not so obvious when the stiffness matrix is not assembled. As a fist step, this paper only deals with a Jacobi preconditioner for the CG involving tensor products (method A2).

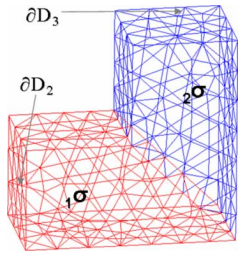

Fig. 2. L-shape academic case.

TABLE I
RANDOM PROPERTIES OF THE CONDUCTIVITY, L SHAPE

Conductivity	Mean	Standard Deviation
${}_1\sigma$~Ln(5.19,0.22)	200	100
${}_1\sigma$~Ln(3.84,0.15)	50	20

IV. Validation and Numerical Considerations

A. Academic Case

Let us consider a mesh (Fig. 2) with 912 tetrahedrals, 276 nodes giving 234 scalar potential unknowns. The difference of potential between ∂D_2 and ∂D_3 is equal to 1. The domain is divided in two subdomains where the conductivity is supposed to be a random variable following the lognormal law whose means and standard deviations are given in Table I.

This problem has been solved with the three previous SSFEM algorithms for $p_{in} = 2 * p_{\text{out}}$ with p_{out} chosen to be equal to 4, 5, and 6. That leads to a polynomial chaos size (P_{out}) that is equal to 15, 21, and 28. Previously, the SSFEM classical algorithm (A1) had been validated compared with an MCSM [1], [2]. To validate the Kronecker approach, the value of each unknown coefficient ${}_i^j\varphi$, obtained by the three methods, has been compared. By denoting ${}_i^j\varphi_1$ (respectively, ${}_i^j\varphi_2$), the value obtained by A1 (respectively, A2) lets us define an error criterion (err) in percent by

$$\text{err} = \max_{(i,j)\in\{1,\ldots,n\}\otimes\{1,\ldots,P_{out}\}} \left(\frac{\left| {}_i^j\varphi_2 - {}_i^j\varphi_1 \right|}{\left| {}_i^j\varphi_1 \right|} * 100 \right). \tag{15}$$

As we can see in Table II, the difference between the methods is negligible: the maximum error on all of the meshes for the selected output order is $6.10^{-4}\%$.

Table III summarizes some numerical considerations: Unk is the number of unknowns involved in the problem, NbIter is the number of iterations needed by CG to converge, Tb represents the time to build the stiffness matrix for A1 and the time needed

TABLE II
ERROR CRITERION ON ALL OF THE MESHES, L-SHAPE $(10^{-3}\%)$

	$p_{out} = 4$	$p_{out} = 5$	$p_{out} = 6$
err	0.6	0.5	0.5

TABLE III
STANDARD AND TENSOR CG ALGORITHM CHARACTERISTICS

Method	Unk	NbIter	Tb (s)	TCG (s)	Rate	Total Time (s)
A1 $p_{out} = 4$	3510	30	5.06	0.31	0.061	5.37
A2 $p_{out} = 4$	3510	283	2.29	0.54	0.24	2.84
A1 $p_{out} = 5$	4914	30	13.35	0.78	0.058	14.13
A2 $p_{out} = 5$	4914	353	2.7	1.22	0.45	3.92
A1 $p_{out} = 6$	6552	30	31.90	1.67	0.052	33.57
A2 $p_{out} = 6$	6552	431	3.12	2.51	0.80	5.63

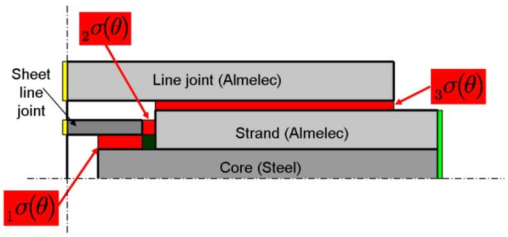

Fig. 3. Axisymmetrical cross section of half a line joint.

to build list B^j and E^j for A2, TCG is the time needed for CG convergence, the rate is the ratio between Tb and TCG, and the total time represents the complete solving time.

First, it can be noticed that the number of unknowns depends only on p_{out} since the spatial mesh is the same for all configurations. As far method A1 is concerned, NbIter remains constant with regard to p_{out} thanks to the Cholesky preconditioner efficiency. Moreover, the CPU time for CG convergence is negligible compared with the time needed to build the stiffness matrix ("rate" column). However, the CG CPU time increases with p_{out} due to the fact that the size of the stiffness matrix increases dramatically with p_{out} so that each matrix vector product involved by CG iterations become more and more CPU time consuming. So the time to build the stiffness matrix increases approximately with the square of p_{out}.

Concerning method A2, by using the Kronecker approach, it is possible to nearly get rid of the assembly time. However, the weak efficiency of the Jacobi preconditioner makes the number of iterations increase with p_{out}. Nevertheless, it turns out that A2 becomes more and more efficient as p_{out} increases (until 6 times faster than A1 for $p_{\text{out}} = 6$). This result points out that focusing on the preconditioning aspects dedicated to Kronecker products should make this technique even faster.

B. Industrial Case

Line joints are commonly used to connect high-power transmission liners made of strands around a steel core (Fig. 3). The study consists in computing the total current through the line joint, taking into account some uncertainties in three contact conductivities: $(_1\sigma, _2\sigma, _3\sigma)$ [2].

TABLE IV
RANDOM PROPERTIES OF THE CONDUCTIVITY LINE JOINT

Conductivity laws	mean	Standard deviation
$_1\sigma \sim U([500;10\,000])$	5,250	2,742
$_2\sigma \sim U([57;2\,270])$	1,163.5	638.8
$_3\sigma \sim U([1\,120;4\,770])$	2,945	1,053.8

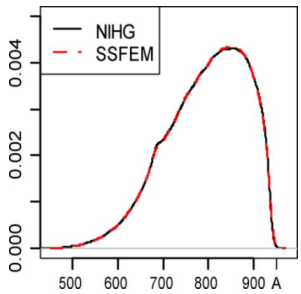

Fig. 4. Probabilistic distribution of the global current in the line joint for NIHG (-) and tensor product SSFEM (- -) methods.

According to the experts, the conductivities have been modelled as random variables with uniform laws (Table IV). The SSFEM problem has been solved thanks to the tensor technique whereas it was previously unfeasible to store the whole stiffness matrix (Section III-A).

The validation has been carried out by comparing the nonintrusive polynomial chaos method by using the Hermite–Gauss projection (NIHG) [3]. It is worth noticing that unlike SSFEM, some theoretical properties about random global quantities [2] are not yet available with NIHG. Nevertheless, as far as the line joint problem is concerned, very good agreement can be observed, with the same discretization parameters between the NIHG and SSFEM results (Fig. 4) with computation times of the same order.

V. CONCLUSION

The CG algorithm involving Kronecker tensor products makes the SSFEM computations run faster than standard implementations. In addition, this algorithm requires much less RAM storage, allowing to deal with some actual industrial studies. Preconditioners dedicated to this tensor technique, such as block-SSOR, should increase the performances even further, which is required to tackle eddy current problems.

REFERENCES

[1] R. Gaignaire, S. Clenet, O. Moreau, and B. Sudret, "3D spectral stochastic finite element method in electromagnetism," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1209–1212, Apr. 2007.

[2] R. Gaignaire, S. Clenet, O. Moreau, and B. Sudret, "Current calculation in electrokinetics using a spectral stochastic finite element method," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 754–757, Jun. 2008.

[3] M. Berveiller, B. Sudret, and M. Lemaire, "Structural reliability using a non intrusive stochastic finite element method," presented at the 12th IFIP Working Group 7.5 Working Conf. Reliability and Optimization of Structural Systems, Aalborg, Denmark, May 2005.

[4] H. G. Matthies and A. Keese, "Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations," *Comput. Methods Appl. Mechan. Eng.*, vol. 194, no. 12–16, pp. 1295–1331, Apr. 8, 2005, Special Issue on Computational Methods in Stochastic Mechanics and Reliability Analysis.