SIAM J. SCI. COMPUT.
Vol. 20, No. 2, pp. 619–647

NUMERICAL IMPLEMENTATION OF A SPACE-TRANSFORMATION APPROACH FOR SOLVING THE THREE-DIMENSIONAL FLOW EQUATION*

DIONISSIOS T. HRISTOPULOS†, GEORGE CHRISTAKOS†, AND MARC SERRE†

Abstract. This work is concerned with the numerical implementation of the space-transformation approach in porous media modeling. Space transformation is a mathematical approach that simplifies the study of multidimensional problems by reducing them to unidimensional ones. In particular, the implementation aspects considered in this work include (i) the development and testing of numerical space-transformation algorithms for three-dimensional, space-dependent functions that can be used to model flow within porous media, and (ii) the use of these algorithms to solve partial differential equations governing groundwater flow. We also establish the competitiveness of the space-transformation-based numerical procedures in terms of accuracy and speed. Further improvements in computational speed can be accomplished by parallel implementation of the space-transformation calculations. The space-transformation method is shown to provide a computationally efficient approach for obtaining three-dimensional flow solutions that honor the boundary conditions. This approach can be also used in stochastic flow modeling to generate multiple realizations of the random hydraulic head field.

Key words. space transformations, stochastic computational analysis, groundwater flow

AMS subject classifications. 86A05, 65C20, 65P05

PII. S1064827594278721

1. Introduction.

Scientific and engineering applications are often based on our ability to obtain accurate solutions of differential equations that model physical processes. In the case of physical processes that involve multidimensional media with spatially and/or temporally variable properties—which include most problems of interest in environmental sciences—the solution of the differential equations represents a major modeling challenge. Indeed, analytical and numerical solutions to multidimensional stochastic problems may be very difficult to obtain (Cushman (1990), NRC (1990)). Numerical approximation of multidimensional problems requires fine discretization in space and time to capture the variability associated with random media. This fine discretization can lead to overwhelming requirements for computational memory and speed. In heterogeneous systems, ill-conditioned matrices may cause iterative matrix solution methods to be prohibitively slow and expensive (e.g., Ababou et al., 1989). Computational effort can be reduced by simulating systems in lower dimensions. However, modeling natural systems in fewer than three dimensions introduces certain errors due to the dependence of the fluctuation effects on the dimensionality (e.g., Bakr et al. (1978), Dagan (1979), Tchelepi (1994), Tchelepi and Orr (1994)).

One approach for overcoming some of the pitfalls associated with numerical solutions in multidimensional media is provided by means of the space-transformation analysis. Generally, space transformations (ST) are mathematical operations that

*Received by the editors December 15, 1994; accepted for publication (in revised form) July 25, 1997; published electronically September 10, 1998. This work was supported by grant P42ES05948-02 from the National Institute of Environmental Health Sciences and the Computational Science Graduate Fellowship Program of the Office of Scientific Computing in the Department of Energy.
http://www.siam.org/journals//sisc/20-2/27872.html

†Department of Environmental Sciences and Engineering, CB#7400, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-7400 (dionisi@cmr.sph.unc.edu, george_christakos@unc.edu, marc_serre@unc.edu).

simplify the study of multidimensional problems by reducing them to unidimensional problems (Christakos (1984a, 1992); Christakos and Panagopoulos (1992)). Even if a closed-form solution cannot be obtained for the corresponding unidimensional problem, numerical solutions in one dimension are much faster and more accurate than for the original problem. Examples of successful applications of the ST concept in stochastic analysis include the development of criteria of permissibility for spatial correlation models (Christakos (1984b)) and the turning bands simulator (Tompson, Ababou, and Gelhar (1989)). In addition, there is great potential for using ST in the study of multidimensional partial differential equations with space- and/or time-dependent coefficients as described in Christakos and Hristopulos (1994, 1997). Here we focus on implementing the procedure outlined in these works for the solution of three-dimensional steady-state groundwater flow by developing the numerical tools for applying ST. This paper is structured as follows: In section 2 we present a brief review of the ST theory. In section 3 we discuss the implementation of one of the ST operators and its inverse. In section 4 we present a brief theoretical overview of the ST solution of the steady-state flow equation together with its numerical solutions.

2. A brief review of the theory of ST. In the analysis of subsurface processes, field parameters such as permeability and porosity are commonly treated as spatial random fields (e.g., Neuman (1982), Naff and Vecchia (1986)), the realizations of which are multidimensional functions from a statistical ensemble that appear with frequency determined by the multivariate probability distribution. Thus, it is possible to obtain estimates that account for the variability and the uncertainties inherent in most environmental processes. In general, STs are mathematical operations that map a process defined on a multidimensional space onto a space of reduced dimensionality. The ST operators can be applied to random field realizations, if the latter are integrable functions, or to the statistical moments of the random field (Christakos (1987, 1992)). Below we briefly review the definitions of ST operators and some of their properties.

Let R^n be the n-dimensional Euclidean space, $\mathbf{s} = (s_1, s_2, \ldots, s_n) \in R^n$ denote the position vector, and $\boldsymbol{\theta}$ denote a direction vector on the generalized unit sphere S^{n-1}. Let $f_n(\mathbf{s})$ be a function defined on R^n that is either rapidly decreasing or has a compact support. Then we can define a mapping T_n^1 from the space R^n into the space $R^1 \times S^{n-1}$ as follows:

$$T_n^1 : f_n(s) \to \int \mathbf{ds} f_n(s)\delta(p - \boldsymbol{s} \cdot \boldsymbol{\theta}), \tag{2.1}$$

where $p \in R^1$ and $\delta(\cdot)$ is the singular distribution known as the Dirac delta function. The image of the mapping for all the direction vectors on S^{n-1} and all real numbers p is known as the Radon transform (Helgason (1980), Deans (1993)) or as the plane wave integral (John (1955)). The T_n^1 transformation in a specific direction represents the projection of the n-dimensional function on the hyperplanes that are perpendicular to this direction. The mapping (2.1) is an ST expressed as the integral of the function $f_n(\mathbf{s})$ over the hyperplane that is perpendicular to the direction vector $\boldsymbol{\theta}$ and is located at a distance p from the origin (this ST is usually called an ST-1 in order to be distinguished from other kinds of ST; see also below).

The infinite line in the direction of the vector $\boldsymbol{\theta}$ is called an ST line. The operator T_n^1 reduces an n-dimensional function to a one-dimensional function defined as

$$T_n^1[f_n](p, \boldsymbol{\theta}) = \hat{f}_{1,\boldsymbol{\theta}}(p). \tag{2.2}$$

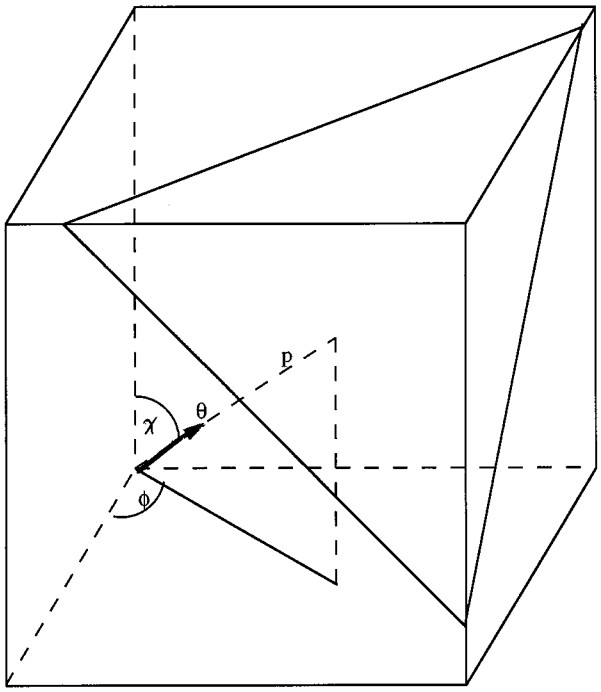

FIG. 1. *A cubic three-dimensional domain and the T_3^1 transformation plane in the direction $\boldsymbol{\theta}$ with projection length p. The $T_3^1[\cdot](p,\boldsymbol{\theta})$ represents the integral of a three-dimensional function over the plane.*

In the case of an isotropic function $f_n(\boldsymbol{s})$ with infinite support, the T_n^1 is also isotropic, i.e., the function $\hat{f}_{1,\boldsymbol{\theta}}(p)$ is independent of the orientation of the direction vector $\boldsymbol{\theta}$. This is not true in the case of a finite support that lacks spherical symmetry. Figure 1 shows the intersection of a transformation plane with a compact rectangular support. The equations that determine the intersections of the transformation planes in a cubic support are obtained in Appendix I. The ST $\hat{f}_{1,\boldsymbol{\theta}}(p)$ is equal to the integral of $f_3(\boldsymbol{s})$ over the planar domain defined by the intersection of the support with the infinite plane that is perpendicular to $\boldsymbol{\theta}$ and is located at distance p from the origin. A different direction vector will, in general, lead to a different ST function. Note that the finite support acts as a spatial filter for the n-dimensional function, which permits the ST integral operator (2.1) to be always well defined. In exchange, the compact support introduces boundary effects in the space-transformed functions.

We present without proof certain properties for the T_n^1 that will be used later; proofs can be obtained directly from the T_n^1 definition using the properties of generalized functions (an excellent review of the theory of generalized functions may be found in Gel'fand and Shilov (1964)):

(a) scaling:

$$T_n^1[f_n(\lambda \boldsymbol{s})] = \frac{1}{\lambda^{n-1}} \hat{f}_{1,\boldsymbol{\theta}}(\lambda p); \tag{2.3}$$

(b) shifting:

$$T_n^1[f_n(\boldsymbol{s}-\mathbf{a})] = \hat{f}_{1,\boldsymbol{\theta}}(p-\boldsymbol{\theta}\cdot\mathbf{a}); \text{ and} \tag{2.4}$$

(c) complementarity:

$$\hat{f}_{1,\boldsymbol{\theta}}(p) = \hat{f}_{1,-\boldsymbol{\theta}}(-p). \tag{2.5}$$

The complementarity property reflects the fact that the mapping from the space $R \times S^{n-1}$ onto the space of the n-hyperplanes leads to a double covering of the hyperplane space. Single covering is obtained by restricting the transformation to positive projections $p > 0$ or to one hemisphere of the unit sphere.

Another ST is the Ψ_n^1-operator defined by

$$\Psi_n^1[f_n](p, \boldsymbol{\theta}) \equiv \Omega T_n^1[f_n](p, \boldsymbol{\theta}) = f_{1,\boldsymbol{\theta}}(p), \tag{2.6}$$

where Ω is a differential operator such that

$$\Omega = \frac{(-1)^m}{2(2\pi)^{2m}} \times \begin{cases} S_{2m+1} \dfrac{\partial^{2m}}{\partial p^{2m}}[\cdot], & n = 2m+1, \\ 2\pi S_{2m} H \left\{ \dfrac{\partial^{2m-1}}{\partial p^{2m-1}} \right\}[\cdot], & n = 2m, \end{cases} \tag{2.7}$$

where $S_n = 2\pi^{n/2}/\Gamma(\frac{n}{2})$ represents the surface area of the n-dimensional unit sphere S^{n-1}, Γ is the gamma function, and H denotes the Hilbert transform (the Ψ_n^1 is sometimes called the ST-2).

The inverse ST Ψ_1^n and T_1^n of the ST Ψ_n^1 and T_n^1, respectively, exist and are defined by

$$\Psi_1^n[f_{1,\boldsymbol{\theta}}](\mathbf{s}) = \frac{1}{S_n} \int_{S_n} f_{1,\boldsymbol{\theta}}(\boldsymbol{s} \cdot \boldsymbol{\theta}) \mathbf{d}\boldsymbol{\theta} = f_n(\mathbf{s}) \tag{2.8}$$

and

$$T_1^n[\hat{f}_{1,\boldsymbol{\theta}}](\mathbf{s}) = \Psi_1^n \Omega[\hat{f}_{1,\boldsymbol{\theta}}](\mathbf{s}) = f_n(\mathbf{s}). \tag{2.9}$$

In the spectral domain, the ST and their inverses turn out to have simple algebraic forms. In particular, for the T_n^1 we have

$$T_n^1[\tilde{f}_n](\omega, \boldsymbol{\theta}) = \tilde{f}_n(\mathbf{w}) = \hat{\tilde{f}}_{1,\boldsymbol{\theta}}(\omega) \tag{2.10}$$

along the wave vector $\mathbf{w} = \omega\boldsymbol{\theta}$, where $\tilde{f}$ denotes the Fourier transform of f. For the inverse ST T_1^n we have

$$T_1^n[\hat{\tilde{f}}_{1,\boldsymbol{\theta}}](\mathbf{w}) = \hat{\tilde{f}}_{1,\boldsymbol{\theta}}(\omega) = \tilde{f}_n(\mathbf{w}). \tag{2.11}$$

Similarly, for the Ψ_n^1 it can be shown that the spectral relation

$$\Psi_n^1[\tilde{f}_n](\omega) = \frac{S_n |\omega|^{n-1} \tilde{f}_n(\mathbf{w})}{(2\pi)^{n-1}} = \tilde{f}_{1,\boldsymbol{\theta}}(\omega) \tag{2.12}$$

holds, while for the Ψ_1^n the corresponding relation is

$$\Psi_1^n[\tilde{f}_{1,\boldsymbol{\theta}}](\mathbf{w}) = \frac{(2\pi)^{n-1} \tilde{f}_{1,\boldsymbol{\theta}}(\omega)}{S_n |\omega|^{n-1}} = \tilde{f}_n(\mathbf{w}). \tag{2.13}$$

3. Numerical implementations of the ST. The ST flow solution is formulated in terms of the ST T_3^1 and its inverse T_1^3. In this work we use a real-space implementation described below. In the case of unbounded domains spectral algorithms may lead to faster numerical codes through the use of fast fourier transforms. First, we study the implementation of the ST involved in the flow solution, and we investigate issues related to the accuracy and the numerical complexity of the algorithms.

3.1. The T_3^1 algorithm. Consider the three-dimensional, spherically symmetric Gaussian function

$$f_3(\boldsymbol{s}) = \exp\left[-\frac{(\mathbf{s}-\mathbf{a})^2}{\ell^2}\right] \tag{3.1}$$

(the translation $\mathbf{a}$ determines the location of the maximum). Here we assume that the Gaussian function is centered at the center of the domain. The T_3^1 can be implemented on a function $f_3(\boldsymbol{s})$ defined over a rectangular support either continuously or discretely by its values at the nodes of the numerical grid. A set of user-specified unit vectors is generated, which defines the directions of the transformation planes. The choice of the transformation directions is discussed in more detail in section 3.2 on the inverse ST. Each unit vector $\boldsymbol{\theta}(\chi,\phi)$ is defined in terms of the polar angle and azimuthal angles (see Fig. 1). The unit vectors have the standard representation in the Cartesian coordinate system

$$\boldsymbol{\theta} = (\sin\chi\cos\phi, \sin\chi\sin\phi, \cos\chi). \tag{3.2}$$

In order to obtain a single covering, as we discussed in section 2 above, only unit vectors $\boldsymbol{\theta}$ in the northern hemisphere (i.e., $\chi \in [0, \pi/2]$) are used. Hence, the projection length p is allowed to assume both positive and negative values.

The T_3^1 involves the planar integrals of the three-dimensional field over the planes perpendicular to the direction vectors for specified projection lengths. Note that the numerical evaluation of the planar integrals involves the value of the function at off-grid points. In the case of a function that is defined only by means of its discrete values on the nodes of a grid, a linear interpolation scheme is used to obtain intermediate values between the grid nodes. For a random field, oversampling with this interpolation scheme may lead to artificial smoothing and variance reduction. Thus, we have implemented a discrete-summation approximation of the integral that adapts to the discretization of the three-dimensional grid. We have also tested a standard adaptive quadrature integration routine for use with smoother functions. If the planar domains involved in the calculation of the ST are large, numerical integration by means of the techniques described above may converge slowly. In this case, as well as in the case of random fields with strong fluctuations, Monte Carlo methods could provide a more efficient integration scheme.

The accuracy of the numerical representation and the integration schemes was evaluated by means of the T_3^1 of the function $f_3(\boldsymbol{s})$ in (3.1). Using the definitions (2.1) and (2.2), together with the scaling (2.3) and shifting (2.4) properties, the T_3^1 of the Gaussian function (3.1) is found to be

$$T_3^1[f_3(\boldsymbol{s})] = \pi\ell^2 \exp\left[-\frac{(p-\mathbf{a}\cdot\boldsymbol{\theta})^2}{\ell^2}\right]. \tag{3.3}$$

In order to accurately implement the ST by means of the asymptotic expression (3.3), the grid dimensions must be larger than the characteristic length of the Gaussian, since

otherwise the T_3^1 will be underestimated. For the purpose of this calculation we use a cubic grid with $N_3 = 49$ nodes per side and length of each side $L = 6\ell$. This support contains 99.7% of the infinite integral involved in (3.3). For the purpose of one-dimensional integration, a fixed discretization with $N_P = 49$ points per transformation line is used.

In Fig. 2 we plot the analytical results from (3.3) and the numerical estimates of the T_3^1 transformation along three directions (lines) in space. Both numerical estimates are based on a discrete-summation approximation of the plane integral. The plots in Fig. 2 compare the analytical expression (3.3) with the numerical ST obtained (a) using the discrete representation of the function at the grid nodes with linear interpolation for points that are not on the grid, and (b) using the exact values of the Gaussian at the intermediate points. All three methods lead to almost identical results. In the plots we show the exact solution (continuous lines) and the estimates obtained using the discrete representation of the function (triangles and circles). Both representation schemes provide accurate estimates of the analytical expression (3.3).

In order to provide quantitative error estimates we used the normalized estimation error defined by

$$S_3^1(\boldsymbol{\theta}) = \frac{\sum_{i=1}^{N_P-2} [\hat{f}^*_{1,\boldsymbol{\theta}}(p_i) - \hat{f}_{1,\boldsymbol{\theta}}(p_i)]^2}{\sum_{i=1}^{N_P-2} \hat{f}^{*2}_{1,\boldsymbol{\theta}}(p_i)}, \tag{3.4}$$

where $\hat{f}^*_{1,\boldsymbol{\theta}}(p_i)$ is the numerical value of the T_3^1 in the $\boldsymbol{\theta}$-direction. The T_3^1 summation along $\boldsymbol{\theta}$ does not include the values of the transformation at the endpoints of the line. The values of the normalized estimation error for five directions $\boldsymbol{\theta}$ are given in Table 1. The first entry in each of the rows of the second column represents the value of the error parameter using the interpolation scheme; the second entry represents the value obtained using the exact representation of the three-dimensional function. In both schemes the normalized estimation error is less than 0.05%. Therefore, we conclude that the discrete on-grid representation—in combination with the linear interpolation for off-grid points—provides an accurate scheme for the calculation of T_3^1. The integrations involved in T_3^1 were carried out using the trapezoidal rule. Nearly identical results are obtained if an adaptive quadrature integration scheme is used instead. In calculations of the T_3^1 of random fields the trapezoidal rule is used instead of the adaptive quadrature for the reasons discussed above. The dependence of $S_3^1(\boldsymbol{\theta})$ on the grid discretization for five random directions is shown in Fig. 3.

Finally, we have studied the performance of the T_3^1 routine in terms of CPU time, denoted by t_{CPU}, for various sizes of the three-dimensional grid. The numerical calculations for this test have been performed on an SGI Iris R3000 workstation. In Fig. 4 we show a logarithmic plot of the CPU time t_{CPU} (in seconds) vs the logarithm of the grid size N_3. The slope of the curve represents the exponent of the t_{CPU} vs. N_3 power-law scaling, and is approximately equal to 1.95 ± 0.13. The primary cost of T_3^1 is due to the integration over the transformation planes. For a fixed number of ST lines this process scales proportionally with the number of nodes on the planes. Therefore, a leading-order estimate that neglects finite-size effects gives $t_{CPU} \propto N_3^2$. The observed scaling exponent is smaller, due to finite-size effects.

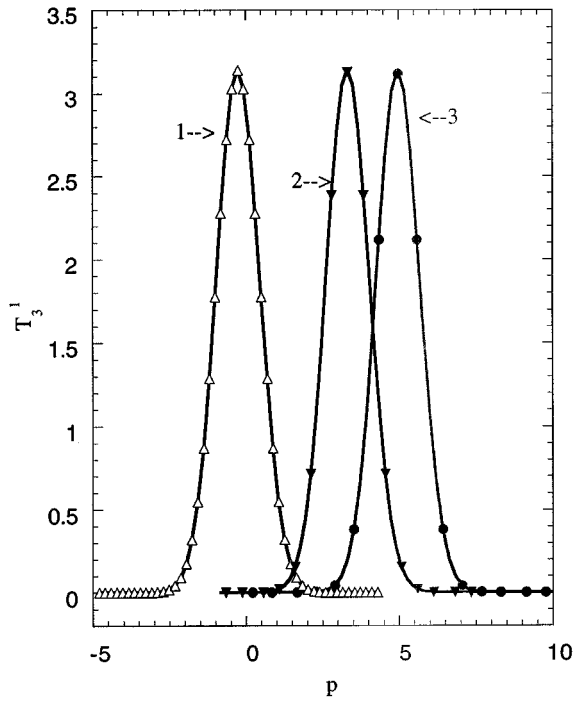

FIG. 2. *Plots of the exact solution (continuous lines) and the numerical estimates (markers) of the T_3^1 transformation of the Gaussian function* (3.1) *for three randomly selected transformation directions $\boldsymbol{\theta}(\chi,\phi)$. The direction* (1) *corresponds to* $(1.22, 0.68)$*; the direction* (2) *to* $(1.73, 2.16)$*; and the direction* (3) *to* $(0.94, 2.98)$*. For the direction* (1) *we show all the estimation points (open triangles), while 15 points are shown in the directions* (2) *(full triangles) and* (3) *(full circles).*

TABLE 1
Values of the normalized error estimate $S_3^1(\boldsymbol{\theta})$ for five different ST directions as defined by the direction vector $\boldsymbol{\theta}(\chi,\phi)$.

$\boldsymbol{\theta}(\chi,\phi)$	$S_3^1(\boldsymbol{\theta})$
(1.2247,0.6771)	0.433 E–4
	0.188 E–4
(0.8742,1.4147)	0.191 E–5
	0.622 E–5
(0.9410,2.9841)	0.233 E–5
	0.352 E–5
(1.7097,0.3216)	0.128 E–4
	0.256 E–5
(1.7341,2.1635)	0.374 E–4
	0.201 E–4

3.2. The T_1^3 algorithm. The inverse ST T_1^3 of the one-dimensional function $\hat{f}_{1,\boldsymbol{\theta}}(p)$ involves two steps: (i) application of the Ω operator (2.7) to obtain $f_{1,\theta}(p)$ and (ii) using the Ψ_1^3 ST (2.8) to reconstruct the three-dimensional function $f_3(\boldsymbol{s})$. The Ω operator in three dimensions is given by

$$f_{1,\boldsymbol{\theta}}(p) = \Omega[\hat{f}_{1,\boldsymbol{\theta}}(p)] = -\frac{1}{2\pi}\frac{\partial^2 \hat{f}_{1,\boldsymbol{\theta}}(p)}{\partial p^2}. \tag{3.5}$$

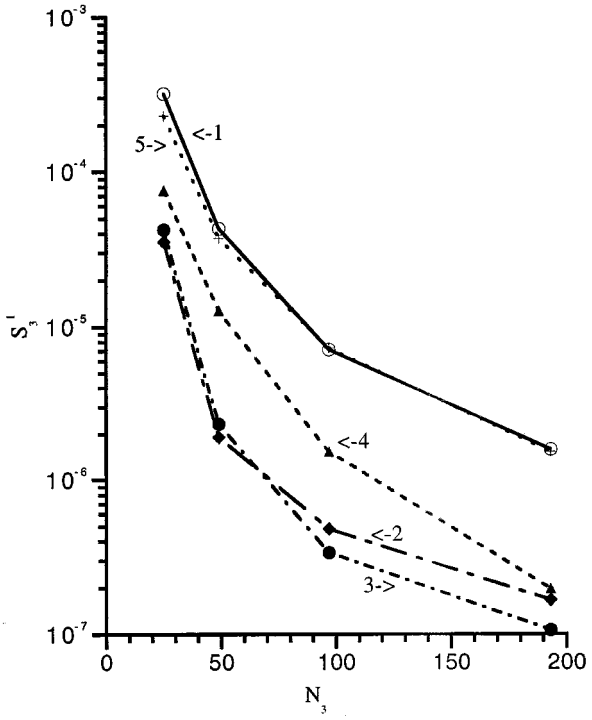

FIG. 3. *Plot of the error estimate* S_3^1 *for the function* (3.1) *vs. the number of nodes per side* N_3 *for five transformation directions:* $(1.22, 0.68)$, $(0.87, 1.41)$, $(0.94, 2.98)$, $(1.73, 2.16)$, *and* $(1.71, 0.32)$. *The* T_3^1 *transformation is evaluated using* $N_L = 200$ *lines and* $N_P = 90$ *points per transformation line.*

Using the polar and azimuthal angles, the Ψ_1^3 in three dimensions is defined by

$$f_3(\boldsymbol{s}) = \Psi_1^3[f_{1,\boldsymbol{\theta}}](s) = \frac{1}{4\pi}\int_0^\pi d\phi \int_0^\pi d\chi \sin\chi f'_{1,\boldsymbol{\theta}}(\boldsymbol{s}\cdot\boldsymbol{\theta}), \tag{3.6}$$

where $f'_{1,\boldsymbol{\theta}}(p) = f_{1,\boldsymbol{\theta}}(p) + f_{1,\boldsymbol{\theta}}(-p)$. In theory, the Ψ_1^3 reconstructs exactly a three-dimensional function from its projections on all possible directions (Christakos (1984a, 1992)). In practice, however, Ψ_1^n is evaluated from a finite set of projections, and (3.6) is replaced with a discrete summation. Discretization poses a number of problems related to the nonuniqueness of the function reconstructed from a finite number of projections (Louis (1986)). Fortunately, it has been shown that arbitrarily good approximations can be constructed by increasing the number of ST lines. It is also possible to improve the solution by imposing conditions based on optimality criteria or a priori information. In addition, the inverse problem is technically ill posed, thus leading to solutions that may be unstable with respect to small perturbations of the projections (e.g., Tikhonov and Goncharsky (1987)). The ill-posedness may be particularly relevant in the calculation of ST of spatial random fields. The stability of the ST solutions for random fields can be improved by using frequency filters that smooth out the fluctuations of the field (Jain (1989)). Several schemes have been investigated that lead to different choices of ST lines (e.g., Jain (1989), Tompson, Ababou, and Gelhar (1989), Deans (1993)). Ill-posedness is due to the numerical implementation of the inverse operator $T_1^3 = \Psi_1^3\Omega$ (2.9) that involves the second-

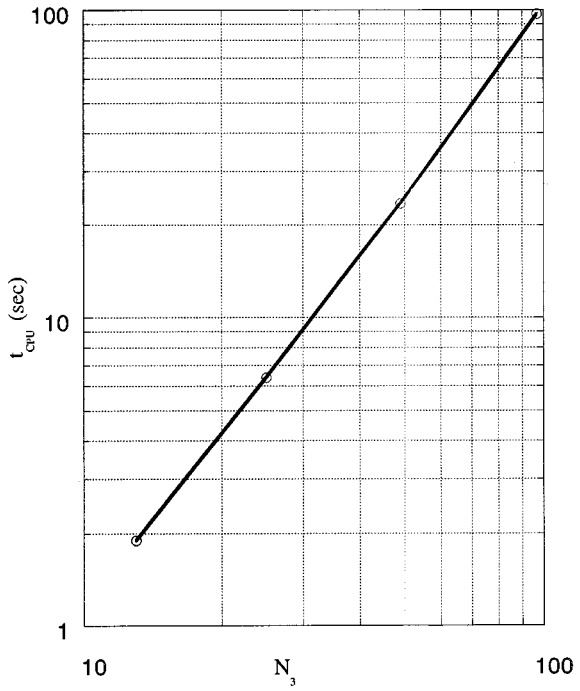

FIG. 4. *Log-log plot of the CPU time (seconds) vs. the grid size. All transformations have been calculated using $N_L = 200$ lines and a line discretization of $N_P = 49$ points per transformation line on an SGI Iris R3000 workstation.*

order derivative of the T_3^1. Hence, the inverse ST operation becomes unstable to small perturbations of the projections. However, the ST solution of the flow equation presented in section 4.1 below does not involve the differential operator Ω, and, thus, it overcomes this problem.

Herein we use a discretization scheme that leads to a uniform distribution of the projections over the surface of the unit sphere (Jain (1989)). The set of direction vectors $\boldsymbol{\theta}_{jk}$ is given in terms of pairs (χ_j, ϕ_{jk}) of azimuthal and polar angles determined by the following expressions:

$$\chi_j = \left(j - \frac{1}{2}\right)\frac{\pi}{N}, \qquad j = 1, 2, \ldots, N, \tag{3.7}$$

and

$$\phi_{jk} = \frac{2\pi k}{K_j}, \qquad k = 1, 2, \ldots, K_j, \tag{3.8}$$

where the number K_j of azimuthal angles in each polar direction is given by the expression

$$K_j = \left\lfloor \frac{2\pi \sin \chi_j}{c} \right\rfloor, \tag{3.9}$$

where $\lfloor\ \rfloor$ denotes the integer part and the constant c is obtained by specifying K_1.

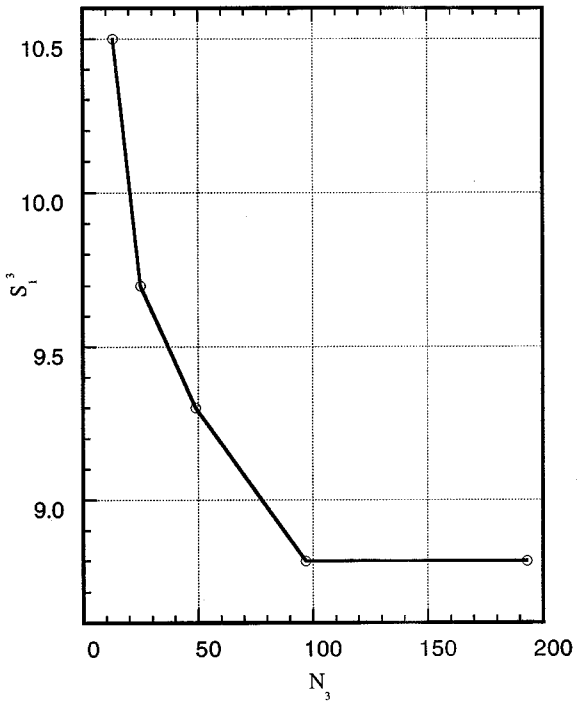

FIG. 5. *Plot of the normalized estimation error S_1^3 for the Ψ_1^3 transformation of the one-dimensional function* (3.12) *vs the grid size N_3. The transformation was evaluated using $N_L = 200$ transformation lines and $N_P = 90$ nodes per line.*

The total number of ST lines generated by this method is equal to

$$N_L = \sum_{j=0}^{N} K_j. \tag{3.10}$$

The numerical implementation of the inverse ST Ψ_1^3 for a user-specified finite set of N_L lines is obtained in terms of the summation of the Ψ_3^1-values of the function $f_3(\boldsymbol{s})$ for all N_L directions as follows:

$$f_3(\boldsymbol{s}) = \frac{1}{N_L} \sum_{i=1}^{N_L} f_{1,\boldsymbol{\theta}_i}(\mathbf{s} \cdot \boldsymbol{\theta}_i). \tag{3.11}$$

The above discrete approximation of (3.6) is justified because the projections of the ST lines are uniformly distributed over the unit sphere, and consequently they all have identical integration weights $1/N_L$. Note that (3.11) can also be used when the transformation lines are randomly oriented with uniform distribution over the sphere, as for the reconstruction routine of the turning bands code (Tompson, Ababou, and Gelhar (1989)). We used both uniformly and randomly oriented ST lines to test the accuracy of the inverse ST, but only uniformly oriented lines for solving the flow equation (section 4 below), because this scheme gave more accurate results.

The Ω operator does not have to be implemented explicitly for solving the flow equation, since the second derivative of the T_3^1 is obtained directly as the solution

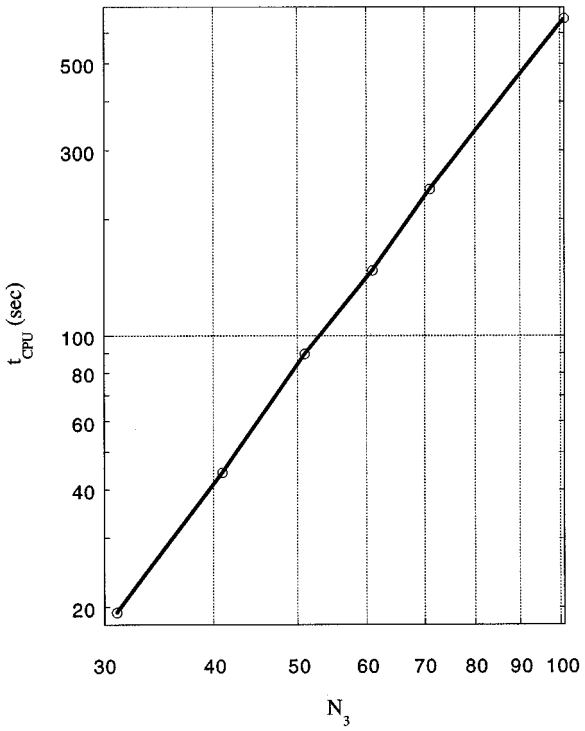

FIG. 6. *Log-log plot of the CPU time (seconds) required for the evaluation of the* Ψ_1^3 *transformation of the one-dimensional function* (3.12) *versus the grid size. The* Ψ_1^3 *transformation has been evaluated using* $N_L = 196$ *lines on an HP* 9000/735 *workstation.*

of the transformed one-dimensional equation, as we show in section 4 below. Hence, we are interested in the accuracy of the operator Ψ_1^3. Note that in view of (2.6) and (2.8), $\Psi_1^3\Psi_3^1 = I$, where I denotes the identity operator. The numerical Ψ_1^3 inversion routine was tested by applying it on the transformation $\Psi_3^1[f_3(\mathbf{s})]$ of the Gaussian (3.1), where

$$\Psi_3^1[f_3(\boldsymbol{s})] = \left[1 - \frac{2(p - \mathbf{a}\cdot\boldsymbol{\theta})^2}{\ell^2}\right] \exp\left[-\frac{(p - \mathbf{a}\cdot\boldsymbol{\theta})^2}{\ell^2}\right]. \tag{3.12}$$

The numerical result was then compared with (3.1). There are three discretization parameters that affect the accuracy of the inversion routine: (i) the number of transformation lines N_L; (ii) the discretization of the three-dimensional grid, i.e., the number of nodes per side N_3; and (iii) the line discretization, i.e., the number of points per transformation line N_P.

We examined the accuracy of the inversion routine as a function of the above discretization parameters using a cubic grid with sides $L = 6\ell$. In order to evaluate the performance of the inversion routine we compare the summation of the Gaussian field values over the grid with the sum of the values estimated using the inversion scheme (3.6). The normalized estimation error is defined by

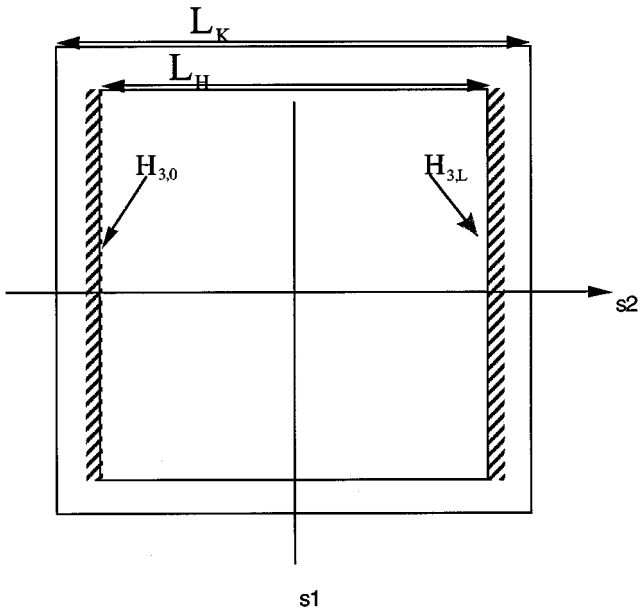

FIG. 7. *A two-dimensional cross section of the head and conductivity domains used in the flow problem. The hydraulic conductivity is defined on a cubic grid of size L_K, while the hydraulic head is evaluated on a cubic grid of size L_H. The constant hydraulic head conditions are imposed on the two faces of the head domain normal to the direction vector $\mathbf{e}_2$.*

$$S_1^3 = \frac{\sum_{\mathbf{x}\in Grid} [f_3^*(\mathbf{x}) - f_3(\mathbf{x})]^2}{\sum_{\mathbf{x}\in Grid} [f_3^*(\mathbf{x})]^2}, \tag{3.13}$$

where $f_3^*(\mathbf{x})$ represents the estimate obtained from the numerical inversion, and $f_3(\mathbf{x})$ represents the actual value of the field given by (3.1) above. We first evaluated the Ψ_1^3 with an inversion routine that uses 200 randomly oriented transformation lines and a line discretization with 90 points per transformation line. In Fig. 5 we plot S_1^3 as a function of the nodes per side N_3. The S_1^3 levels off at a value of 8.8% for N_3 greater than 97, and it decreases by less than 2% as N_3 increases from 13 to 193. We also investigated the accuracy of the inversion algorithm by using uniformly oriented lines, and found very similar results. In addition, we tested the accuracy of the inversion algorithm with respect to the number of lines. We found that a large increase in the number of randomly oriented lines (from 100 to 1000), resulted in about 15% reduction in S_1^3. The reduction in S_1^3 was less noticeable for uniformly oriented lines. Furthermore, we found that in solving the flow equation the accuracy does not improve by increasing the number of uniformly oriented lines above a certain threshold, as we will discuss in more detail in section 4 below. We also tested the accuracy of the inversion routine as a function of the line discretization N_P and found that it is not sensitive to this parameter.

Finally, we investigated the efficiency of the numerical Ψ_1^3 operation. We found that the CPU time t_{CPU} scales linearly with the number of ST lines used, as is expected based on (3.11). In Fig. 6 we plot the logarithm of the CPU time (in seconds) required for the evaluation of the Ψ_1^3 vs. the logarithm of the grid discretization. We

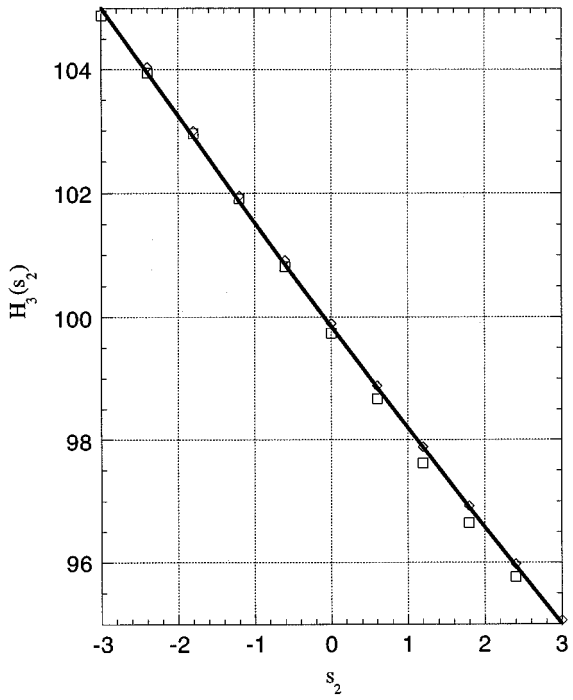

FIG. 8. *Hydraulic head profile vs. coordinate in the flow direction for linear conductivity variation. The exact solution (continuous line) and the ST estimate through the planes* $s_1 = s_3 = 0$ *(squares) and* $s_1 = s_3 = -L/2$ *(diamonds) are shown.*

used 196 lines for this run. It is shown that the CPU time scales as $t_{CPU} \propto N_3^{2.99\pm0.10}$. The exponent of this experimental scaling relationship shows that, for a constant number of lines, the CPU time scales linearly with the number of three-dimensional nodes N_3^3.

4. Solution of the three-dimensional flow equation.

4.1. A theoretical review of the ST method. Christakos and Hristopulos (1994, 1997) have presented an ST procedure for the numerical solution of the steady-state flow equation

$$\sum_{j=1}^{3}\left[\frac{\partial^2 H_3(\boldsymbol{s})}{\partial s_j^2}+\frac{\partial \ln K_3(\mathbf{s})}{\partial s_j}\frac{\partial H_3(\mathbf{s})}{\partial s_j}\right]=0, \tag{4.1}$$

where $H_3(\mathbf{s})$ is the hydraulic head and $K_3(\mathbf{s})$ is the hydraulic conductivity. Below, we briefly review the main results of the ST approach.

The three-dimensional hydraulic head is given by the expression

$$H_3(\boldsymbol{s})=-\frac{1}{2\pi}\Psi_1^3\left[\hat{J}_{1,\boldsymbol{\theta}}(0)\tilde{\kappa}_{\boldsymbol{\theta}}(\mathbf{s}\cdot\boldsymbol{\theta})\exp\left\{-\int_0^{\boldsymbol{s}\cdot\boldsymbol{\theta}} du\,\tilde{\kappa}_{\boldsymbol{\theta}}(u)\right\}\right], \tag{4.2}$$

where

$$\hat{J}_{1,\boldsymbol{\theta}}(0)=-\frac{\partial \hat{H}_{1,\boldsymbol{\theta}}(\sigma)}{\partial\sigma}\bigg|_{\sigma=0} \tag{4.3}$$

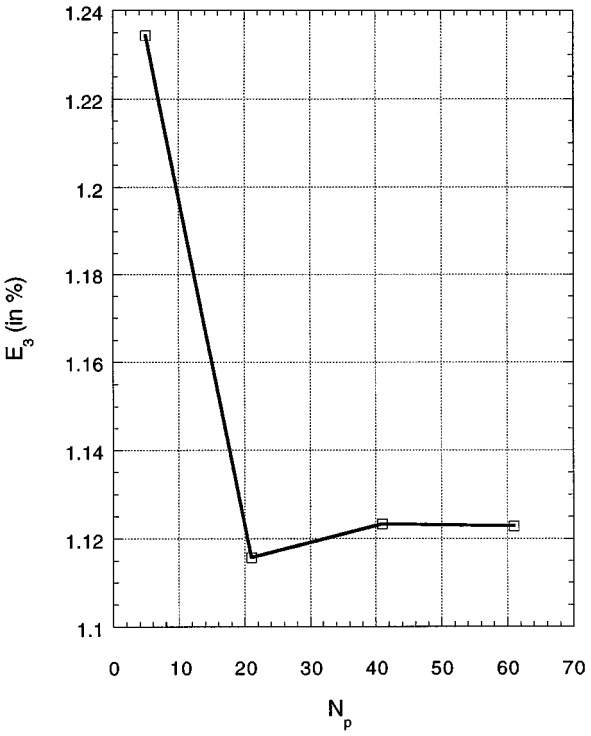

FIG. 9. *Plot of the normalized estimation error E_3 vs. the line discretization (i.e., the number of points per transformation line).*

is a function that depends on the direction vector and represents the unidimensional boundary conditions (BC). The solution (4.2) is based on an ST of the three-dimensional flow equation that leads to unidimensional flow equations in the direction of the ST lines. In the above, surface terms generated by the ST of the partial derivatives are neglected. The function $\tilde{\kappa}_{\boldsymbol{\theta}}(\cdot)$ is given by the ratio of two STs that involve both the unknown local three-dimensional hydraulic gradient and the hydraulic conductivity. In the case of a constant specific discharge vector $\mathbf{Q} = -K_3(\mathbf{s})\nabla H_3(\mathbf{s})$, it is possible to express the function $\tilde{\kappa}_{\boldsymbol{\theta}}(\cdot)$ purely in terms of the hydraulic conductivity ST as

$$\tilde{\kappa}_{\boldsymbol{\theta}}(p) = \frac{T_3^1[K_3^{-1}(\mathbf{s})(\boldsymbol{\theta}\cdot\nabla \ln K_3(\mathbf{s}))](p,\boldsymbol{\theta})}{T_3^1[K_3^{-1}(\mathbf{s})](p,\boldsymbol{\theta})}. \tag{4.4}$$

In more general situations, the function $\tilde{\kappa}_{\boldsymbol{\theta}}(\cdot)$ can be estimated by means of the uniform hydraulic gradient approximation presented in Appendix II. This scheme as well as the solution of the flow equation including the surface terms will be investigated in detail in a forthcoming publication.

The numerical implementation of (4.2) is based on the discretized expression

$$H_3(\mathbf{s}) = -\frac{1}{2\pi N_L}\sum_{i=1}^{N_L}\hat{J}_{1,\boldsymbol{\theta}_i}(0)\tilde{\kappa}_{\boldsymbol{\theta}_i}(\mathbf{s}\cdot\boldsymbol{\theta}_i)\exp\left[-\int_0^{\mathbf{s}\cdot\boldsymbol{\theta}_i} du\,\tilde{\kappa}_{\boldsymbol{\theta}_i}(u)\right]. \tag{4.5}$$

In the case of three-dimensional flow within a rectangular domain the BC commonly involve constant hydraulic head on two opposite domain faces and no-flow conditions

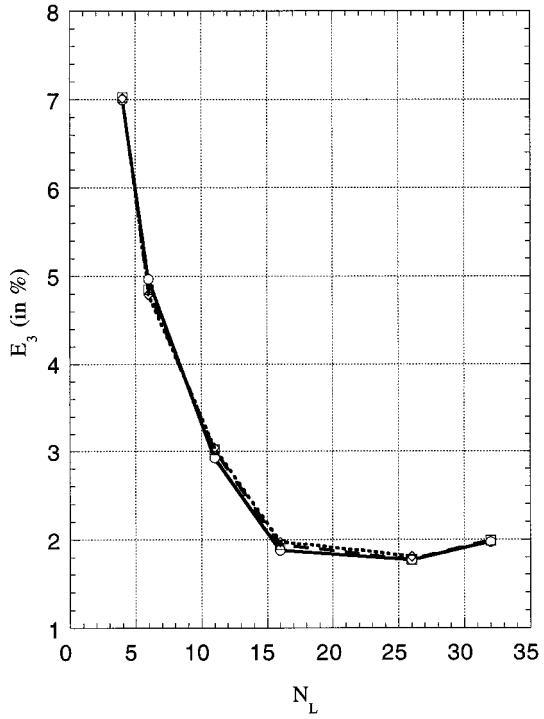

FIG. 10. *Plot of the normalized estimation error E_3 vs. the number of transformation lines for the following discretizations ($N_P = N_3$):* (i) $N_3 = 21$ *(circles),* (ii) $N_3 = 41$ *(squares), and* (iii) $N_3 = 81$ *(diamonds).*

on the remaining faces (e.g., de Marsily (1986)). The unidimensional BC given by (4.3) above represent the rate of change of the hydraulic head projection on the transformation planes. Hence, the unidimensional BC cannot be determined entirely from the three-dimensional boundaries. In practice, a finite number N_L of ST lines is used, and the unidimensional BC are subsequently represented by a finite set of N_L values $\hat{J}_{1,\boldsymbol{\theta}}(0)$. If the hydraulic head is known at N_L locations on the boundaries or within the flow domain, the $\hat{J}_{1,\boldsymbol{\theta}}(0)$ can be determined by solving the linear system of N_L equations generated by (4.5). In this scheme there is a one-to-one correspondence between the three-dimensional BC and the unidimensional BC. In addition, an ST expression can be obtained for the hydraulic gradient directly from (4.2). The normal component of the gradient at the boundaries is then given by

$$\begin{aligned}(4.6)\qquad \mathbf{n}\cdot\nabla H_3(\mathbf{s}) &= -\frac{1}{2\pi N_L}\sum_{i=1}^{N_L}(\mathbf{n}\cdot\boldsymbol{\theta}_i)\hat{J}_{1,\boldsymbol{\theta}_i}(0)\left[\frac{\partial\tilde{\kappa}_{\boldsymbol{\theta}_i}(\mathbf{s}\cdot\boldsymbol{\theta}_i)}{\partial\sigma}\,\bigg|_{\sigma=\boldsymbol{s}\cdot\boldsymbol{\theta}_i} - \tilde{\kappa}^2_{\theta_i}(\boldsymbol{s}\cdot\boldsymbol{\theta}_i)\right]\\ &\quad\times\exp\left[-\int_0^{s\cdot\boldsymbol{\theta}_i} du\,\tilde{\kappa}_{\boldsymbol{\theta}_i}(u)\right],\end{aligned}$$

where $\mathbf{n}$ is the outward normal on the surface of the boundary. Equation (4.6) can be used in order to impose no-flow conditions on the Neumann boundaries.

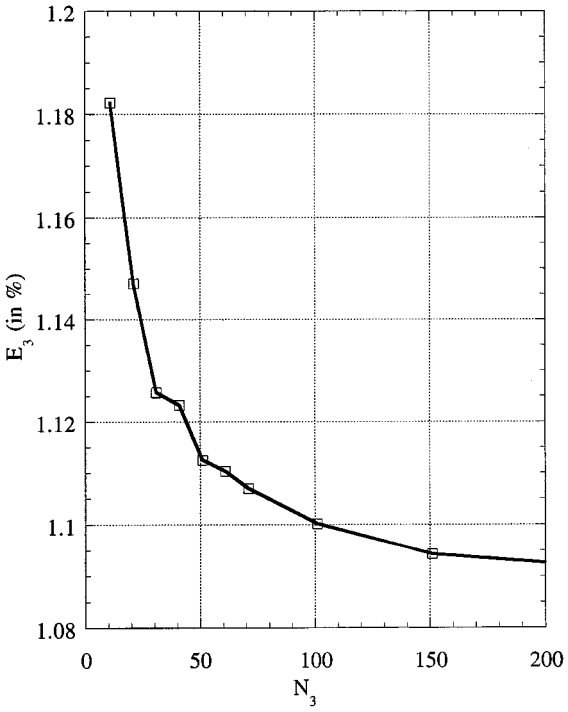

FIG. 11. *Plot of the normalized estimation error E_3 vs. the discretization of the three-dimensional grid.*

4.2. Numerical implementation of the ST method. Certain technical aspects of the ST implementation are addressed here. The first step in the procedure is the definition of a numerical grid that covers the flow domain. In the simulations we use separate cubic grids for the hydraulic conductivity and the hydraulic head. The hydraulic conductivity is defined on a cubic grid of size L_K, while the hydraulic head is evaluated on a cubic grid of size L_H. Both grids are centered at the origin, and the hydraulic head grid is equal to or smaller than the conductivity grid, i.e., $L_H \leq L_K$. In Fig. 7 we show a two-dimensional cross section. The two-grid scheme is used in order to compensate for the noninclusion of the surface terms in the space-transformed flow equation. The BC involve constant head on the two faces of the head domain normal to the direction vector $\mathbf{e}_2$, and no-flow conditions through the remaining domain faces. In most of the simulations we used grids with 41 nodes per side for both the hydraulic conductivity and head domains. Note that in general the domain for the head is smaller than that of the conductivity and, therefore, the nodes are not collocated. It is also possible to use different discretization for the head and the conductivity domains. The length L_K of the conductivity domain is an arbitrary number related to the grid spacing (resolution) b through the expression $L_K = N_3 b$. The size of the domain is assumed to be given and the discretization is determined on the basis of the required resolution and computational cost. In this study we illustrate the ST flow solution using a grid of length $L_K = 6$ (in appropriate length units). The length L_H of the head domain is defined by the aspect ratio L_K/L_H, which is used as a free parameter. We experimented with aspect ratios in the range between one and three.

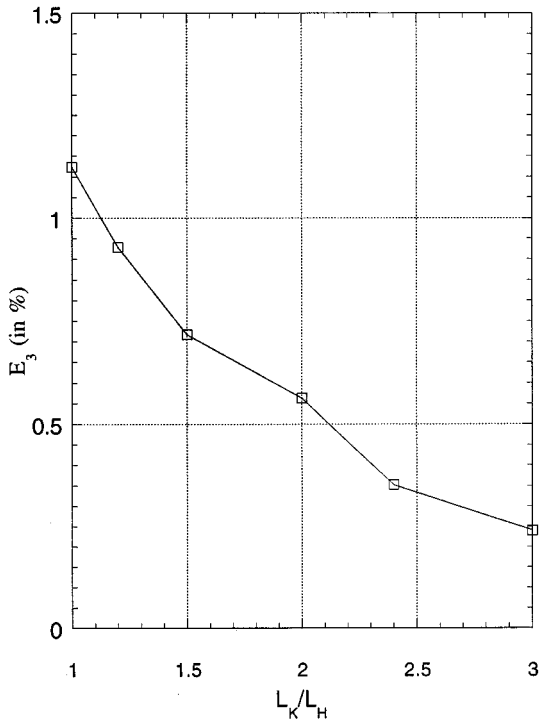

FIG. 12. *Plot of the normalized estimation error E_3 vs. the aspect ratio L_K/L_H.*

The gradients of both the log-conductivity and the hydraulic head are approximated using first-order differences (Abramowitz and Stegun, 1970):

$$W_i(\mathbf{s}) = \frac{\ln K_3(\mathbf{s} + b\mathbf{e}_i) - \ln K_3(\mathbf{s} - b\mathbf{e}_i)}{2b}, \tag{4.7}$$

where $i = 1, 2$, and 3; $\mathbf{e}_i$ represents the unit vector in the i-direction; and $W(\mathbf{s}) = \nabla \ln K_3(\mathbf{s})$ represents the log-conductivity gradient.

A set of direction vectors for the ST lines is obtained using a discrete number of uniformly oriented ST lines as discussed in section 3 above. This approach is computationally more efficient than using randomly oriented lines. Finally, the steps involved in the evaluation of the hydraulic head by means of (4.5) above are as follows. First, the one-dimensional functions $\tilde{\kappa}_{\boldsymbol{\theta}}(\cdot)$ are obtained along the N_L ST lines by numerical evaluation of the T_3^1 involved in (4.4). The $\tilde{\kappa}_{\boldsymbol{\theta}}(\cdot)$ are evaluated over the conductivity domain. Secondly, the head-generating function m_1 defined by

$$m_1(\boldsymbol{\theta}, p) = \tilde{\kappa}_{\boldsymbol{\theta}}(p) \exp\left[-\int_0^p du\, \tilde{\kappa}_{\boldsymbol{\theta}}(u)\right] \tag{4.8}$$

is evaluated; the trapezoidal rule is used in order to approximate the integral in the exponent of m_1. Thirdly, the N_L boundary coefficients $\hat{J}_{1,\boldsymbol{\theta}_i}(0)$ are calculated by solving a system of equations obtained by writing (4.5) at N_B nodes on the boundary of the hydraulic head domain. These equations are as follows:

$$H_3(\boldsymbol{s}_j^*) = -\frac{1}{2\pi N_L} \sum_{i=1}^{N_L} \hat{J}_{1,\boldsymbol{\theta}_i}(0) m_1(\boldsymbol{\theta}_i, \boldsymbol{s}_j^* \cdot \boldsymbol{\theta}_i), \tag{4.9}$$

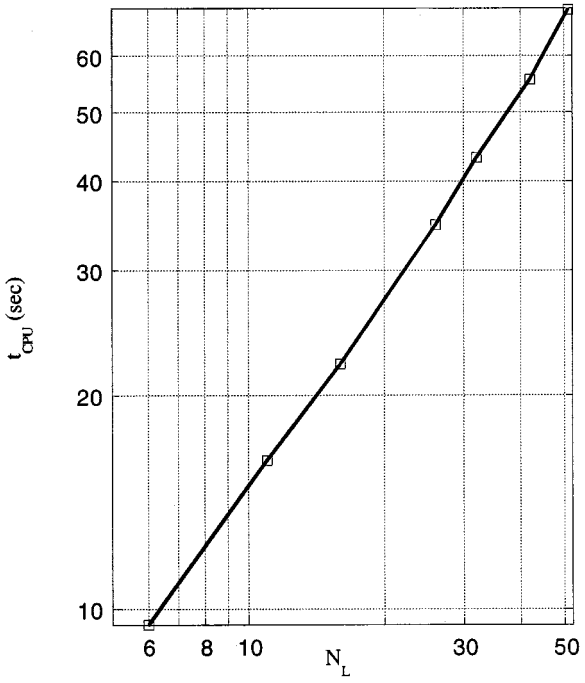

FIG. 13. *Plot of the CPU time (seconds) required for the ST solution of the flow equation on an HP* 9000/735 *workstation vs. the number of transformation lines.*

where $j = 1, \ldots, N_B, \boldsymbol{s}_j^*$ denotes a point on the head domain boundary, and $H_3(\boldsymbol{s}_j^*)$ denotes the value of the hydraulic head at this point. Finally, the hydraulic head is calculated at the nodes of the head numerical grid using (4.5). As we mentioned above, we kept the same discretization with N_3 nodes per side for both domains. In the third step we used an overdetermined system of equations, i.e., $N_B > N_L$, in order to avoid a singular system. The resulting system of equations was solved using the singular value decomposition (SVD) method (e.g., Press et al. (1989)). Additionally we found that for a given length L_H of the hydraulic head domain, the accuracy of the calculated hydraulic head improves by increasing the length L_K of the conductivity domain.

At this point it must be noted that the implementation of the ST flow solution is ideally suited for parallel implementation, since the T_3^1 can be computed independently for each transformation line.

4.3. Flow in linear hydraulic conductivity field. The flow equation (4.1) is first solved in the special case of a hydraulic conductivity field of the form

$$K_3(\mathbf{s}) = K_0 + \mathbf{g} \cdot \mathbf{s}, \tag{4.10}$$

where $\mathbf{g} = \nabla K_3(s)$ is the hydraulic conductivity gradient assumed to be collinear with the direction $\mathbf{e}_2$. The BC involve constant head values $H_{3,0}$ and $H_{3,L}$ at the boundaries of the head domain. In this case the flow equation can be solved directly leading to

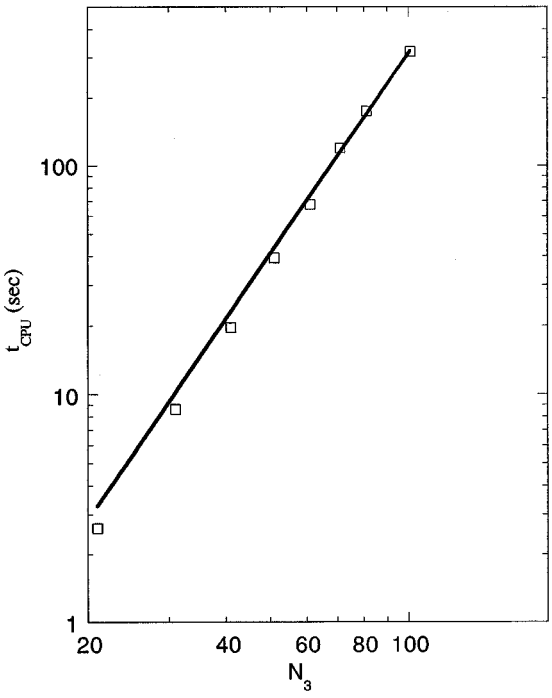

FIG. 14. *Plot of the CPU time (seconds) required for the ST solution of the flow equation on an HP* 9000/735 *workstation vs. the grid discretization.*

$$H_3(\mathbf{s}) = H_{3,0} - J_0 L \frac{\ln\left(1+\frac{gs_2}{K_0}\right) - \ln\left(1+\frac{gs_{2,0}}{K_0}\right)}{\ln\left(1+\frac{gs_{2,L}}{K_0}\right) - \ln\left(1+\frac{gs_{2,0}}{K_0}\right)}, \tag{4.11}$$

where $J_0 = -(H_{3,L} - H_{3,0})/L$ is the hydraulic head drop in the flow direction.

We solve the flow equation using the ST method assuming $K_0 = 1.0$ and gradient $g = 0.03$, within a cubic domain of length $L_K = 6$ centered at the origin of the Cartesian coordinate frame. Unless otherwise noted, the standard values of the ST parameters used in the simulations are $N_L = 14$ transformation lines, $N_3 = 41$ nodes per side (for both the hydraulic conductivity and head domains), $N_P = 41$ points per transformation line, and aspect ratio $L_K/L_H = 1$. In Fig. 8 we show the profile of the calculated hydraulic head in the direction of the applied gradient; the hydraulic head obtained with the ST solution is shown to offer an excellent approximation of the exact solution.

Next, we examine the accuracy of the ST solution with respect to the line discretization N_P, the number of the ST lines N_L, the grid discretization N_3, and the aspect ratio L_K/L_H. In order to provide a quantitative error estimate we use the normalized error parameter

$$E_3 = \frac{1}{N_3^3 |H_{3,L} - H_{3,0}|} \sum_{\mathbf{x} \in Grid} |H_3^*(\mathbf{x}) - H_3(\mathbf{x})|. \tag{4.12}$$

First, we test the sensitivity to the line discretization by varying N_P from 6 to 61 nodes per line, all other parameters being constant; the corresponding normalized

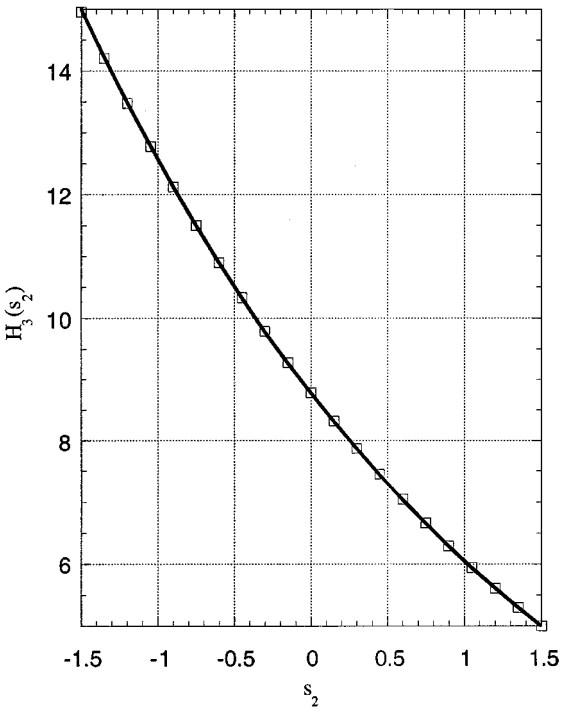

FIG. 15. *Hydraulic head profile vs. coordinate in the flow direction for exponential hydraulic conductivity. The exact solution (continuous line) and the ST estimate (squares) for the unit aspect ratio are shown.*

error E_3 is plotted in Fig. 9. We find an initial sharp decrease in the error which levels off to about $E_3 \cong 1.1\%$ for $N_P \cong 40$ (which is approximately equal to the number of nodes per side on the grid). In the following numerical experiments, we set the line discretization equal to the number of nodes per side of the grid.

Next we study the effect of the number of ST lines on the error. In Fig. 10 we plot the error with respect to the number of lines N_L for three different discretizations N_3 equal to 21, 41, and 81 lines. After a sharp initial decrease, the error levels off for N_L approximately equal to 14 and increases slightly for larger values of N_L. The magnitude of the error (only about 2%) signifies that excellent accuracy is obtained with approximately 14 uniformly oriented lines. The small increase in the error for larger number of lines N_L can be attributed to the solution of the BC system: the gain in accuracy that should result from the additional lines is canceled out by the error introduced when solving for a larger number of unknown boundary coefficients $\hat{J}_{1,\boldsymbol{\theta}_i}(0)$. In order to demonstrate this behavior quantitatively, we calculated the condition number of the system of equations involving the unknowns $\hat{J}_{1,\boldsymbol{\theta}_i}(0)$ and found that it increases with N_L, leading to large errors in the solution. This result was tested for different grid discretizations, and it was found that in all cases a number of $N_L = 14$ lines gives excellent results.

In Fig. 11 we show that the error decreases as the number of nodes increases. In Fig. 12 we plot the error E_3 as a function of the aspect ratio L_K/L_H; it is seen that E_3 decreases significantly as the aspect ratio increases from one to three. Tuning of the aspect ratio provides a means of improving the accuracy of the method at

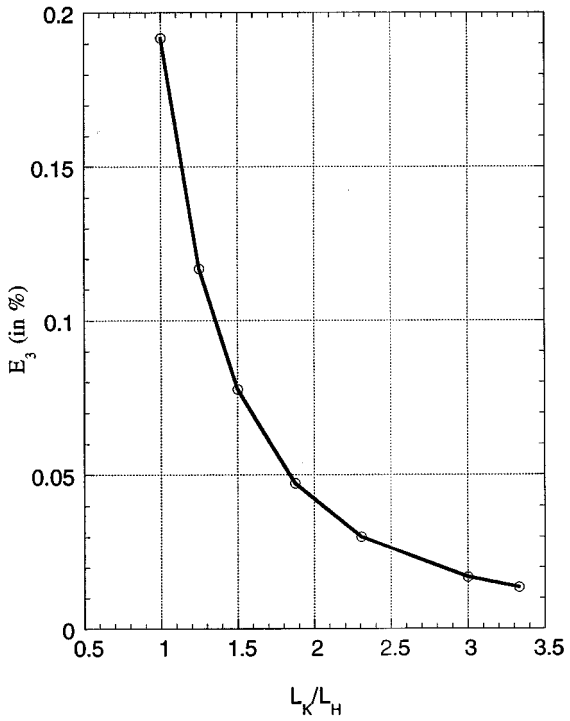

FIG. 16. *Plot of the normalized estimation error E_3 vs. the aspect ratio L_K/L_H in the case of exponential hydraulic conductivity.*

some computational cost, which is associated with the additional computational work required for the numerical estimation of the T_3^1 over a larger grid.

Finally we investigated the performance of the ST method as a flow equation solver. All the runs for this test were done on an HP 9000/735 workstation. Figure 13 shows the CPU time (in seconds) required to solve the flow equation vs the number of lines N_L. It is found that the CPU time scales almost linearly with the number of ST lines (the exponent obtained from the slope of the plot in Fig. 13 is equal to 0.92 ± 0.12). In Fig. 14 we plot the logarithm of the CPU time vs the discretization and find that the CPU times scales as $t_{CPU} \propto N_3^{\alpha}$, where $\alpha \cong 3.09 \pm 0.05$ within the range of grid sizes observed. Note that this result is in agreement with the observations made in section 3 above.

4.4. Flow for exponential hydraulic conductivity profile. Next, the flow equation (4.1) was solved in the special case of an exponential hydraulic conductivity field of the form

$$K_3(\mathbf{s}) = K_0 \exp\left(\frac{s_2}{g}\right). \tag{4.13}$$

In this case the flow equation can be solved directly leading to the following expression for the hydraulic head:

$$H_{\text{exact}}(\mathbf{s}) = H_{3,0} + (H_{3,L} - H_{3,0}) \frac{\exp\left(\frac{-s_{2,0}}{g}\right) - \exp\left(\frac{-s_2}{g}\right)}{\exp\left(\frac{-s_{2,0}}{g}\right) - \exp\left(\frac{-s_{2,L}}{g}\right)}. \tag{4.14}$$

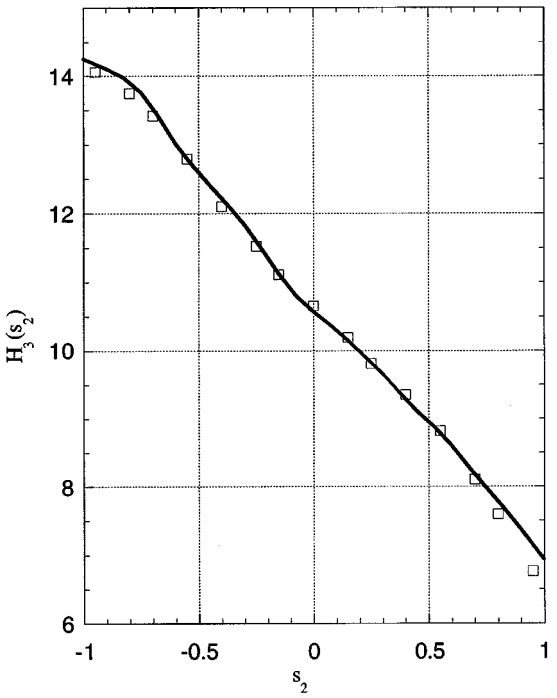

FIG. 17. *Hydraulic head profile vs. the coordinate in the flow direction for a unidimensional, correlated, log-normal hydraulic conductivity random field with* $\bar{K} = 17.3$ *and* $\sigma_K = 10$. *The solution obtained with the FDM (continuous line) and the ST estimates (squares) are shown.*

We now solve the flow equation by means of the ST method within a cubic domain of length $L_K = 3$ (as described in section 4.3 above), assuming $K_0 = 1$ and $g = 3$ in (4.13). Figure 15 shows profiles of the ST solution and the exact head solution in the longitudinal direction. The estimated hydraulic head reproduces the exact solution very accurately. The plot in Fig. 16 shows that the error is a rapidly decreasing function of the aspect ratio L_K/L_H.

4.5. Flow in a domain with random hydraulic conductivity. A correlated log-conductivity random field with unidimensional variability in the direction $\mathbf{e}_2$ is simulated. A Gaussian covariance

$$c_f(\boldsymbol{h}) = \sigma_f^2 \exp\left[-\frac{h_2^2}{\ell^2}\right] \tag{4.15}$$

was used for the log-conductivity random field, where the correlation length is assumed to be $\ell = 3$, and is generated using the sequential Gaussian simulation method (Deutsch and Journel (1993)). The hydraulic conductivity has mean $\bar{K} = 17.3$ and standard deviation $\sigma_K = 10$. We solved the flow equation with the standard BC by means of the ST method with the same discretization parameters as described in sections 4.3 and 4.4 above. In this case the exact solution of the flow equation is not known, hence we calculated a numerical solution using the finite difference method (FDM) for comparison purposes. In Fig. 17 we show the longitudinal profile of the hydraulic head calculated for an aspect ratio $L_K/L_H = 1.5$. We find that the solution obtained with the ST method reproduces adequately the results of the FDM. In

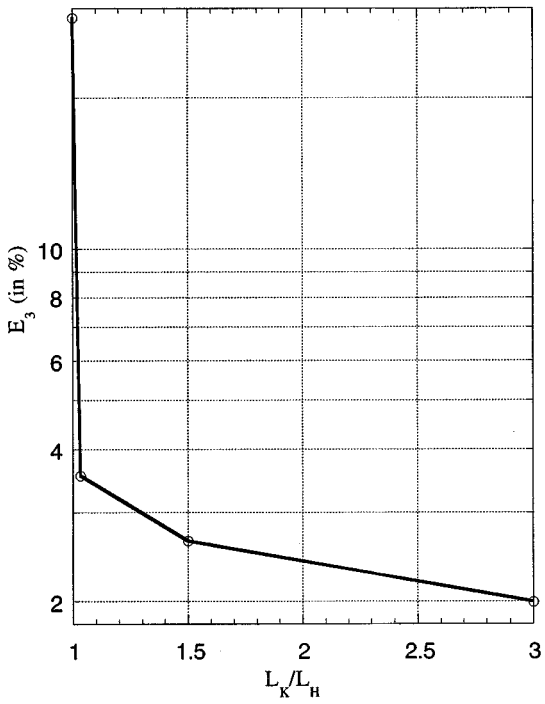

FIG. 18. *Plot of the normalized estimation error E_3 vs. the aspect ratio L_K/L_H for the random hydraulic conductivity field.*

Fig. 18 we plot the normalized difference between the ST method and the FDM. We find that as the ratio increases from one to three, the difference decreases to less than 2%. Therefore, in this case the ST method gives results compatible with the FDM. In order to investigate the numerical efficiency of the ST approach we compared the scaling of the CPU time required by the FDM and the ST methods as a function of the number of discretization nodes. The FDM was implemented using a centered second-order finite difference approximation to discretize the flow equation (Ababou et al. (1989)), in conjunction with the GMRES routine with Cholesky preconditioner from the MESCHACH (Stewart and Leyk, 1994) numerical library on linear algebra. In the case of the ST method we found that the CPU time, t_{CPU}, scales almost linearly with the number of nodes used in the discretization, i.e., $t_{CPU} \propto (N_3^3)^\beta$, with $\beta \cong 1.03$. For the FDM we obtained a scaling exponent $\beta \cong 1.36$. This indicates that the ST is an efficient method for solving large problems. A more detailed comparison requires analyzing how the accuracy of the ST method scales with the number of nodes. This point as well as further improvements in the accuracy of the ST method is considered in work that is currently in progress.

5. Conclusions. This paper is concerned with the numerical implementation of the ST procedure for solving multidimensional partial differential equations with space- and/or time-dependent coefficients proposed in Christakos and Hristopulos (1994, 1997). This has led to numerical codes that evaluate the ST T_3^1 and its inverse T_1^3. The numerical routines have been tested and shown to be accurate and efficient. Their efficiency can be further improved in future implementations by means of parallel calculation of the integrals on different transformation planes. The ST method

provides an efficient solver for the three-dimensional, steady-state flow equation by transforming it to one dimension without any loss of information. The inverse ST is implemented numerically and provides the three-dimensional hydraulic head in terms of a summation of the unidimensional head solutions along the directions of the transformation lines. The three-dimensional hydraulic head solution can be conditioned to honor physical BC and measurements of the hydraulic head within the domain. We tested the ST method for two special cases where the exact solution is available and showed that the method is accurate. We investigated the performance of the ST method and found that for a constant number of ST lines the CPU time scales almost linearly with the total number of nodes in the domain. In the case of a random hydraulic conductivity field the ST method gives results compatible with the finite difference technique. Future directions include the solution of flow problems with nonuniform flux, as well as explicit handling of the surface terms that arise at the domain boundaries.

Appendix I. In order to calculate the $T_3^1[f_3](p,\boldsymbol{\theta})$ of a function $f_3(\boldsymbol{s})$ with bounded support it is necessary to obtain the equations that represent the intersections of the support with the transformation planes. We assume here a cubic support whose sides have length L. Let us denote by $f_2(s_1,s_2)$ the functions in R^2 obtained from $f_3(\mathbf{s})$ by fixing the coordinate s_3, and let $\hat{f}_{1\boldsymbol{\phi}}(r) = T_2^1[f_2](r,\boldsymbol{\phi})$ denote the azimuthal ST obtained from the integral of $\hat{f}_2(s_1,s_2)$ along lines that are orthogonal to the azimuthal vector $\boldsymbol{\phi}$ and lie at distance r from the origin of the coordinate system. By allowing s_3 to vary again, we obtain the two-dimensional function $\hat{f}_{2\boldsymbol{\phi}}(r,s_3) = \hat{f}_{1\boldsymbol{\phi}}(r)|_{s_3}$. Finally, let us denote by $\hat{f}_{\boldsymbol{\phi},\boldsymbol{\chi}}(p) = T_2^1[\hat{f}_{2\boldsymbol{\phi}}](p,\boldsymbol{\chi})$ the ST of the above function with respect to the polar vector $\boldsymbol{\chi}$. It can be shown that the T_3^1 transformation can be expressed as

$$T_3^1[f_3](p,\boldsymbol{\theta}) = \hat{f}_{\boldsymbol{\phi},\boldsymbol{\chi}}(p) = T_2^1[T_2^1[f_2](r,\boldsymbol{\phi})](p,\boldsymbol{\chi}). \tag{I.1}$$

The T_3^1 transformation is well defined if the angles χ and ϕ are constrained within the interval $0 \le \phi, \chi \le \pi$, and the projection p is free to assume both negative and positive values. The ST $\hat{f}_{1\boldsymbol{\phi}}(r)$ is calculated in the rotated coordinate frame $\mathbf{z}^T = (z_1, z_2)$:

$$z^T = \begin{pmatrix} z_1 & z_2 \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}, \tag{I.2}$$

where the coordinate z_1 is equal to the projection of the vector $\mathbf{s}_a = (s_1, s_2)$ on the azimuthal vector. In the rotated system the azimuthal ST is expressed by

$$\hat{f}_{1\boldsymbol{\theta}}(r) = \int_{\zeta_1(r;\boldsymbol{\phi})}^{\zeta_2(r;\boldsymbol{\phi})} dz_2\, f_2(r,z_2). \tag{I.3}$$

The integration limits are determined by the cross sections of the ST lines with the domain boundaries in the transformed domain. Since the rotation transformation is linear, the following equations hold:

$$\zeta_1(r;\boldsymbol{\phi}) = La_1(\phi) + b_1(\phi)r \tag{I.4}$$

and

$$\zeta_2(r;\boldsymbol{\phi}) = La_2(\phi) + b_2(\phi)r. \tag{I.5}$$

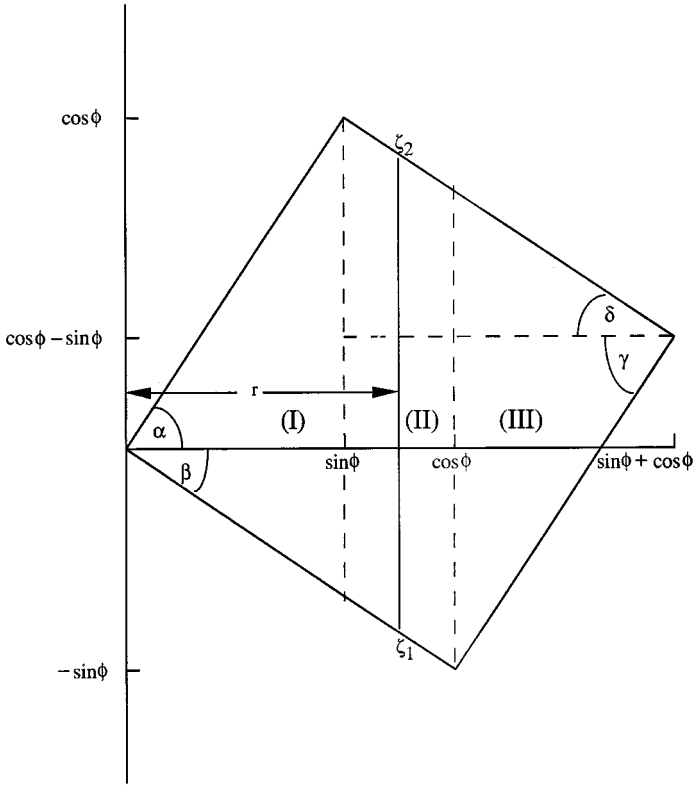

FIG. I.1. *A representation of the image of a square domain used in the azimuthal ST.*

Explicit expressions for the function $\zeta_1(r;\boldsymbol{\phi})$ and $\zeta_2(r;\boldsymbol{\phi})$ are obtained below. The next step involves the evaluation of the T_2^1 of the function $\hat{f}_{2\boldsymbol{\phi}}(r,s_3)$. This is accomplished best in the rotated frame

$$(y_1 \quad y_2) = \begin{pmatrix} \cos\chi & \sin\chi \\ -\sin\chi & \cos\chi \end{pmatrix} \begin{pmatrix} s_3 \\ r \end{pmatrix}, \tag{I.6}$$

where y_1 is the projection in the direction of the polar vector χ. The equation below is obtained:

$$\begin{aligned} T_3^1[f_3](p,\boldsymbol{\theta}) &= \int ds_3 \int dr_1 \hat{f}_{2\boldsymbol{\phi}}(r,s_3)\delta(p - r\sin\chi - s_3\cos\chi) \\ &= \int_{\psi_2(p,\boldsymbol{\theta})}^{\psi_1(p,\boldsymbol{\theta})} dy_2 \hat{f}_{2\boldsymbol{\phi}}(p,y_2). \end{aligned} \tag{I.7}$$

The integration limits $\psi_1(p,\boldsymbol{\theta})$ and $\psi_2(p,\boldsymbol{\theta})$ are given by the following linear expressions:

$$\psi_1(p,\boldsymbol{\theta}) = Lc_1(\boldsymbol{\theta}) + pd_1(\boldsymbol{\theta}) \tag{I.8}$$

and

$$\psi_2(p,\boldsymbol{\theta}) = Lc_2(\boldsymbol{\theta}) + pd_2(\boldsymbol{\theta}). \tag{I.9}$$

TABLE I.1

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$0 \le \phi < \frac{\pi}{2}$	$a_1(\phi)$	$b_1(\phi)$
$0 \le r < L\cos\phi$	0	$-\tan\phi$
$L\cos\phi \le r \le L_\phi$	$-(\sin\phi)^{-1}$	$\cot\phi$

TABLE I.2

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$0 \le \phi < \frac{\pi}{2}$	$a_2(\phi)$	$b_2(\phi)$
$0 \le r < L\sin\phi$	0	$\cot\phi$
$L\sin\phi \le r \le L_\phi$	$(\cos\phi)^{-1}$	$-\tan\phi$

TABLE I.3

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$\frac{\pi}{2} \le \phi < \pi$	$a_1(\phi)$	$b_1(\phi)$
$L\cos\phi \le r < L_\phi$	$-(\sin\phi)^{-1}$	$\cot\phi$
$L_\phi \le r \le L\sin\phi$	$(\cos\phi)$	$-\tan\phi$

TABLE I.4

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$\frac{\pi}{2} \le \phi < \pi$	$a_2(\phi)$	$b_2(\phi)$
$L\cos\phi \le r < 0$	0	$-\tan\phi$
$0 \le r < L\sin\phi$	0	$\cot\phi$

TABLE I.5

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$0 \le \chi < \frac{\pi}{2}, 0 \le \phi \le \frac{\pi}{2}$	$c_1(\boldsymbol{\theta})$	$d_1(\boldsymbol{\theta})$
$0 \le p \le L\cos\chi$	0	$-\tan\chi$
$L\cos\chi < p \le L_\phi\sin\chi + L\cos\chi$	$-(\sin\chi)^{-1}$	$\cot\chi$

TABLE I.6

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$0 \le \chi < \frac{\pi}{2}, 0 \le \phi \le \frac{\pi}{2}$	$c_2(\boldsymbol{\theta})$	$d_2(\boldsymbol{\theta})$
$0 \le p \le L_\phi\sin\chi$	0	$\cot\chi$
$L_\phi\sin\chi < p \le L_\phi\sin\chi + L\cos\chi$	$\frac{\cos\phi + \sin\phi}{\cos\chi}$	$-\tan\chi$

TABLE I.7

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$0 \le \chi < \frac{\pi}{2}, \frac{\pi}{2} \le \phi < \pi$	$c_1(\boldsymbol{\theta})$	$d_1(\boldsymbol{\theta})$
$L\cos\phi\sin\chi \le p \le L\cos\phi\sin\chi + L\cos\chi$	$\cos\phi(\cos\chi)^{-1}$	$-\tan\chi$
$L\cos\phi\sin\chi + L\cos\chi < p \le L\sin\phi\sin\chi + L\cos\chi$	$-(\sin\chi)^{-1}$	$\cot\chi$

TABLE I.8

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$0 \le \chi < \frac{\pi}{2}, \frac{\pi}{2} \le \phi < \pi$	$c_2(\boldsymbol{\theta})$	$d_2(\boldsymbol{\theta})$
$L\cos\phi\sin\chi \le p \le L\sin\phi\sin\chi$	0	$\cot\chi$
$L\sin\phi\sin\chi < p \le L\sin\phi\sin\chi + L\cos\chi$	$\sin\phi(\cos\chi)^{-1}$	$-\tan\chi$

TABLE I.9

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$\frac{\pi}{2} \le \chi \le \pi, 0 \le \phi \le \frac{\pi}{2}$	$c_1(\boldsymbol{\theta})$	$d_1(\boldsymbol{\theta})$
$L\cos\chi \le p < L_\phi\sin\chi + L\cos\chi$	$-(\sin\chi)^{-1}$	$\cot\chi$
$L_\phi\sin\chi + L\cos\chi \le p < L\phi\sin\chi$	$\dfrac{\cos\phi + \sin\phi}{\cos\chi}$	$-\tan\chi$

TABLE I.10

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$\frac{\pi}{2} \le \chi \le \pi, 0 \le \phi \le \frac{\pi}{2}$	$c_2(\boldsymbol{\theta})$	$d_2(\boldsymbol{\theta})$
$L\cos\chi \le p < 0$	0	$-\tan\chi$
$0 \le p < L_\phi\sin\chi$	0	$\cot\chi$

TABLE I.11

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$\frac{\pi}{2} \le \chi \le \pi, \frac{\pi}{2} \le \phi \le \pi$	$c_1(\boldsymbol{\theta})$	$d_1(\boldsymbol{\theta})$
$L\cos\chi + L\cos\phi\sin\chi \le p < L\sin\phi\sin\chi + L\cos\chi$	$-(\sin\chi)^{-1}$	$\cot\chi$
$L\sin\phi\sin\chi + L\cos\chi \le p < L\sin\phi\sin\chi$	$\sin\phi(\cos\chi)^{-1}$	$-\tan\chi$

TABLE I.12

Coefficients of the integration limits for the azimuthal and polar ST (see (I.4), (I.5), (I.8), *and* (I.9)), $L_\phi = L(\cos\phi + \sin\phi)$.

$\frac{\pi}{2} \le \chi \le \pi, \frac{\pi}{2} \le \phi \le \pi$	$c_2(\boldsymbol{\theta})$	$d_2(\boldsymbol{\theta})$
$L\cos\chi + L\cos\phi\sin\chi \le p < L\cos\phi\sin\chi$	$\cos\phi(\cos\chi)^{-1}$	$-\tan\chi$
$L\cos\phi\sin\chi \le p < L\sin\phi\sin\chi$	0	$\cot\chi$

The integration limits of the two-dimensional transformations can be determined as follows. First, the positions of the two-dimensional domain vertices are determined in the nonrotated frame. Next, the images of the vertices under the relevant rotation transformation (by ϕ or χ), and the equations of the lines (domain edges) that connect the vertices are obtained. Fig. I.1 represents the image of a square domain used in the azimuthal ST; the lower and upper integration limits are given by distinct expressions within three different zones, denoted by (I), (II), and (III). In Tables I.1 through I.12 below, we give the analytical expressions for the boundary functions $a_i(\phi), b_i(\phi), c_i(\boldsymbol{\theta})$, and $d_i(\boldsymbol{\theta})$, where $i = 1, 2$. The form of these coefficients is general for cubic supports and does not depend on the transformed function.

Appendix II. Here we calculate the exponent of the unidimensional solution (4.2) using the uniform-gradient approximation. The general expression for the exponent is given by Christakos and Hristopulos (1994):

$$\tilde{\kappa}_{\boldsymbol{\theta}}(p) = \frac{T_3^1 \left[\boldsymbol{\theta} \cdot \nabla \ln K_3(\mathbf{z}) \dfrac{\partial H_3(z)}{\partial z_1} \right] (p, \boldsymbol{\theta})}{T_3^1 \left[\frac{\partial H_3(\mathbf{z})}{\partial z_1} \right] (p, \boldsymbol{\theta})}, \tag{II.1}$$

where $\boldsymbol{z}$ denotes the position vector in the rotated coordinate frame

$$\mathbf{z}^T = \mathbf{R}(\boldsymbol{\theta}) \cdot \mathbf{s}, \tag{II.2}$$

and $\mathbf{R}(\boldsymbol{\theta})$ is the rotation matrix given in terms of the direction cosines as

$$\mathbf{R}(\boldsymbol{\theta}) = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ -\theta_1\theta_2/q & q & -\theta_2\theta_3/q \\ -\theta_3/q & 0 & \theta_1/q \end{bmatrix}. \tag{II.3}$$

The parameters of the rotation matrix above are

$$\boldsymbol{\theta}^T = (\theta_1, \theta_2, \theta_3) = (\sin\chi\cos\phi, \sin\chi\cos\phi, \cos\chi) \tag{II.4}$$

and

$$q = \sqrt{\theta_1^2 + \theta_3^2}. \tag{II.5}$$

In the uniform hydraulic gradient approximation we assume a linear variation of the hydraulic head to leading order, i.e.,

$$H_3(\mathbf{s}) = H_0 + \mathbf{J} \cdot \mathbf{s}. \tag{II.6}$$

The partial derivative of the head $\frac{\partial H_3(\mathbf{z})}{\partial z_1}$ in the rotated coordinate frame can be expressed as

$$\frac{\partial H_3(\mathbf{z})}{\partial z_1} = \sum_{i=1}^{3} \frac{\partial H_3(s)}{\partial s_i} \frac{\partial s_i}{\partial z_1} = \mathbf{J} \cdot \boldsymbol{\theta}. \tag{II.7}$$

In the above we made use of the orthogonality of the transformation $\mathbf{R}(\boldsymbol{\theta})$ to obtain $\frac{\partial s_i}{\partial z_1} = R_{i1}^{-1} = R_{1i}$. In light of (II.7), the general expression (II.1) is reduced to

$$\tilde{\kappa}_{\boldsymbol{\theta}}(p) = \frac{T_3^1[\boldsymbol{\theta} \cdot \nabla \ln K_3(\mathbf{z})](p, \boldsymbol{\theta})}{A(p, \boldsymbol{\theta})}, \tag{II.8}$$

where $A(p, \boldsymbol{\theta})$ denotes the area that corresponds to the cross section of the transformation plane and the flow domain.

REFERENCES

R. Ababou, D. McLaughlin, A. L. Gutjahr, and A. F. B. Tompson (1989), *Numerical simulation of three-dimensional saturated flow in randomly heterogeneous porous media*, Transport in Porous Media, 4, pp. 549–565.

M. Abramowitz and I. A. Stegun (1970), *Handbook of Mathematical Functions*, Dover, New York.

A. A. Bakr, L. W. Gelhar, A. L. Gutjahr, and J. R. MacMillan (1978), *Stochastic analysis of spatial variability in subsurface flows:* 1. *Comparison of one-dimensional and three-dimensional flow*, Water Resources Research, 14, pp. 263–271.

G. Christakos (1984a), *The space transformations and their applications in systems modelling and simulation*, in Proc. 12th Intern. Confer. on Modelling and Simulation (AMSE), 1(3), Athens, Greece, pp. 49–68.

G. Christakos (1984b), *On the problem of permissible covariance and variogram models*, Water Resources Research, 20, pp. 251–265.

G. Christakos (1987), *Stochastic stimulation of spatially correlated geo-processes*, Math. Geol., 19, pp. 807–831.

G. Christakos (1992), *Random Field Models in Earth Sciences*, Academic Press, San Diego, CA.

G. Christakos and D. T. Hristopulos (1994), *Stochastic space transformations in subsurface hydrology—Part* 2: *Generalized spectral decompositions and Plancherel representations*, Stochastic Hydrology and Hydraulic, 8, pp. 117–138.

G. Christakos and D. T. Hristopulos (1997), *Stochastic Radon operators in porous media hydrodynamics*, Quart. Appl. Math., LV, pp. 89–112.

G. Christakos and C. Panagopoulos (1992), *Space transformation methods in the representation of geophysical random fields*, IEEE Trans. Geosciences and Remote Sensing, 30, pp. 55–70.

J. H. Cushman, ed. (1990), *Dynamics of Fluids in Porous Media*, Academic Press, San Diego, CA.

G. Dagan (1979), *Models of groundwater flow in statistically homogeneous porous formations*, Water Resources Research, 15, pp. 47–63.

S. R. Deans (1993), *The Radon Transform and Some of Its Applications*, Krieger, Malabar, FL.

G. de Marsily (1986), *Quantitative Hydrogeology*, Academic Press, San Diego, CA.

C. V. Deutsch and A. G. Journel (1993), *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press, New York.

I. M. Gel'fand and G. E. Shilov (1964), *Generalized Functions*, Vol. 1, Academic Press, New York.

S. Helgason (1980), *The Radon Transform*, Birkhäuser-Verlag, Boston, Basel, Stuttgart.

A. K. Jain (1989), *Fundamentals of Digital Image Processing*, Prentice–Hall, Englewood Cliffs, NJ.

F. John (1955), *Plane Waves and Spherical Means*, Springer-Verlag, New York.

A. Louis (1986), *Incomplete data problems in X-ray computerized tomography*, Numer. Math., 48, pp. 251–262.

R. L. Naff and A. V. Vecchia (1986), *Stochastic analysis of three-dimensional flow in a bounded domain*, Water Resources Research, 22, pp. 695–704.

S. P. Neuman (1982), *Statistical characterization of aquifer heterogeneities: An overview*, Recent Trends in Hydrogeology, Spec. Papers Geol. Soc. Amer., 189, pp. 81–102.

NRC-National Research Council (1990), *Ground Water Models: Scientific and Regulatory Applications*, National Academy Press, Washington DC.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1989), *Numerical Recipes*, Cambridge University Press, Cambridge, UK.

D. E. Stewart and Z. Leyk (1994), *Meschach: Matrix Computations in C*, Centre for Mathematics and its Applications, School of Mathematical Sciences, Australian National University, Canberra, Australia.

H. A. Tchelepi (1994), *Viscous Fingering, Gravity Segregation and Permeability Heterogeneity in Two-Dimensional and Three-Dimensional Flows*, Ph.D. dissertation, Stanford University, 1994.

H. A. Tchelepi and F. M. Orr, Jr. (1994), *The interaction of viscous fingering, permeability heterogeneity and gravity segregation in 3D*, SPE Reservoir Engineering.

A. N. Tikhonov and A. V. Goncharsky (1987), *Ill-Posed Problems in the Natural Sciences*, Mir Publishers, Moscow.

A. F. B. Tompson, R. Ababou, and L. W. Gelhar (1989), *Implementation of the three-dimensional turning bands random field generator*, Water Resources Research, 25, pp. 2227–2243.