SUPERCOMPUTERS AND THEIR USE IN MODELING SUBSURFACE SOLUTE TRANSPORT

D. A. Barry
Centre for Water Research
University of Western Australia, Nedlands

Abstract. Supercomputers offer a wide range of possibilities for advancing knowledge and solving difficult, real-world problems. Solute transport–related theory and applications have long been an area of specialization within hydrology. In this review, recent applications of supercomputers to solute transport problems are presented. These applications are based mostly on the assumption of a dilute solute, allowing the governing equations to be decoupled. An application of supercomputers is the simulation of solute transport in large heterogeneous spatial domains by both direct solution of the governing equations and by particle-tracking methods. An impetus for this research has been the rapid increase in predictive dispersion models based on analytical stochastic theory. Up until now, relatively few applications have incorporated algorithms designed with a targeted supercomputer in mind, with the result that the machine's potential cannot be exploited fully. Since many numerical problems are reduced to solutions of algebraic systems when they are implemented on computers, algorithms for linear algebraic systems of equations suitable for vector and parallel machines are discussed in some detail. Data analysis, necessary both for evaluating emerging theories and for more "routine" applications like determining model parameters, is identified as another general area that is ably handled by supercomputers. In particular, supercomputers offer a way to become less dependent on parametric assumptions and constraints, substituting, instead, raw computational power. Visualization techniques and associated facilities have flourished alongside the development of supercomputer centers. These offer the possibility for real-time viewing of solute transport processes, both real and simulated, although little has been produced thus far. It is expected that many more researchers involved with subsurface solute transport will make use of supercomputers in the future.

INTRODUCTION

Hydrological uses of computers are increasing monotonically with the passage of time. We are now into the third phase of the "computer revolution" where "problems are being routinely defined that could not even have been conceived without the existence of computers" [*Wallis*, 1987]. Certainly, available supercomputers continue on this "revolutionary" path. There are two rather obvious trends in the use of supercomputers. First, the size of problems which becomes feasible continues to increase in magnitude. Here the word "size" refers not only to computer-related limitations, e.g., speed or number of processors, amount of memory, input/output and physical storage capacity, but also to the complexity level of the models investigated. Second, smaller, more routine analyses are carried out much more rapidly and accurately, sometimes in real time.

Supercomputers embody the notion of speed. Unfortunately, it can be difficult to realize high rates of computation. There are two basic paths by which computer calculational rates can be improved. In the first, rates are increased by speeding up the passage of information in the machine and the response time of the processors. Theoretically, information cannot be transmitted faster than the speed of light. Superconductors offer a possible way to increase presently available transmission rates. Reducing the distances between the machine components is another, more obvious method of speeding up information transfer. The circular architecture of the Cray X-MP illustrates an approach of this type. This and other limitations are discussed by *Seitz and Matisoo* [1984], who indicate that the limits of the available technology may be in sight.

The second way to increase computational rates is to perform more calculations at the same instant. Vector machines achieve their speed by operating on vectors rather than scalars, whereas parallel machines increase the number of processors in the computer. With this simple statement it is apparent that, on the one hand, a vector

8755-1209/90/90RG-00342 *$05.00*

Reviews of Geophysics, 28, 3 / *August 1990*
pages 277– 295
Paper number 90RG00342

machine needs to operate on vectors most of the time to achieve speeds close to its rated maximum. That is, the degree to which an algorithm will vectorize determines its suitablity for running on a vector processor. On the other hand, parallel machines achieve the same result by completing many computational tasks simultaneously on different processors. The combination of an array of vector processors produces a hybrid, vector-parallel machine. For a given algorithm, then, the degree of parallelism, or concurrency [*Sharp*, 1987], achievable in its implementation is the overriding factor in determining its computation rate. So far, only "course-grained" parallelism has been referred to. The operations to be performed, such as matrix multiplication, can be separated into independent component parts, each of which is carried out independently, "in parallel." Often, the instructions to a machine central processing unit (CPU) (e.g., loads, stores, adds) can also be carried out in parallel, yielding "fine-grained" parallelism. Very long instruction word (VLIW) machines [*Fisher*, 1984] overlap processor instructions that can be carried out in parallel with substantial increases in computation rate. The advantage is that VLIW machines can operate in scalar mode, relying on a sophisticated compiler to produce fine-grained parallelism in object code. There are, apparently, no VLIW machines that are supercomputers, although minisupercomputers of this type are available.

Simply transferring codes from serial to vector or parallel machines may not result in very impressive speedups. This point is demonstrated clearly by *Pelka and Peters* [1986], who compare computation rates for scalar and vectorized finite element solutions of a groundwater flow model. Particularly for parallel machines, the algorithms used may turn out to be not very efficient. Not surprisingly, algorithms favored on sequential machines tend to be those that minimize the number of arithmetic operations necessary to complete the task at hand. Vector processors carry out pipelined operations. Pipelining refers to the process by which the arithmetic operations are broken down into a number of elementary subcomponents. Each subcomponent operation depends on the results from the previous subcomponent. By operating on vectors, the calculations on each pair of operands can be performed simultaneously in assembly line fashion, hence the term pipeline. The nature of the pipelining is such that for some time no results are available while the necessary subcomponent operations are completed for the first time. This is known as the start-up time. It is clear that short vectors should be avoided as much as possible, since start-up time is incurred for every new pair of vectors operated upon. The effect of the degree of vectorization of an algorithm, as given by Amdahl's law [*Amdahl*, 1976; *Emmen*, 1987], is presented in Figure 1, where it is shown that the relative (or actual) performance is a function of the fraction of the algorithm that vectorizes. The curve labels list possible speedup, i.e., the ratio of possible vector speed to scalar speed on the computer. A machine that has a possible speedup of 50 but vectorizes only 90% of the code will achieve a speedup of only 8.5. *Denning* [1988] discusses Amdahl's law as it applies to parallel processors.

Parallel machines are affected by factors apart from simply the total number of operations. Both the time taken to communicate between processors and the synchronization of the computational tasks need to be considered. *Adams and Crockett* [1984] present methods to quantify these timing costs using a linear system example. More generally, on parallel machines there are two ways by which speedup can be enhanced. First, the problem is broken into independent subproblems. Monte Carlo simulations provide an immediate example suited to this treatment. Other examples include matrix multiplication or scalar products, both of which can be parallelized. Second, the algorithm can be reordered so that the degree of parallelism increases.

This brief introduction indicates that realizing the potential performance of a supercomputer is often more than simply loading and running an existing computer code. The characteristics of the machine being used dictate the methods by which substantial speedups can be obtained. Supercomputers can be used to provide very substantial gains in solute transport modeling capability and concomitant understanding of the relative importance of the processes involved. Problems involving large-scale, heterogeneous domains and complex chemical reactions become open to analysis. However, the time taken to run a problem is always a limiting factor to some degree. Therefore the design of a computer code is worthy of serious consideration. General methods by which supercomputers can be utilized efficiently are one focus of this review. An overview of transport theory for a single solute species in a density-dependent flow is presented first, followed by numerical approaches to solving the

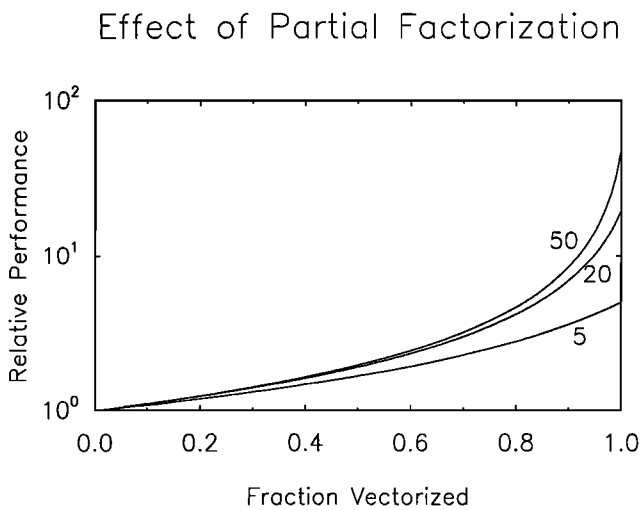

Figure 1. Relative performance, or speedup, *R/S*, plotted on a logarithmic scale as a function of the fraction of the code vectorized. *R* is the realized speed and *S* is the machine's scalar speed. Three hypothetical vector processors are considered, with the curve labels corresponding to the value of *R/S* for fully vectorized code.

governing equations. It is shown that solute transport models are certainly candidates for numerical solution using supercomputers. The foundation of these numerical solutions is solving linear systems of equations in an efficient manner. Modern algorithms developed for vector and parallel machines are presented. Data analysis constitutes another area where supercomputers can be of great benefit. A summary of conceptually simple but highly computationally intensive, nonparametric estimation procedures is included. The number of studies concerning various aspects of solute transport that have used supercomputers is relatively limited. There is little doubt that supercomputer-based research endeavors will increase. Even considering the existing research, it is apparent that very little effort has been devoted to "pushing the envelope" of supercomputer performance in solute transport applications.

THEORY OF MISCIBLE FLUID FLOW

Solute transport problems must always be considered in the context of a particular scale, e.g., a catchment basin, part of a basin, a geological structure, or a laboratory apparatus. Many researchers have developed theories applicable to the macroscale, defined as being much larger than the microscale or pore scale [e.g., *Gelhar et al.*, 1979; *Chu and Sposito*, 1980, 1981; *Gelhar and Axness*, 1983; *Winter et al.*, 1984; *Dagan*, 1982, 1984, 1987]. These foundational theories are not directly relevant to supercomputer applications except inasmuch as they are necessary to provide the governing models which must be solved by some means. It is worth noting, however, that the approaches used in deriving these theories are open to debate. For example, *Sposito et al.* [1979] identified and reviewed three approaches used to derive foundational theories leading to the commonly used macroscopic solute transport equation applicable in an isothermal, saturated porous medium:

$$\partial c/\partial t = \nabla\cdot(\mathbf{D}'\cdot\nabla c) - \mathbf{v}\cdot\nabla c \qquad (1)$$

where t is time, $\mathbf{D}'$ is the solute dispersion tensor, $\mathbf{v}$ is the solute velocity, and c is the solute concentration in solution. Equation (1) is just the familiar convection-dispersion equation (CDE). The three approaches, which are based explicitly on, or are tantamount to, averaging operations applied at the microscale, are all subject to either mathematical or physical assumptions which have yet to be proved rigorously [*Sposito et al*, 1979].

Under appropriate circumstances, foundational theories produce macroscopic equations like (1) that, with some assumptions, are applicable in practice. Typically, a number of coupled, nonlinear equations are reduced to a few, possibly decoupled linear equations using simplifying assumptions. It is apparent, however, that complex field applications could require numerical models describing nonisothermal, multiphase, multispecies, density-dependent solute transport in variably saturated heterogeneous porous media. The parameterization requirements of such a model could well exceed available knowledge of the necessary constitutive or chemical relationships. Even with adequate resolution of these issues, the numerical solution of such a model is a significant task requiring, certainly, a supercomputer for results to be obtained in a reasonable time scale, if at all. There have been few attempts to develop general codes for these extreme, but relevant, conditions.

The density of subsurface water is very often nearly constant, or at least it is assumed to be so. The density varies with changes in temperature, pressure, or presence of dissolved chemicals. Density changes can become noticeable if these quantities undergo large variations. It is possible for dissolved solutes to produce extreme density changes. In some countries, waste disposal in salt formations is under consideration. In such areas, and in the freshwater/seawater interfaces of coastal regions, the density dependence of the fluid components must be considered in any modeling of the system [*Huyakorn et al.*, 1987; *Hassanizadeh and Leijnse*, 1988; *Herbert et al.*, 1988].

In groundwater applications, density effects of thermal variations are typically ignored. Likewise, the problem is simplified further if only nondeformable media are considered. Both of these effects will be important in some cases, and both are the subject of readily available treatments. For example, the problem of heat and mass transport is discussed by *Bear* [1972, section 10.7.2]. Thermal effects cannot be ignored, however, if geologic time scales are considered. *Bethke* [1985] and *Bethke et al.* [1988] model reactive chemical transport in sedimentary basins over long time scales making use of a machine with a novel parallel-vector structure [*Kuck et al.*, 1986].

A thorough derivation of the governing equations for a single solute species in a rigid, saturated medium is given by *Hassanizadeh and Leijnse* [1988]. With suitable assumptions they recover the familiar forms:

$$n\frac{\partial \rho}{\partial t} = \nabla\cdot\rho\frac{\mathbf{k}}{\mu}\cdot(\nabla p - \rho\mathbf{g}) \qquad (2a)$$

$$n\frac{\partial \omega}{\partial t} - \frac{\mathbf{k}}{\mu}\cdot(\nabla p - \rho\mathbf{g})\cdot\nabla\omega = \nabla\cdot(\mathbf{D}\cdot\nabla\omega) \qquad (2b)$$

In obtaining these equations, *Hassanizadeh and Leijnse* [1988] have assumed the presence of a rigid porous matrix and isothermal conditions. No production or removal of solute is accounted for. The density of the fluid is dependent on both its solute mass fraction ω and fluid pressure p. A suitable relationship between these quantities is given by the density state equation [*Bear and Verruijt*, 1987, section 3.2; *Hassanizadeh and Leijnse*, 1988]:

$$\rho(\omega, p)/\rho_0 = \exp[\gamma\omega - \beta(p - p_0)] \qquad (3)$$

The coefficients γ and β in this formula are best considered as fitting parameters which attain specific values for given solutes. The dynamic viscosity of the fluid, μ, will change with ω as well:

$$\mu = m(\omega)\mu_0 \tag{4}$$

where $m(\omega)$ is a function to be specified for the specific solute. If μ_0 is defined to be the reference dynamic viscosity when $\omega = 0$, then $m(0) = 1$.

The ease with which numerical solutions of the nonlinear system (2) are obtained depends, among other things, on the dimensionality and extent of the spatial domain and the degree of variability of the coefficients. These factors alone, in the context of a practical simulation, make the use of a supercomputer desirable. If the system is further complicated by, for example, a nonlinear chemical reaction or the coupled transport of a number of solutes, then the need for a supercomputer becomes more advantageous.

To show that (2) reduces to (1) as a special case, we consider the definitions of concentration and mass fraction. Solute concentration c is equivalent to M_S/V_T, and the mass fraction ω is equivalent to M_S/M_T. Thus $c/\omega = M_T/V_T \equiv \rho$, or $\omega = c/\rho$. Figure 1 of *Herbert et al.* [1988] shows that, for a solution containing up to 26% of salt by mass,

$$\rho/\rho_0 \approx 1 + 0.2\omega \qquad 0 \le \omega \le 1 \tag{3'}$$

For $\omega \ll 1$ (dilute solutions), $\omega \approx c$ and $\mu \approx \mu_0$. Under steady flow conditions, (2*b*) then reduces, approximately, to (1). An alternative expression to (3) or (3') for the solute density state equation is available when the volumes of the components in the final solution are additive [*Herbert et al.*, 1988]:

$$\rho/\rho_0 = \rho^s/[(1 - \omega)\rho^s + \omega\rho_0] \tag{3''}$$

Herbert et al. [1988, Figure 1] show (3") is likely to be a good approximation in practical cases. However, it should be noted that the dependence of fluid density on pressure has been neglected in (3') and (3").

The governing equations can be manipulated into forms suitable for numerical solution [e.g., *Huyakorn et al.*, 1987, Appendix A; *Hassanizadeh and Leijnse*, 1988]. It is very common to ignore the coupling between (2*a*) and (2*b*) for dilute solutions. The question of how dilute the solution has to be before this assumption is valid is not easily answered. Even for very dilute solutions moving in the subsurface environment this assumption may need to be considered carefully, particularly in aquifers. *Sudicky et al.* [1983] and *Freyberg* [1986] analyzed nonreactive solute transport experiments at the Borden site. These experiments tracked the movement of plumes in a sandy, unconfined aquifer [*MacFarlane et al.*, 1983]. In both experiments the solute mass fraction of the injected fluid was very small ($< 5 \times 10^{-3}$), but appreciable vertical plume movement (approximately 1 m in 120 days) was observed. Identified as possible causes for this downward movement were the presence of a vertical component in the local groundwater velocity field due, perhaps, to aquifer recharge or, second, density differences between the injected plume and the native groundwater. The relative magnitude of these effects is uncertain at this time.

Stochastic Solute Transport

Recent, comprehensive reviews of the stochastic approaches to water flow and solute transport in groundwater have been published by *Dagan* [1986] and *Gelhar* [1986]. The starting point is the decoupled form of the water flow and solute transport equations, meaning that a dilute solute is considered. The logarithm of the hydraulic conductivity, ln (K), is treated as a random spatial function with known constant mean and spatially stationary covariance function. Stationarity refers to the assumption that the covariance function is unchanged by spatial translation. The low-order statistics of the piezometric head (i.e., fluid potential) and flow field can then be calculated. With the assumption of a constant porosity the statistics of the groundwater velocity field follow directly. The stochastic solute transport equation is just (1) except that v is now a random spatial function with known low-order statistical properties. The solute concentration c is the dependent variable in (1), and hence it is a random function as well. Theoretical analyses have focused on ensemble averaging, i.e., averaging over all possible realizations of the velocity field in (1) to derive a governing equation of $\langle c \rangle$, the ensemble mean concentration [*Gelhar and Axness*, 1983; *Dagan*, 1984, 1987, 1988; *Neuman et al.*, 1987]. Theories based on the stochastic CDE essentially treat the solute drift velocity as a random variable with assumed low-order statistical properties. They are, however, not without fundamental problems, as discussed by *Sposito et al.* [1986]. These authors noted that each aquifer represents a single realization of the stochastic process under consideration, whereas the results of a stochastic analysis represent the statistics for the ensemble of realizations. The results of the stochastic CDE approaches, therefore, are not applicable to a particular aquifer unless additional assumptions are made [*Dagan*, 1984]. Numerical simulations reinforcing this point are presented below.

The ensemble mean concentration statisfies, approximately, a macroscale transport equation similar to (1), except that the dispersion coefficient components are now functions of time. For a coordinate system aligned with the mean flow direction (axis 1), *Sposito and Barry* [1987] derived the macrodispersion coefficients

$$D_{Mii}(t) = \int_0^t \int S_{ii}(\kappa, t') \exp\left(-\sum_i D_{ii}\kappa_i^2\right) \cos(v\kappa_1 t')\, d\kappa\, dt' \tag{5}$$

where $S_{ii}(\kappa, t)$ is the Fourier transform of the stationary velocity covariance function and D_{ii} are laboratory scale

dispersion coefficients. Once the function S_{ii} is specified, the macroscale dispersion coefficients can be evaluated without difficulty.

A major advantage of the theory lies in its practical applicability when some basic conditions are satisfied. For example, the authors cited above assume the presence of a large-scale velocity field with a uniform, constant mean. Because the theory quantifies the effect of the local-scale velocity variations on solute dispersion, one needs "only" to estimate the low-order statistics of the hydraulic conductivity field in order to predict macroscale dispersivity, and thus spread, of large-scale solute plumes. Various researchers have investigated the value of theoretical predictions in the light of field experimental data. The main body of analytical results derived using stochastic theory applies only for ideal tracers under assumptions that may not be widely applicable. The degree to which the assumptions can be relaxed in practice needs to be clearly defined. Second, the effect of chemical and biological reactions at the field scale is difficult to determine analytically. Some of these issues have been investigated using supercomputer simulations, as discussed below.

Numerical Solutions

Fully three-dimensional, transient numerical solutions of the complete governing solute transport equations have not been the subject of many studies as the computational load is very large. Saltwater intrusion poses a serious threat to groundwater supplies in many coastal areas [*Newport*, 1977], and this has prompted modeling efforts, many of which have been based on the "sharp-interface" assumption. A variety of numerical solutions have been presented by *Lee and Cheng* [1974], *Segol et al.* [1975], *Segol and Pinder* [1976], *Huyakorn and Taylor* [1977], *Taylor and Huyakorn* [1978], *Frind* [1982*a*, *b*], and *Huyakorn et al.* [1987]. The Galerkin approach is used typically. For example, *Huyakorn et al.* [1987] consider the governing system

$$\nabla\cdot[\mathbf{K}\cdot(\nabla\phi + \frac{\eta \mathbf{g}}{g}c)] = S_s\frac{\partial\phi}{\partial t} + n\eta\frac{\partial c}{\partial t} - \frac{\rho q}{\rho_w} \qquad (6a)$$

$$\nabla\cdot(\mathbf{D}\cdot\nabla c) - \mathbf{v}\cdot\nabla c = n\frac{\partial c}{\partial t} + q(c - c_0) \qquad (6b)$$

$$\rho/\rho_0 = 1 + \varepsilon c/c_s \qquad (6c)$$

Application of the Galerkin technique for these equations yields [*Huyakorn et al.*, 1987]

$$\mathbf{A}_{IJ}\phi_J + \mathbf{B}_{IJ}\frac{d\phi_J}{dt} = \mathbf{F}_I \qquad I = 1, \cdots, m_t \qquad (7a)$$

$$\tilde{\mathbf{E}}_{IJ}\mathbf{c}_J + \tilde{\mathbf{B}}_{IJ}\frac{d\mathbf{c}_J}{dt} = \tilde{\mathbf{F}}_I \qquad I = 1, \cdots, m_t \qquad (7b)$$

where the subscripts refer to nodes, m_t is the total number of nodes, ϕ_J and $\mathbf{c}_J$ are unknown head and concentration vectors, respectively, to be determined, and the coefficient matrices are given by *Huyakorn et al.* [1987]. The finite element equations (7) represent a transformation of the model (6) to a system of algebraic equations. Note that all numerical solutions involve the solutions of linear systems. Perhaps the most crucial part of using a supercomputer efficiently is solving such systems efficiently. Because of the importance of linear algebraic systems in the solution of solute transport models, various solution methods are examined in detail below.

Having a supercomputer available can influence even the basic approach to obtaining numerical solutions. The finite difference method applied to the above problem would give a different linear system of equations which would also have to be solved numerically. The finite element method, although more suited to irregular domains, is usually time consuming to code. Finite difference methods are more simple to set up but not as versatile. A scientist must invest considerable time and effort to set up a problem so that a high-accuracy numerical solution with reduced CPU time and storage requirements will result. The question which then arises is whether this investment is justified in the face of the relatively cheap, high computation rates and large memory available on a supercomputer.

MATRIX EQUATIONS: MODERN SOLUTION METHODS

In this section, algorithms leading to the solution of the linear system $\mathbf{Ay} = \mathbf{b}$ (where $\mathbf{A}$ is a known, often banded, square coefficient matrix of dimension N, $\mathbf{y}$ is an unknown vector, and $\mathbf{b}$ is a known vector) will be discussed. Elements of these arrays are identified by a_{ij} for the elements of $\mathbf{A}$, etc. The type of supercomputer available will be the arbiter of the best algorithm to use. In particular, algorithms designed for a single vector processor cannot be expected to perform well on a massively parallel machine. The literature devoted to numerical methods for the solution of linear systems of equations on supercomputers is very extensive, and only a small overview can be attempted here. *Ortega and Voigt* [1985] review general applications of vector and parallel machines to the solution of partial differential equations. For the purposes of this section, we will refer frequently to the degree of parallelism of an algorithm, meaning the number of operations that can be carried out at the same time [*Hockney and Jesshope*, 1981, section 5.1.2]. On a vector machine this is given by the maximum vector length that can be processed at a single time, and on a parallel machine it is the number of processors operating concurrently.

Direct Methods

The most familiar direct method is Gaussian elimination, which solves $\mathbf{Ay} = \mathbf{b}$ by reducing $\mathbf{A}$ to triangular form and then proceeding to arrive at the solution by back

substitution. On vector machines it can be shown that Gaussian elimination is not a suitable method of factorization. Parallel machines fare somewhat better, since for each step in the reduction, each column of A can be handled independently. *Voigt* [1977] details the efficient implementation of the factorization phase of Gaussian elimination in the Cray-1.

Ortega and Voigt [1985] point out three drawbacks to using factorization algorithms on parallel arrays: (1) the degree of parallelism is reduced at each step, (2) significant communication between processors is necessary, and (3) it may be difficult to schedule the work efficiently. An example of drawback 3 is row pivoting, i.e., swapping rows in the matrix to avoid rounding errors induced when dividing by small numbers. It is likely that introducing a sequential process such as a pivoting strategy would cause substantial programming difficulties.

Another approach is to use the Givens or Givens-Householder algorithms [*Young and Gregory*, 1972, section 4.2] to tridiagonalize A by a series of plane rotations. For parallel processors the Givens algorithm is slightly more effective [*Sameh and Kuck*, 1978]. The resulting tridiagonal systems can be solved by substitution using the Thomas algorithm [*Noye*, 1982]. The Thomas algorithm, however, is manifestly serial. For example, a lower triangular, upper triangular (LU) factorization needs to incorporate operations of the type

$$u_i = d_i - f_i e_{i-1}/u_{i-1} \qquad i = 2, \cdots, N \tag{8}$$

where the elements of the upper and main diagonals and subdiagonals are given by d_i, e_i, and f_i, respectively, and the u_i are the diagonal elements of the upper matrix of the LU factorization. The dependency of u_i on u_{i-1} precludes vectorization or parallelization of this statement. VLIW machines, on the other hand, perform very well in this situation. An immediate way to parallelize (8) is to solve it iteratively in vector form [*Traub*, 1973]. The convergence rate of the iterations increases with increasing diagonal dominance of the system. Parallel direct methods have been developed as well. *Stone* [1973] reformulates (8) as

$$\begin{bmatrix} q_i \\ q_{i-1} \end{bmatrix} = \prod_{j=2}^{i} \begin{bmatrix} d_j & -f_j e_{j-1} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ 1 \end{bmatrix} \tag{9}$$

where $u_i = q_i/q_{i-1}$. All the q_i can be found in $\log_2 N$ iterations. By comparison, serial machines solving (8) take time proportional to N.

Iterative Methods.

Iterative methods tend to be more suited to parallel and vector machines than direct methods. The Jacobi and Gauss-Seidel methods are well-known examples of iterative schemes. Relaxed Gauss-Seidel iteration is written as

$$y_i^{k+1} = y_i^k + w\left(b_i - \sum_{j=1}^{i-1} a_{ij} y_j^{k+1} - \sum_{j=1}^{N} a_{ij} y_j^k\right) a_{ii}^{-1} \tag{10}$$

$$i = 1 \cdots, N \qquad k \geq 0$$

where w is the relaxation parameter and k indexes iterations. The scheme, which diverges unless $0 \leq w \leq 2$ [*Hager*, 1988, section 7-3], is usually termed successive over relaxation (SOR) for $1 < w \leq 2$ or successive under relaxation (SUR) if $0 \leq w < 1$, while it reduces to the Gauss-Seidel algorithm if $w = 1$.

Although the Jacobi method is eminently suited to computation on parallel or vector machines, it converges too slowly to be useful. At the same time, the more rapidly convergent SOR method does not present any easy routes to computation by parallel methods. Asynchronous SOR [*Kung*, 1976; *Baudet*, 1978; *Barlow and Evans*, 1982] is an alternative for parallel machines. Basically, Jacobi-like iterations are carried out by independent processors, with each processor picking up the most recent iterates from a common memory. Under fairly general conditions the rate of convergence is not reduced when compared with the usual serial form of the algorithm.

An interesting iterative scheme, termed the quadrant interlocking method, was developed by *Evans and Hatzopoulos* [1979], *Evans et al.* [1981] and *Evans* [1982]. The coefficient matrix A is set equal to **WZ**, where

$$\mathbf{W} = \begin{bmatrix} 1 & & & 0 \\ & 1 & 0 & \\ & 0 & 1 & \\ 0 & & & 1 \end{bmatrix} \begin{bmatrix} & W_{ij} & & \\ & & W_{ij} & \\ & & & \\ & & & \end{bmatrix} \tag{11a}$$

and

$$\mathbf{Z} = \begin{bmatrix} & Z_{ij} & & \\ 0 & & & 0 \\ & & & \\ & Z_{ij} & & \end{bmatrix} \tag{11b}$$

The triangular portions of (11) are filled with elements to be determined. These unknown elements are calculated beginning with the outermost elements in the unknown matrices. The algorithm proceeds by first assigning

$$z_{1i} = a_{1i} \qquad z_{Ni} = a_{Ni} \qquad i = 1, \cdots, N \tag{12a}$$

Next, the 2×2 linear systems are solved:

$$\begin{bmatrix} z_{11} & z_{N1} \\ z_{1N} & z_{NN} \end{bmatrix} \begin{bmatrix} w_{i1} \\ w_{iN} \end{bmatrix} = \begin{bmatrix} d_{i1} \\ d_{iN} \end{bmatrix} \qquad i = 2, \cdots, N-1 \tag{12b}$$

Equations similar to (12) can be solved to find the other unknown elements of **W** and **Z**. Each ring of elements can be calculated independently, and the computation is suited to parallel processing. With these operations completed, the original linear system becomes

$$\mathbf{WZy} = \mathbf{b} \tag{13a}$$

or

$$\mathbf{Wz} = \mathbf{b} \tag{13b}$$

$$\mathbf{Zy} = \mathbf{z} \tag{13c}$$

Equation (13*b*) is solved for the intermediate unknown vector **z**, whereupon (13*c*) is solved for **y**.

The Preconditioned Conjugate Gradient Method

The conjugate gradient method (CG) was first presented by *Hestenes and Stiefel* [1952]. The method is based on the fact that the solution of $\mathbf{Ay} = \mathbf{b}$ minimizes the quadratic functional $\mathbf{y}^T\mathbf{AY} - 2\mathbf{b}^T\mathbf{y}$ if **A** is symmetric and positive definite [*Hager*, 1988, section 7-5], and the superscript *T* indicates transposition. It is this functional which is minimized to obtain the unknown vector **y**. In the absence of rounding error the method converges to the exact solution for **y** in *N* steps. It has been observed that for some matrix types a reasonable approximation to the solution is obtained after relatively few steps, leading to the use of the method as an acceleration technique. The convergence rate of CG depends on the condition number of **A**. Preconditioning is therefore used to increase the convergence rate [*Hager*, 1988, section 7-5]. *Concus et al.* [1976] note the following advantages of the method: (1) not all of **A** needs to be stored in memory at one time, (2) unlike relaxation schemes, e.g., (10), parameters used by the algorithm are automatically determined, and (3) some restrictions on **A** required by other methods can be relaxed. A vectorized preconditioned CG (PCG) algorithm was given by *Kershaw* [1982].

The first step in the algorithm is to choose a preconditioning matrix **B**, related to **A**, such that **B** is symmetric positive definite and the system $\mathbf{B}\hat{\mathbf{r}} = \mathbf{r}$ is easily solved. For example, **B** could be the tridiagonal part of **A**. One version of the PCG algorithm is

0. Guess $\mathbf{y}_0$. Solve $\mathbf{B}\hat{\mathbf{r}}_0 = \mathbf{b} - \mathbf{A}\mathbf{y}_0 = \mathbf{r}_0$. Let $\mathbf{p}_0 = \hat{\mathbf{r}}_0$.

1. Increment *k* by 1, then calculate

$$\beta_k = \mathbf{r}_k^T \mathbf{r}_k / \mathbf{p}_k^T \mathbf{A}\mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \beta_k \mathbf{A}\mathbf{p}_k$$

$$\hat{\mathbf{r}}_{k+1} = \mathbf{B}^{-1}\mathbf{r}_{k+1}$$

$$\alpha_k = \mathbf{r}_{k+1}^T \hat{\mathbf{r}}_{k+1} / \mathbf{r}_k^T \hat{\mathbf{r}}_k$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \alpha_k \mathbf{p}_k$$

2. Compute $\mathbf{y}_{k+1} = \mathbf{y}_k + \beta_k \mathbf{p}_k$

3. If convergence condition is not satisfied, go to step 1.

If **A** is not symmetric and positive definite but is invertible, then premultiplying the equation $\mathbf{Ay} = \mathbf{b}$ by $\mathbf{A}^T$ gives a system in the required form. The PCG algorithm then involves more computations, however. An alternative is the ORTHOMIN algorithm [*Behie and Vinsome*, 1982]. Qualitatively, the ORTHOMIN algorithm is similar to the one given above, although the computational work load is increased because of the nonsymmetric coefficient matrix **A**. Details of the algorithm, related vectorization issues, and applications to petroleum reservoirs are given by *Behie and Forsyth* [1983, 1984].

On vector machines, PCG-type methods appear to be the most efficient available for linear systems of equations encountered in subsurface flow problems, as shown in a recent study by *Meyer et al.* [1989]. The choice of preconditioner is related to the type of computer used. *Meyer et al.* [1989] compared a number of preconditioning schemes implemented on the Cray X-MP/48, a vector machine, and the Alliant FX/8, an eight-processor vector-parallel machine. Overall, low-order polynomial preconditioning [e.g., *Saad*, 1985] was found to give the best results. The largest problem solved by *Meyer et al.* [1989] involved 980,000 unknowns and took approximately 1 hour CPU time on the Alliant, with a factor of 15 speedup expected for the Cray. *Ababou et al.* [1988] used a Cray-2 to perform a large-scale three-dimensional flow simulation but did not use the PCG method. For this case, *Meyer et al.* [1989] estimated that using polynominal preconditioning with CG would have reduced the computation time by an order of magnitude. This example highlights the importance of algorithm selection for large-scale problems.

Other Methods

Other solution methods, although useful, have not been discussed in this section because of space limitations. For example, many problems lead to sparse coefficient matrices that are solved efficiently by using the multigrid method [*McCormick*, 1987]. Supercomputers have been applied in a variety of multigrid applications [*McCormick*, 1988]. Spectral solution methods exist as well [*Voigt et al.*, 1984]. Further information can be obtained from the review of *Ortega and Voigt* [1985] or the summary of *Hockney and Jesshope* [1988, chapter 5].

DATA ANALYSIS

New or updated solute transport models appear regularly in the hydrological literature. Like any scientific theory, models must be tested and evaluated by comparison with experimental data to ascertain their usefulness in practice. Two types of tests are performed commonly. In the first, more rigorous case, the model parameters are determined independently of the experiment, if possible. Then, with knowledge of the experimental conditions, the model is applied to predict some initially unknown aspect of the experimental system. The second case, which is not a model test, is simply to fit the model to the data by some procedure which determines its parameters. If the model fits well and the parameter values coincide with the analyst's expectations, then the model has been "validated." We refer to these two possibilities as testing by prediction and testing by validation, respectively.

An example of the experimental difficulties involved in applying theory predictively is provided by the Borden aquifer solute transport experiment [*Roberts and Mackay*, 1986]. This extremely valuable and meticulously performed experiment was carried out in a hydrologically uncomplicated aquifer. Known quantities of solutes were injected into the aquifer, and their subsequent movement was monitored by an extensive sampling array over a 3-year period. By averaging the experimental data over the vertical axis, *Freyberg* [1986] and *Sudicky* [1986] found that the theory of *Dagan* [1982, 1984] predicted the temporal growth of the tracer plume's second central moments reasonably well. Analyses of this experiment have highlighted the large uncertainties introduced by lack of knowledge of the initial configuration of the solute in the aquifer [*Barry et al.*, 1988; *Naff et al.*, 1988].

Because testing by prediction is not often a viable option, testing by validation becomes the standard by which models are judged. In a given case the inclusion of substantial experimental noise ensures that a large class of models fits the data equally well. This scenario will always favor more simple, even empirical models, as there is little point in invoking a complex, albeit more theoretically sound model if it offers no advantages over simpler approaches. Therefore experimental uncertainties can be seen to have a profound effect on the degree to which a new theory is regarded as being an improvement.

Scientists and engineers frequently wish to characterize a transport process in the light of a selected model. Mechanistically, this case is identical to testing by validation in that the goal is to determine the model's parameters. Nonlinear regression is used to fit the model and so obtain estimates of the parameters. Statistical theory provides a very elegant solution to this problem for linear regression with normally distributed measurement errors. *Efron* [1982, chapter 1, 1988] suggests that the constraints imposed by parametric statistical theory can be abandoned in an age of readily available, large-scale computers and inexpensive computation. Some of these computationally intensive techniques are reviewed in the following. In doing so, we echo the sentiment expressed in the recent article of *Tichelaar and Ruff* [1989], who aimed to "promote the use of resampling techniques in geophysics."

Presume that the model under consideration is a scalar function $f = f(\mathbf{E}; \mathbf{P})$ of the independent (known) variables $\mathbf{E}$ and unknown parameters $\mathbf{P}$, where $\mathbf{E}$ is a matrix and $\mathbf{P}$ is a vector composed of elements p_j. The experimental data consist of n_e measurements, $f_i(\mathbf{E}_i)$, $i = 1, \cdots, n_e$. Define a scalar "distance" function as $H(\mathbf{e}_1, \mathbf{e}_2)$, where $\mathbf{e}_1$ and $\mathbf{e}_2$ are locations in the space of independent variables. Usually, we take $H(\mathbf{e}_1, \mathbf{e}_2) = |\mathbf{e}_1 - \mathbf{e}_2|^2$ or $H(\mathbf{e}_1, \mathbf{e}_2) = |\mathbf{e}_1 - \mathbf{e}_2|$, where the former choice is the square of Euclidean distance and the latter is the absolute value of the distance. In these and many other cases, H satisfies some "stationarity" assumption, i.e., $H = H_r(r)$, meaning that H is unchanged by translation. In fitting the model, one minimizes the discrepancies between the model and the data by suitable choice of $\mathbf{P}$; i.e., choose $\mathbf{P}$ such that

$$d = \sum_{i=1}^{n_e} H[f(\mathbf{E}; \mathbf{P}) - f_i(\mathbf{E}_i)] \tag{14}$$

is minimized. Minimization of d can be achieved by differentiation of (14) with respect to the model parameters, or

$$\frac{\partial d}{\partial p_j} = \frac{\partial}{\partial p_j} \sum_{i=1}^{n_e} H[f(\mathbf{E}; \mathbf{P}) - f_i(\mathbf{E}_i)] \qquad j = 1, \cdots, n_p \tag{15}$$

where n_p is the length of $\mathbf{P}$. The combination of parameters yielding a minimum d is obtained by the solution of the simultaneous equations:

$$\partial d/\partial p_j = 0 \qquad j = 1, \cdots, n_p \tag{16}$$

Equation (16) is, in general, a system of nonlinear equations. In this outline of regression it has been assumed that the model output is scalar. Extension to a vector of predictions follows without difficulty.

Traditional linear regression results when the system (16) is linear in the elements of $\mathbf{P}$, the function H in (15) measures the square of Euclidean distance, and the distribution of prediction errors satisfies the normality assumption. After fitting the model it is necessary to check a posteriori, among other things, that the distribution of residuals, s_i (where $s_i = H(f(\mathbf{E}; \mathbf{P}) - f_i(\mathbf{E}_i))$), $i = 1, \cdots, n_e$, is approximately normal [*Cox and Snell*, 1968]. If this condition is satisfied, it follows that one can calculate a covariance matrix of the residuals as well. Linear theory gives both an estimate of $\mathbf{P}$ and an indication of the accuracy of this estimate.

The Bootstrap

What happens when the residuals do not satisfy the normality assumption or when the model is nonlinear? A

useful computer-based method is the bootstrap. *Efron* [1982, section 5.7] outlines the method. In brief, the idea is to use the data set to define the distribution of residuals. This empirical distribution is resampled a large number of times to create synthetic data sets. For each resampling an estimate $\mathbf{P}^*$ of $\mathbf{P}$ is obtained using some nonlinear fitting method such as that of *Marquardt* [1963]. The bootstrap estimate of $\mathbf{P}$ is $\mathbf{P}_B$, the mean of the $\mathbf{P}^*$. The bootstrap estimate of the covariance matrix $\mathbf{C}_B$ of $\mathbf{P}_B$ is

$$\mathbf{C}_B = \frac{1}{n^* - 1} \sum_{k=1}^{n} (\mathbf{P}^{*k} - \mathbf{P}_B)(\mathbf{P}^{*k} - \mathbf{P}_B)^T \qquad (17)$$

where n^* is the number of resamplings and the superscript k identifies the various $\mathbf{P}^*$ for the resampled data set. Of course, many variations of this procedure are possible. Note that the groundwater literature contains examples of precisely this method [e.g., *Yeh and Wang*, 1987], although the term bootstrapping has not been used as such. The research of *Yeh and Wang* [1987] was directed at parameterizing solute transport models. These authors cite other, similar analyses. The role of supercomputers in parameter estimation is apparent from the observation of *Yeh and Wang* [1987] that the "intensive computational effort required by the Monte Carlo method . . . may prohibit [its] application to . . . large scale problems." A more complete review of model parameterization studies in groundwater hydrology is given by *Yeh* [1986]. The generality of these methods is that they can be applied in conjunction with many existing applications. Recent examples to which this comment applies are the parameter estimation studies of *Kool and Parker* [1988], who analyzed unsaturated flow, and *Mishra and Parker* [1989], who dealt with both water flow and solute transport.

Cross Validation

Cross validation has been used frequently to estimate spatial covariance functions [e.g., *Samper and Neuman*, 1989]. The method has at its base a very simple idea which can be applied in many contexts. For example, data from the solute transport experiment at the Borden aquifer had to go through a multistep analysis before the desired spatial statistics were obtained [*Freyberg*, 1986; *Barry et al.*, 1988]. In the case of solute plumes the statistics of interest are often the plume's zeroth-, first-, and second-order spatial movements, from which one can estimate the plume mass, mean position and variance. The calculation of these quantities amounts to a number of triple integrations to be carried out numerically. Central spatial moments are defined by

$$\mu_{ijk}(t) = \frac{1}{M_{000}(t)} \int (x_1 - \mu_1)^i (x_2 - \mu_2)^j \cdot (x_3 - \mu_3)^k c_\theta(x, t)\, dx \qquad (18)$$

where $M_{000}(t)$ is the plume mass and $c_\theta(x, t) = c(x, t)\theta(x, t)$. The function c_θ will be known or estimated at only a finite number n_t of locations or as averages over small volumes centered at those locations. An interpolation procedure must be performed to estimate c_θ at unsampled locations. Again, this involves a nonlinear regression procedure, and in principle, the nonlinear regression procedure described above, or any of the methods described by *Yeh* [1986] could be used. This includes methods from spatial statistics [*Ripley*, 1981], including geostatistical methods [*Journel and Huijbregts*, 1978]. However, as the aim is simply to estimate the integral in (18), simple alternatives based on cross validation seem reasonable. One procedure could consist of defining the interpolator:

$$c_\theta(x, t) = \frac{\sum_{i=1}^{n_t} g[H_g(x, x_i, t, t_i); \mathbf{P}] c_\theta(x_i, t_i)}{\sum_{i=1}^{n_t} g[H_g(x, x_i, t, t_i); \mathbf{P}]} \qquad (19)$$

where H_g is a "distance" function of both space and time, e.g., $H_g \equiv H_g(x_1, x_2, t_1, t_2)$, and the function $g(H_g; \mathbf{P})$ generalizes H_g, to include a number of adjustable parameters in the vector $\mathbf{P}$. Elements of $\mathbf{P}$ can be used in the definition of H_g, e.g., to scale the distance function. Cross validation then consists of choosing $\mathbf{P}$ so as to minimize S_{cv}, where

$$S_{cv} = \sum_{j=1}^{n_t} H_r[c_\theta(x_j, t_j) - \sum_{\substack{i=1 \\ i \neq j}}^{n_t} c_\theta(x_i, t_i)] \qquad (20)$$

If $g[H_g(x, x, t, t); \mathbf{P}] = \infty$, then the function in (19) will reproduce the measured data. The cross-validation idea, written as an equation in (20), is to try to predict each measurement using the rest of the data set. The sum of the prediction errors, S_{cv}, is minimized to yield $\mathbf{P}$. The method given here is to try any number of different functions, g, in (19), selecting the one that gives the least S_{cv}. Other functional forms could be used in (19) instead of the linear combination suggested. Note that g can be defined to include only a limited spatial or temporal domain in (19). A variation of this procedure (called the "supersmoother"), discussed by *Efron* [1988, section 6], is to define the function g as a linear interpolant and then to select the size of the spatial domain using a criterion like that in (20).

When a suitable g is chosen, c_θ in (18) is replaced by the interpolant in (20), and the required integrations are performed numerically. The moment estimates obtained in this manner are point estimates only. It is possible to calculate estimated variances of these point estimates, again by some resampling scheme. One such scheme is jackknifing.

Jackknifing

Assume, as in (18), that we wish to derive a statistic, Q from a data set. The method used to calculate Q (e.g., maximum likelihood) is known or suspected to be biased.

Jackknifing [*Quenouille*, 1956; *Gray and Schucany*, 1972] is a technique to reduce bias in the estimate of the statistic Q. The data set, consisting of n_e measurements, is divided into k groups, each containing m elements. Each group is removed, in turn, and the statistic $Q(j)$, $j = 1, \cdots, k$, corresponding to Q is calculated. The jackknifed estimate Q_j is then given by

$$Q_j = kQ - (k-1)\overline{Q} \tag{21}$$

where

$$\overline{Q} = \frac{1}{k}\sum_{j=1}^{k} Q(j)$$

The variance of Q_j, Var (Q_j), is estimated from [*Tukey*, 1958]

$$\text{Var}\,(Q_j) = \frac{k-1}{k}\sum_{j=1}^{k}[\overline{Q} - Q(j)]^2 \tag{22}$$

Mosteller and Tukey [1968, section E2] show that under certain conditions, confidence intervals for Q_p, the population parameter being estimated, can be derived from

$$T_Q = (Q_J - Q_P)\,[\text{Var}\,(Q_J)]^{-1/2} \tag{23}$$

where T_Q has a t distribution with $k - 1$ degrees of freedom. A detailed description of the construction of confidence intervals using this approach is given by *Barry and Sposito* [1990].

Visualization

Improvements in data visualization have kept pace with developments in computing hardware. Plotters and associated software are standard equipment. Much specialized hardware and software is available also. A powerful combination for real-time simulation is the combination of supercomputer computational rates and a dedicated graphics workstation. An example utilizing this technology was undertaken as part of a more general study on solute transport. Three-dimensional visualizations of the bromide tracer plume from the Borden site experiment [*Roberts and Mackay*, 1986] were generated using facilities at the National Center for Supercomputing Applications (NCSA). Briefly, the raw data, consisting of solute concentration measurements at given spatial locations for the 14 plume samplings, were fitted to an interpolation function using the NCSA's Cray X-MP/48 computer following the cross-validation procedure described above. Full details are given by *Barry and Sposito* [1990]. Dense ($4–8 \times 10^4$ points), regular grids of interpolated concentration data were produced, and surface contours—three-dimensional versions of contour lines—were computed using the "marching cubes" algorithm [*Lorensen and Cline*, 1987]. The contour surfaces are allowed to be semitransparent and so allow the passage of light. Since contours are formed at regular concentration increments, domains of higher concentration are surrounded by more surfaces than areas of lower concentrations. When (mathematical) light rays are passed through the surfaces, they are diffused. Domains containing relatively higher concentrations appear as being relatively more opaque. There is complete freedom in how the final images are viewed due to variations in the light source and observation positions.

Some renditions of this process appear in Plate 1. The bromide plume after 462 days of travel time in the aquifer is viewed as moving, approximately, from the upper right to lower left portions of Plate 1*a*. Plate 1*b* provides a side view of the plume. The image is clearly very unsymmetrical, the front and rear portions corresponding to the vertically averaged bimodal two-dimensional plumes noted by *Freyberg* [1986]. Solute transport appears to be highly concentrated along very stratified layers within the aquifer. The difficulty in achieving complete plume sampling is revealed by the lagging edge of the plume apparently extending beyond the image into an unsampled area. Plate 1*c* displays the plume moving directly toward the viewpoint. The suggestion that solute transport is dominated by the more highly permeable regions within the aquifer is reinforced. These observations are consistent with the finding of *Sudicky* [1986] that the ratio of the vertical and horizontal correlation length scales of the hydraulic conductivity at the Borden site suggests that the aquifer is strongly anisotropic and, indeed, reflects the lenticular geological structure. There is a clear indication, also, that a mobile-immobile region model might provide an adequate representation of solute breakthrough in this system, as already found by *Golz and Roberts* [1986].

The images in Plate 1 are but a few of a large group. In terms of computational effort, the total CPU time required to produce each image ranged from 10 min to 10 hours on an eight-processor Alliant X-8 minisupercomputer.

SOLUTE TRANSPORT ON SUPERCOMPUTERS: APPLICATIONS

Subsurface hydrology researchers are utilizing the increasingly available supercomputer time and resources. The extent of supercomputer use, however, is not extensive at the present time. An overview of recent applications follows.

Particle Tracking

Nonreactive solute transport can be modeled quite effectively by using particle-tracking techniques. All approaches to date are based on the dilute solute assumption; i.e., the water flow model is solved separately to give the groundwater velocity field. This velocity is then used to provide the deterministic motion of the "solute" particles in the simulation, with an additional random component providing the dispersive mechanism. The theoretical basis of the method is the Langevin equation.

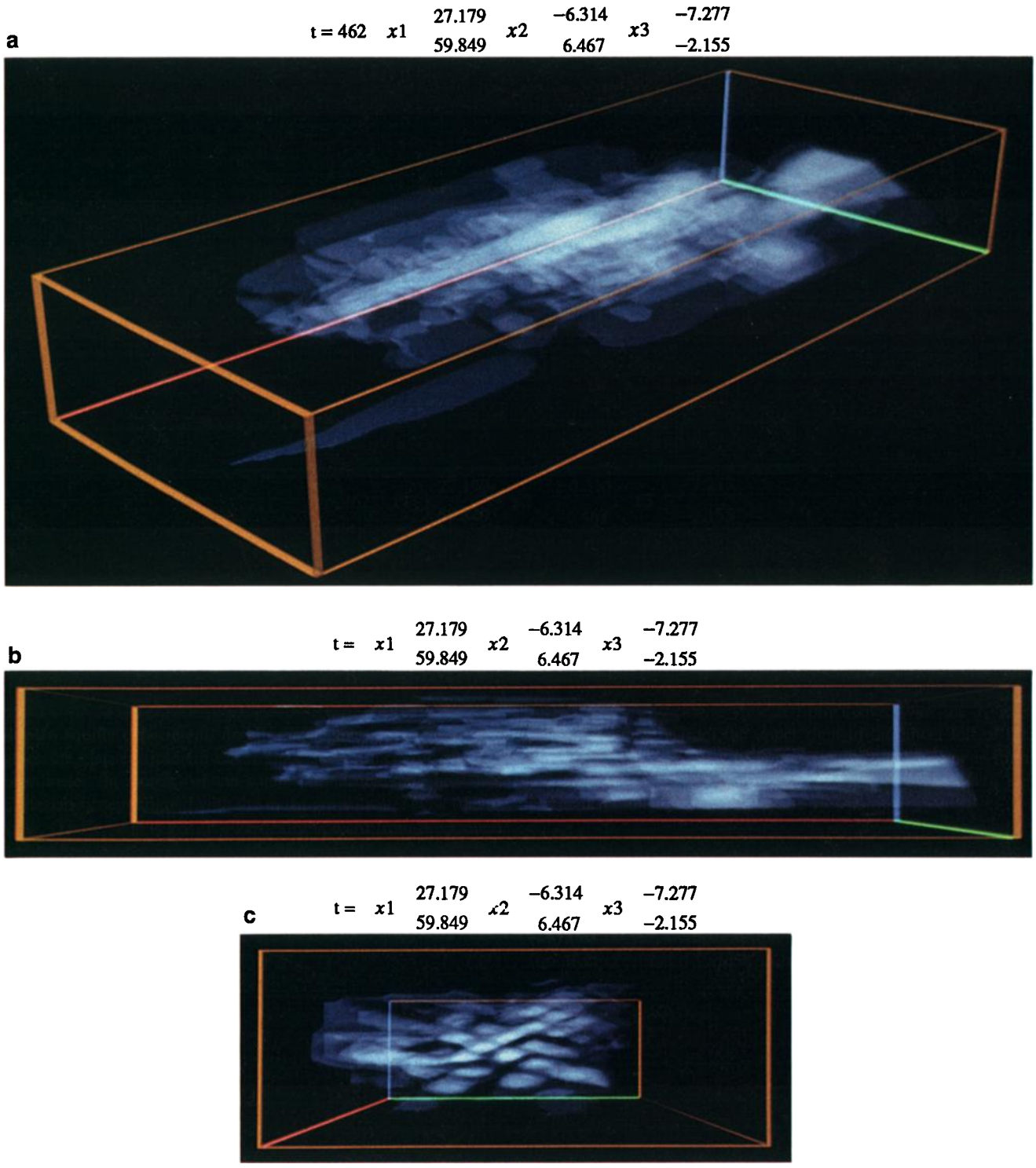

Plate 1. Three-dimensional images of the bromide tracer in the Borden aquifer experiment. In all images, the upper line of data gives the elapsed time (in days) since solute injection, and the coordinates (in meters) of the solute plume bounding box in a Cartesian system aligned with the mean groundwater flow (the x_1 axis). The x_3 axis is positive vertically upward from the ground surface. (*a*) The plume viewed from above, with the x_1 axis passing diagonally from the upper right to the lower left of the viewpoint. (*b*) Side view of the plume. The mean flow is from left to right. (*c*) Frontal view of the plume. In this case the plume is moving directly toward the viewpoint.

The one-dimensional form of this stochastic differential equation is [*Gardiner*, 1983, section 4.1]

$$dz/dt = a(z, t) + b(z, t)\,\xi(t) \qquad (24)$$

where a and b are known functions and $\xi(t)$ is a stochastic forcing function idealized as white noise. As it stands, (24) has no physical meaning because white noise is delta function correlated [*van Kampen*, 1981, chapter VIII]. If,

however, we follow the Itô interpretation, then (24) is equivalent to

$$z(t+\Delta t) - z(t) = a[z(t), t]\Delta t + b[z(t), t] \int_{t}^{t+\Delta t} \xi(t')\, dt' \quad (25)$$

where Δt is a finite time interval. Clearly, different realizations of $\xi(t)$ will influence the location of the particle under consideration. Under suitable conditions, (25) is equivalent to the Fokker-Plank or Kolmogorov forward equation [*Cox and Miller*, 1965, chapter 5]:

$$\frac{\partial P}{\partial t} = \frac{1}{2}\frac{\partial^2 (b^2 P)}{\partial z^2} - \frac{\partial (aP)}{\partial z} \quad (26)$$

$P(z, t)$ is the probability density function of the position of any particle whose movement is controlled by (25). Conversely, the distribution of a "large" number of particles will be described by (26) [*Barry and Sposito*, 1988]. With appropriate choices of the functions a and b, (26) is equivalent to the equation describing the transport of a nonreactive solute under the assumption that density effects can be ignored. A common problem with particle-tracking methods is the noisy solute distributions resulting from the simulations. Distributions are derived by assigning a solute mass to each particle and averaging over a specified domain size. The noise is proportional to $1/N'^{1/2}$, where N' is the number of particles. The most straightforward way to reduce noise is to increase N', although smoothing or filtering of the concentration predictions may be more economical computationally. Similar equations can be derived for multidimensional forms of (24). Thus solutions to (26) can be obtained by using a Monte Carlo approach applied to (25). Each particle in the simulations moves independently, so (25) is well suited to evaluation by a supercomputer. Indeed, the degree of parallelism can be at least the number of particles in the simulation. Computationally, particle tracking is algebraically uncomplicated and extremely repetitive, making for an easily vectorized code.

A recent study using particle tracking is that reported by *Tompson et al.* [1988]. These authors review in detail the theory outlined above, including the multidimensional forms of the functions a and b in (24) which are appropriate for simulations of solute transport for both saturated and variably saturated flows. The simulations of *Tompson et al.* [1988] were aimed at evaluating stochastic theories, as discussed elsewhere in this review. To that end, flow in a heterogeneous domain was generated by using the large-scale flow simulator of *Ababou et al.* [1988], developed for the Cray-2 supercomputer. *Ababou et al.* [1988] used the strongly implicit procedure, a less efficient forerunner of the PCG method, to solve their linear systems of equations [*Meyer et al.*, 1989]. *Tompson et al.* [1988] did not discuss the specific algorithm design or computer used in their simulations. Their code, however, is designed to track many thousands of particles on grids of up to 10^7 nodes. Simulations of this kind are possible only on supercomputers.

Another example of the particle-tracking method, although not supercomputer based, was presented by *Illangasekare and Döll* [1989]. These authors developed a model of reactive solute transport based on the discrete kernel [*Illangasekare and Morel-Seytoux*, 1978] numerical solution of the vertically averaged flow equation. A particle-tracking scheme was implemented to model solute transport. Following *Garder et al.* [1964], solute particles were moved deterministically through the system in response to the local driving groundwater velocity. The concentration in each computational cell was determined by areal averaging assuming the particles covered an area rather than a point location. Within each cell an explicit finite difference solution of the solute transport equation gave concentration changes to a number of processes, including dispersion, sorption, and mobile-immobile region exchange. The areal averaging scheme was very effective in removing the problem of noisy solute concentrations.

Numerical Solutions

Conventional solutions based on the governing equations of flow and solute transport are more common. It seems that most applications, including those discussed below, use the assumption of a dilute solute, so that the governing equations may be treated separately.

Perhaps the most elaborate solute transport model yet developed and implemented on a supercomputer is the DYNAMIX code [*Liu and Narasimhan*, 1989*a*]. This is a multidimensional, multispecies, reactive chemical transport model that includes the transformation mechanisms: acid-base aqueous complexation, oxidation-reduction, precipitation-dissolution, and kinetic mineral dissolution. The code represents the influence of supercomputers in enabling modeling efforts that move away from the simple distribution coefficient (K_d) approach to modeling adsorption/desorption reactions [*Liu and Narasimhan*, 1989*a*]. This version of DYNAMIX updates the model of *Narasimhan et al.* [1986]. The transport model will handle large numbers of chemical components, species, and phases. The numerical solution of the model requires solving a set of multidimensional transport equations, one for each chemical component, and a set of thermodynamic equilibrium chemistry equations. An explicit finite difference scheme is used to solve the transport equations. The thermodynamic equations form a nonlinear algebraic system solved by using the Newton-Raphson method. Within each element of the numerical grid, complete mixing of the aqueous chemical species is assumed.

Example problems solved by DYNAMIX are given by *Liu and Narasimhan* [1989*b*]. After favorable comparison of DYNAMIX to the one-dimensional codes of *Walsh et al.* [1984] and *Carnahan* [1986], *Liu and Narasimhan* [1989*b*] performed a simulation of solute transport from a

contamination site. The reactive trace elements selenium and arsenic were included. Their 30-year simulation period showed that both chemicals were strongly retarded in the subsurface. The total CPU time needed was 3.4 hours on a Cray X-MP/14. The code does not take advantage of the Cray's vector speed capabilities. Rewriting computationally intensive portions of the code to allow vectorization would likely lead to substantial improvements in the CPU time used.

A model that explicitly makes use of supercomputer characteristics in algorithm selection was developed by *Chiang* [1985] (cf. *Chiang et al.* [1989]). The code was developed to model pumping-induced flow and contaminant transport in a two-dimensional domain. The model uses a modified method-of-characteristics procedure to solve the governing equations. Example simulations included transport in domains having a random hydraulic conductivity field. Realizations of the latter were developed by using the turning bands method [*Mantoglou and Wilson*, 1982]. The name "turning bands" denotes an algorithm that generates correlated random variates on a group of lines passing through a point. A two- or three-dimensional random field, with a predefined correlation structure, is derived from a series of orthogonal projections from the group of lines. *Tompson et al.* [1989] report on the method's performance and efficiency. *Chiang* [1985] selected the PCG algorithm to provide the necessary numerical solution to the linear system of equations. It was concluded that the code developed was efficient, accurate, and capable of modeling transport in systems with rapidly varying hydraulic conductivity fields.

Frind et al. [1989] report a three-dimensional simulation of transport of organic solutes undergoing aerobic biodegradation. They model a saturated groundwater system in which the solute transport process is described by a nonlinear, coupled set of equations, one for the contaminant and one for dissolved oxygen. The dual-Monod equation [*Borden and Bedient*, 1986] is used to model the growth of the microbial population. By formulating the numerical model to produce a symmetric coefficient matrix, the PCG solution method could be used effectively. A 24×10^3 node simulation over 155 time steps (300 days) took about 2 hours of CPU time on a Cray X-MP/24. The authors concluded that comparable two-dimensional simulations were to be avoided, as oxygen availability was markedly affected by problem dimensionality.

Sudicky [1989*a*] considered the transport of a dilute solute in a steady, nonuniform, two-dimensional flow field. The spatial derivatives were discretized by using the Galerkin finite element procedure. Instead of using a finite difference procedure for the temporal derivative, Laplace transforms were used. The Laplace-transformed problem was shown to not suffer from the numerical dispersion problems associated with other methods. The *Crump* [1976] algorithm for numerical inversion of the Laplace-transformed solution can be used to calculate the time domain solution from a single set of Laplace domain solution values. *Sudicky* [1989*b*] extended the application of the method to the case of fractured porous formations. In the case of the traditional finite difference approximation of the temporal derivative, numerical difficulties arise because of the large differences between the rapid advection-dominated transport in the fractures and much slower diffusional flux into the porous matrix. *Sudicky* [1989*b*] combined the Laplace transform method with ORTHOMIN acceleration for the rapid solution of the algebraic system of equations derived from application of the Galerkin procedure. The combined procedure was shown to be very efficient and accurate. A 30,651-node simulation of downward seepage of solute in a medium containing several thousand fractures took 72 s on a Cray X-MP/24 to obtain solute distributions valid for all times of interest. By using algorithms especially suited to vector supercomputers, Sudicky has developed a code capable of simulating very large, field scale transport problems.

Evaluating Stochastic Solute Transport Theory

As mentioned already, the data base from the Borden aquifer transport experiment has been used to evaluate the predictions of stochastic solute transport theory. Both *Freyberg* [1986] and *Naff et al.* [1988] compared theoretical predictions of the vertically averaged tracer plume dispersivity with those estimated from experimental data and found excellent agreement. *Garabedian et al.* [1988] analyzed reactive solute data from field experiments conducted at Chalk River, Ontario, and Cape Cod, Massachusetts, using theory developed on the assumption that both the flow field and the solute retardation factor are random. Again, only the plume variance was evaluated.

Following the success of the two-dimensional *Dagan* [1984] model in predicting the spread of the vertically averaged tracer plume at the Borden site, *Barry et al.* [1988] checked whether or not the actual plume concentrations conformed with ensemble mean concentrations. The theory was used to predict solute plumes by using the measured plume from a number of different samplings as initial conditions and then comparing the predicted and measured plumes at later times. The data analysis and generation of the predicted plumes, which involved integrating the initial condition numerically, were carried out using a vectorized code on a Cray X-MP/48. It was found that the model predicted the measured plume quite accurately on relatively short time scales, becoming progressively worse with increasing elapsed time. However, the assumptions necessary in the data analysis made equivocal any comparisons of model-predicted and measured plumes.

The ensemble approach has led to elegant analytical results, but as discussed by *Dagan* [1987], they cannot be applied unless certain conditions are met. A demonstration of the possible differences between separate plume realizations is given by *Frind et al.* [1987]. These authors use a two-dimensional vertical slice model of an aquifer

with a hydraulic conductivity field that reproduced the estimated low-order statistics of the Borden aquifer. The random hydraulic conductivity field was produced by using the turning bands method. The local water fluxes, and hence velocities, were computed using the "principal direction technique" [*Frind*, 1982*c*; *Frind and Pinder*, 1982]. The computed velocities in turn were used in the solution of the solute transport equation. Two simulations were performed, one of which used a spatial grid of 10^6 elements. It was clearly demonstrated that local variations in the local hydraulic conductivity dominate plume movement, particularly at early times. The short-term behavior can have persistent, long-term effects. The simulation of *Frind et al.* [1987], which used 10 hours of CPU time on a Cray X-MP/48, revealed the formation of two subplumes due to the initial placement of the solute source in a region of rapidly varying conductivity. The initial vertical dimension of the plume was only a factor of 3.5 times the vertical correlation scale of the ln (K) field. As discussed by *Dagan* [1987], the initial condition dimension must be much greater than the characteristic length scale of the conductivity heterogeneity for the ensemble approach to be valid. That the factor of 3.5 used by *Frind et al.* [1987] was insufficient for ensemble averaging to apply indicates the importance of considerations of scale in the application of the analytical theory. Other, similar simulations reported by *Sudicky et al.* [1990] demonstrate that in many cases, less dramatic results are obtained. For example, *Sudicky et al.* [1990] discuss simulations in which the solute plume approaches a Gaussian distribution, as predicted by stochastic theory.

Sudicky et al. [1990] go on to consider the transport of a biodegradable solute in a heterogeneous porous medium. In this case there are three nonlinear governing equations, one each for the solute, the available oxygen, and the microbial population [*MacQuarrie et al.*, 1990]. The behavior of the plume is quite different. If the biodegradation rate is large enough, then the plume will tend to decrease in size. For this reason, the plume will encounter, and be influenced by, only a small portion of the groundwater velocity field. Under these conditions, plumes resulting from different realizations will not necessarily converge very rapidly to those predicted by ensemble averaging of the governing equations. More comprehensive simulations presented by *MacQuarrie and Sudicky* [1990] show in detail that biodegradable solute plumes evolve differently than plumes composed of nonreactive solutes. In particular, it is suggested that the scale dependence of the plume dispersivity, which is well documented for field scale transport of tracer solutes [*Gelhar*, 1986, Figure 12], does not apply in the case of biodegradable, organic solute plumes.

The appropriateness of ensemble averaging was investigated also by *Black and Freyberg* [1987], who simulated solute transport of an ideal tracer in a perfectly stratified aquifer consisting of homogeneous layers. The hydraulic conductivity was assumed to vary transverse to the direction of flow (and layering). Generation of the synthetic conductivity field involved a significant computational cost, as it involved the numerical solution of a nonlinear algebraic system of equations. The perfectly stratified aquifer, although certainly unrealistic physically, is useful because it tends to represent a "worst case" scenario for ensemble averaging. An interesting result of the simulations was that the concentration uncertainty tends to increase with decreasing ensemble mean concentration.

Black [1988] extended the one-dimensional approach to a vertical slice, two-dimensional model. The turning bands method was used to generate realizations of the ln (K) field. The turning bands method requires the generation of numerous one-dimensional correlated fields. Unlike most previous investigators who used the method of *Shinozuka and Jan* [1972] for this task, *Black and Freyburg* [1990] present an alternative called the "matrix-factorization moving average" method. The method, which was implemented on a Cray X-MP/48, was shown to be both more efficient and more accurate than that of *Shinozuka and Jan* [1972]. After generation of the two-dimensional correlated hydraulic conductivity field, steady flow conditions were produced. Particle tracking was then used to simulate the transport of a solute plume. It was shown that as the "vertical multiplicity," i.e., the ratio of the vertical length dimension of the system to the vertical correlation scale of the ln (K) field, increases, the mean distance traversed by the plume increases because of the reduced influence of low-conductivity regions. Also, under these circumstances the concentration uncertainty tends to decrease.

CONCLUDING DISCUSSION

No doubt other supercomputer applications, apart from those discussed here, are worthy of mention. On the other hand, the topics covered serve to illustrate possibilities for supercomputer applications. Also, the mechanisms for realizing some degree of the available computational performance in the solution of algebraic systems of equations have been discussed in some detail. In general, choosing algorithms suited to the class of computer being used can well have a large payoff for relatively little effort. Researchers with access to supercomputers naturally want to make use of existing codes as rapidly as possible. It has been pointed out already that although codes may run with little or no modification on different machines, including supercomputers, the speedup in execution times expected may not occur without modifying the code. There is ample evidence that codes optimized for scalar machines will not achieve high computational rates on current machines. The vast majority of scientific codes are written in the FORTRAN language. "Standard" FORTRAN 77 is not well suited to vector or parallel machines, and manufacturers have introduced extensions to allow machine

features to be exploited. For example, *Pelka and Peters* [1986] demonstrate the usefulness of the Cray's GATHER/SCATTER feature for vectorizing finite element solutions of the groundwater flow equation. The research of Sudicky and coworkers cited above and that of *Meyer et al.* [1989] demonstrate the very substantial speedups that can be gained in a wide class of transport problems. The development of more sophisticated compilers will reduce the need for special programming of codes. In the meantime, if researchers are to realize supercomputer computation rates, they will profit by (1) choosing suitable algorithms and (2) rewriting portions of their code. On vector machines like Crays, available timings show that linear systems of equations are most efficiently solved by preconditioned conjugate gradient-type algorithms. If an existing code is being transferred to a supercomputer, a very useful exercise is to time the various tasks performed. It often happens that most of the total CPU time is devoted to performing only small sections of the code. Large speedups will result by making sure that these small sections vectorize as much as possible.

Besides fast computation rates, supercomputers are often linked with other facilities or have other features that make possible new ways of going about research. Accessible memory on supercomputers is generally vast compared with that of usual mainframes. Visualization makes possible ways of viewing processes or results that cannot be achieved otherwise. A typical case is that of a graphics workstation connected to a supercomputer. Results of a simulation can then be viewed directly. Alternatively, slides or movies can be generated.

In this review an attempt has been made to link aspects of solute transport–related research and modeling that appear, on the surface, to link some rather disparate scientific threads. The common denominator, of course, is supercomputer applications. This situation is just as it should be: solute transport encompasses both the physics of flow in heterogeneous media and complex chemical reactions and mixing processes over multiple temporal and spatial scales. Diverse mathematical tools and analyses are necessary to elucidate the behavior of specified systems. The role of a supercomputer should be as another specialized instrument that can be called upon as necessary. It is emphasized that there are a wide variety of circumstances in which supercomputers will aid in yielding valuable insight. One instance where this has occurred is the macroscale behavior of a biodegradable, organic solute plume. *MacQuarrie and Sudicky* [1990] report supercomputer simulations that illustrate the significant differences between this type of plume and the well-studied case of an ideal tracer plume. In the latter case, both field experiments and numerical simulations indicate that under known conditions a reasonably accurate macroscopic description of the plume is possible. Biodegradation effectively limits the spatial domain sampled by the plume with the result that a macroscale description does not appear tenable on realistic time scales.

Dagan [1986], in his review of stochastic flow and solute transport theory, noted the vigorous growth in the number of research papers devoted to stochastic modeling of groundwater flow published in *Water Resources Research.* In retrospect, this surge in publications is entirely reasonable given the applicability of the theoretical approach to modeling fluid transport in heterogeneous permeable formations. Supercomputers offer another potentially valuable avenue for the understanding and resolution of solute transport problems. Their rapidly increasing use in future investigations is therefore a plausible expectation.

NOTATION

$a(x, t)$ known function in the Langevin equation.
$\mathbf{A}$ square coefficient matrix.
$b(x, t)$ known function in the Langevin equation.
$\mathbf{b}$ vector of known quantities.
c solute concentration, $M L^{-3}$.
c_s concentration corresponding to maximum density, $M L^{-3}$.
c_0 initial concentration, $M L^{-3}$.
c_θ $c\theta$, $M L^{-3}$.
$\langle c \rangle$ ensemble mean of c, $M L^{-3}$.
d sum of residuals.
$\mathbf{D}$ solute dispersion tensor, $L^2 T^{-1}$.
D_{ii} dispersion coefficient along the i axis, $i = 1, 2, 3$, $L^2 T^{-1}$.
D_{Mii} macrodispersion coefficient along the i axis, $i = 1, 2, 3$, $L^2 T^{-1}$.
$\mathbf{D}'$ reduced solute dispersion tensor ($=\mathbf{D}/n$), $L^2 T^{-1}$.
$\mathbf{e}$ location vector, L.
$\mathbf{E}$ array of independent variables.
f arbitrary scalar function.
$\mathbf{g}$ gravitational acceleration vector (magnitude g) $L T^{-2}$.
H distance function.
H_g generalized "distance" function.
H_r distance function dependent on separation only.
k scalar version of $\mathbf{k}$, L^2.
$\mathbf{k}$ permeability tensor, L^2.
K hydraulic conductivity, $L T^2$.
$\mathbf{K}$ hydraulic conductivity tensor, $L T^{-1}$.
$m(\omega)$ function in (4) to be specified for a given solute.
M_{ijk} the ijkth raw spatial moment, $M L^i L^j L^k$.
M_S mass of salt.
M_T total mass (solute plus solvent).
n porosity of the porous medium.
n_e number of experimental measurements.
n_l number of locations.
n_p number of parameters.
N dimension of $\mathbf{A}$.
N' number of particles in a particle-tracking simulation.
p thermodynamic pressure of the fluid, $M L^{-1} T^{-2}$.
p_0 reference pressure, $M L^{-1} T^{-2}$.

P probability density function.
$\mathbf{P}$ vector of parameters.
$\mathbf{P}^*$ estimate of $\mathbf{P}$.
P_j element of $\mathbf{P}, j = 1, \cdots, n_p$.
q volumetric flow rate of fluid sources or sinks per unit volume of porous medium, T^{-1}.
$\mathbf{q}$ fluid flux vector, $L\,T^{-1}$.
Q arbitrary statistic.
Q_p population parameter.
R relative performance.
$\mathbf{r}$ residual vector in the PCG algorithm.
$\hat{\mathbf{r}}$ temporary residual vector.
S scalar speed.
S_{cv} cross-validated sum of squares.
S_{ii} Fourier transform of the stationary solute drift velocity covariance function along the i axis, $i = 1, 2, 3, L^5\,T^{-2}$.
S_s specific storativity, L^{-1}.
t time, T.
w relaxation parameter.
x_i distance along the ith axis in an orthogonal coordinate system, L.
$\mathbf{v}$ velocity (magnitude v), $L\,T^{-1}$.
Var variance operator.
V_T total volume.
$\mathbf{y}$ vector of unknowns.
β compressibility coefficient of the solute, $L\,T^2\,M^{-1}$.
γ coefficient of solute mass fraction in density formula (3).
ε empirical density difference coefficient.
η density coupling coefficient.
θ volumetric water content.
κ wave number vector in Fourier domain, L^{-1}.
κ_i element of κ, L^{-1}.
μ fluid dynamic viscosity, $M\,L^{-1}\,T^{-1}$.
μ_i mean plume position along the ith axis, L.
μ_{ijk} the ijkth central moment, $L^i\,L^j\,L^k$.
μ_0 reference dynamic viscosity, $M\,L^{-1}\,T^{-1}$.
$\xi(t)$ white noise process.
ρ fluid mass density, $M\,L^{-3}$.
ρ_0 reference value of fluid mass density (e.g., fresh water), $M\,L^{-3}$.
ρ^s mass concentration of solute component, $M\,L^{-3}$.
ρ_w mass concentration of water, $M\,L^{-3}$.
ϕ freshwater hydraulic head, L.
ω solute mass fraction, ρ^s/ρ.
ω_0 reference solute mass fraction, ρ^s/ρ.
∇ spatial del operator, L^{-1}.

ACKNOWLEDGMENTS. Support of the Water Authority of Western Australia is gratefully acknowledged. J. B. Yost of the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign was responsible for producing the images in Plate 1. Calculations for these images were performed on the NCSA's Cray X-MP/48 and Alliant FX-8 computers. The hardware used for image generation included an IRIS 3130 workstation. Time on the Cray was provided under a grant from the Academic Computing Center of the University of California, Riverside.

Garrison Sposito was the editor in charge of this paper. He thanks E.A. Sudicky and an anonymous reviewer for serving as technical referees.

REFERENCES

Ababou, R., L. W. Gelhar, and D. McLaughlin, Three-dimensional flow in random porous media, 2 vols., 833 pp., *Rep. 318,* Ralph M. Parsons Lab., Civil Eng. Dep., Mass. Inst. of Technol., Cambridge, 1988.

Adams, L. M., and T. W. Crockett, Modeling algorithm execution times on processor arrays, *Computer, 17*(7), 38–43, 1984.

Amdahl, G. M., Validity of the single processor approach to achieving large scale computing capabilities, *AFIPS Conf. Proc., 30,* 483–485, 1967.

Barlow, R. H., and D. J. Evans, Parallel algorithms for iterative solution to linear systems, *Comput. J., 25*(1), 56–60, 1982.

Barry, D. A., and G. Sposito, Application of the convection-dispersion model to solute transport in finite soil columns, *Soil Sci. Soc. Am. J., 52*(1), 3–9, 1988.

Barry, D. A., and G. Sposito, Three-dimensional statistical moment analysis of the Stanford/Waterloo Borden tracer data, *Water Resour. Res., 26*(8), 1735–1747, 1990.

Barry, D. A., J. Coves, and G. Sposito, On the Dagan model of solute transport in groundwater: Application to the Borden site, *Water Resour. Res., 24*(10), 1805–1817, 1988.

Baudet, G. M., Asynchronous iterative methods for multiprocessors, *J. Assoc. Comput. Mach., 25*(2), 226–244, 1978.

Bear, J., *Dynamics of Fluids in Porous Media,* 764 pp., Elsevier, New York, 1972.

Bear, J., and A. Verruijt, *Modeling Groundwater Flow and Transport,* 414 pp., D. Reidel, Hingham, Mass., 1987.

Behie, A., and P. A. Forsyth, Comparison of fast iterative methods for symmetric systems, *IMA J. Numer. Anal., 3,* 41–63, 1983.

Behie, A., and P. A. Forsyth, Incomplete factorization methods for fully implicit simulation of enhanced oil recovery, *SIAM J. Stat. Comput., 5*(3), 543–561, 1984.

Behie, A., and P. K. W. Vinsome, Block iterative methods for fully implicit reservoir simulation, *Soc. Pet. Eng. J., 22,* 659–668, 1982.

Bethke, C. M., A numerical model of compaction-driven groundwater flow and heat transfer and its application to the paleohydrology of intra-cratonic sedimentary basins, *J. Geophys. Res., 90*(B8), 6817–6828, 1985.

Bethke, C. M., W. J. Harrison, C. Upson, and S. P. Altaner, Supercomputer analysis of sedimentary basins, *Science, 239*(4387), 261–267, 1988.

Black, T. C., Concentration uncertainty for stochastic analysis of solute transport in a bounded, heterogeneous aquifer, 273 pp., Ph.D. thesis, Civil Eng. Dep., Stanford Univ., Stanford, Calif., 1988.

Black, T. C., and D. L. Freyberg, Stochastic modeling of vertically averaged concentration uncertainty in a perfectly stratified aquifer, *Water Resour. Res., 23*(6), 997–1004, 1987.

Black, T. C., and D. L. Freyberg, Simulation of one-dimensional correlated fields using a matrix-factorization moving average approach, *Math. Geol., 22*(1), 39–62, 1990.

Borden, R. C., and P. B. Bedient, Transport of dissolved hydrocarbons influenced by oxygen-limited biodegradation, 1, Theoretical development, *Water Resour. Res., 22*(13), 1973–1982, 1986.

Carnahan, C. L., Simulation of uranium transport with variable temperature and oxidation potential: The computer program THCC, *Rep. 21639,* Lawrence Berkeley Lab., Berkeley, Calif., 1986.

Chiang, C. Y., Numerical simulation of ground water contaminant transport on a supercomputer with injection-pumping networks using the modified MOC and MFE method, 148 pp., Ph.D. thesis, Rice Univ., Houston, Tex., 1985.

Chiang, C. Y., M. F. Wheeler, and P. B. Bedient, A modified method of characteristics technique and mixed finite elements method for simulation of groundwater solute transport, *Water Resour. Res., 25*(7), 1541–1549, 1989.

Chu, S.-Y., and G. Sposito, A derivation of the macroscopic solute transport equation for homogeneous, saturated, porous media, *Water Resour. Res., 16*(3), 542–546, 1980.

Chu, S.-Y., and G. Sposito, a derivation of the macroscopic solute transport equation for homogeneous, saturated, porous media, 2, Reactive solutes at low concentrations, *Water Resour. Res., 17*(3), 333–336, 1981.

Concus, P., G. H. Golub, and D. P. O'Leary, A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations, in *Sparse Matrix Computations,* edited by J. R. Bunch and D. J. Rose, pp. 309–332, Academic, San Diego, Calif., 1976.

Cox, D. R., and H. D. Miller, *The Theory of Stochastic Processes,* 398 pp., Chapman and Hall, London, 1965.

Cox, D. R., and E. J. Snell, A general definition of residuals, *J. R. Stat. Soc., Ser. B., 30*(2), 248–265, 1968.

Crump, K. S., Numerical inversion of Laplace transforms using a Fourier series approximation, *J. Assoc. Comput. Mach., 23*(1), 89–96, 1976.

Dagan, G., Stochastic modeling of groundwater flow by unconditional and conditional probabilities, 2, The solute transport, *Water Resour. Res., 18*(4), 835–848, 1982.

Dagan, G., Solute transport in heterogeneous porous formations, *J. Fluid Mech, 145,* 151–177, 1984.

Dagan, G., Statistical theory of groundwater flow and transport: Pore to laboratory, laboratory to formation, and formation to regional scale, *Water Resour. Res., 22*(9), 120S–134S, 1986.

Dagan, G., Theory of solute transport by groundwater, *Annu. Rev. Fluid Mech., 19,* 183–215, 1987.

Dagan, G., Time-dependent macrodispersion for solute transport in anisotropic heterogeneous aquifers, *Water Resour. Res., 24*(9), 1491–1500, 1988.

Denning, P. J., Speeding up parallel processing, *Am. Sci., 76*(4), 347–349, 1988.

Dongarra, J. J., F. G. Gustavson, and A. Karp, Implementing linear algebra algorithms for dense matrices on a vector pipeline machine, *SIAM Rev., 26*(1), 91–112, 1984.

Efron, B., *The Jackknife, the Bootstrap and Other Resampling Plans,* 92 pp., SIAM, Philadelphia, Pa., 1982.

Efron, B., Computer-intensive methods in statistical regression, *SIAM Rev., 30*(3), 421–449, 1988.

Emmen, A. H. L., A survey of vector and parallel processors, in *Algorithms and Applications on Vector and Parallel Computers,* edited by H. J. J. te Riele, T. J. Dekker, and H. A. van der Vorst, pp. 151–154, Elsevier, New York, 1987.

Evans, D. J., Parallel numerical algorithms for linear systems, in *Parallel Processing Systems,* edited by D. J. Evans, pp. 357–383, Cambridge University Press, New York, 1982.

Evans, D. J., and M. Hatzopoulos, A parallel linear system solver, *Int. J. Comput. Math., B7*(3), 227–238, 1979.

Evans, D. J., A. Hadjidimos, and D. Noutsos, The parallel solution of banded linear equations by the new quadrant interlocking factorization (Q.I.F.) method, *Int. J. Comput. Math., B9*(2), 151–161, 1981.

Fisher, J. A., The VLIW machine: A multiprocessor for compiling scientific code, *IEEE Comput., 17*(7), 45–53, 1984.

Freyberg, L., A natural gradient experiment on solute transport in a sand aquifer, 2, Spatial moments and the advection and dispersion of nonreactive tracers, *Water Resour. Res., 22*(13), 2031–2046, 1986.

Frind, E. O., Seawater intrusion in continuous coastal aquifer-aquitard systems, *Adv. Water Resour., 5,* 89–97, 1982*a*.

Frind, E. O., Simulation of long-term transient density-dependent transport in groundwater, *Adv. Water Resour., 5,* 73–88, 1982*b*.

Frind, E. O., The principal direction technique: A new approach to groundwater contaminant transport modelling, in *Proceedings of the 4th International Conference on Finite Elements in Water Resources,* pp. 13.25–13.42, Springer-Verlag, New York, 1982*c*.

Frind, E. O., and G. F. Pinder, The principal direction technique for solution of the advection-dispersion equation, in *Proceedings of the 10th IMACS Conference,* pp. 305–313, Scientific Computing, North-Holland, Amsterdam, 1982.

Frind, E. O., E. A. Sudicky, and S. L. Schellenberg, Micro-scale modelling in the study of plume evolution in heterogeneous media, *Stochastic Hydrol. Hydraul., 1*(4), 263–279, 1987.

Frind, E. O., E. A. Sudicky, and J. W. Molson, Three-dimensional simulation of organic transport with aerobic biodegradation, paper presented at IAHS 3rd Scientific Assembly, Int. Assoc. of Hydrol. Sci., Baltimore, Md., May 10–19, 1989.

Garabedian, S. P., L. W. Gelhar, and M. A. Celia, Large-scale dispersive transport in aquifers: Field experiments and reactive solute transport theory, *Rep. 315,* 290 pp., Ralph M. Parsons Lab., Civil Eng. Dep., Mass. Inst. of Technol., Cambridge, 1988.

Garder, A. O., D. W. Peaceman, and A. L. Pozzi, Numerical calculation of multidimensional miscible displacement by the method of characteristics, *Soc. Pet. Eng. J., 4*(1), 26–36, 1964.

Gardiner, C. W., *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences,* 442 pp., Springer-Verlag, New York, 1983.

Gelhar, L. W., Stochastic subsurface hydrology from theory to applications, *Water Resour. Res., 22*(9), 135S–145S, 1986.

Gelhar, L. W., and C. L. Axness, Three-dimensional stochastic analysis of macrodispersion in aquifers, *Water Resour. Res., 19*(1), 161–180, 1983.

Gelhar, L. W., A. L. Gutjahr, and R. L. Naff, Stochastic analysis of macrodispersion in a stratified aquifer, *Water Resour. Res., 15*(6), 1387–1397, 1979.

Goltz, M. N., and P. V. Roberts, Interpreting organic solute transport data from a field experiment using physical nonequilibrium models, *J. Contam. Hydrol., 1*(1), 77–93, 1986.

Gray, H. L., and W. R. Schucany, *The Generalized Jackknife Statistic,* 308 pp., Marcel Dekker, New York, 1972.

Hager, W. W., *Applied Numerical Linear Algebra,* 424 pp., Prentice-Hall, Englewood Cliffs, N. J., 1988.

Hassanizadeh, S. M., and T. Leijnse, On the modeling of brine trasnport in porous media, *Water Resour. Res., 24*(3), 321–330, 1988.

Herbert, A. W., C. P. Jackson, and D. A. Lever, Coupled groundwater flow and solute transport with fluid density strongly dependent on concentration, *Water Resour. Res., 24*(1), 1781–1795, 1988.

Hestenes, M. R., and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand., Sect. B, 49,* 409–436, 1952.

Hockney, R. W., and C. R. Jesshope, *Parallel Computers: Architecture Programming and Algorithms,* 423 pp., Adam Hilger, Bristol, England, 1981.

Hockney, R. W., and C. R. Jesshope, *Parallel Computers 2:*

Architecture Programming and Algorithms, 625 pp., Adam Hilger, Bristol, England, 1988.

Huyakorn, P. S., and C. Taylor, Finite element models for coupled groundwater flow and convective dispersion, in *Finite Elements in Water Resources,* edited by W. G. Gray, G. F. Pinder, and C. A. Brebbia, pp. 1.131–1.151, Pentech, London, 1977.

Huyakorn, P. S., P. F. Anderson, J. W. Mercer, and H. O. White, Saltwater intrusion in aquifers: Development and testing of a three-dimensional finite element model, *Water Resour. Res., 23*(2), 293–312, 1987.

Illangasekare, T. H., and P. Döll, A discrete kernel method of characteristics model of solute transport in water table aquifers, *Water Resour. Res., 25*(5), 857–867, 1989.

Illangasekare, T. H., and H. J. Morel-Seytoux, A finite element "discrete kernel generator" for efficient groundwater management, in *Finite Elements in Water Resources,* edited by C. A. Brebbia, W. G. Gray, and G. F. Pinder, pp. 1.195–1.212, Pentech, London, 1978.

Journel, A. G., and C. J. Huijbregts, *Mining Geostatistics,* 600 pp., Academic, San Diego, Calif., 1978.

Kershaw, D., Solution of single tridiagonal linear systems and vectorization of the ICCG algorithm on the Cray-1, in *Parallel Computations,* edited by G. Rodrigue, pp. 85–99, Academic, San Diego, Calif., 1982.

Kool, J. B., and J. C. Parker, Analysis of the inverse problem for transient unsaturated flow, *Water Resour. Res., 24*(6), 817–830, 1988.

Kuck, D. J., E. S. Davidson, D. H. Lawrie, and A. H. Sameh, Parallel supercomputing today and the Cedar approach, *Science, 231*(4741), 967–974, 1986.

Kung, H. T., Synchronized and asynchronous parallel algorithms for multiprocessors, in *Algorithms and Complexity,* edited by J. F. Traub, pp. 153–200, Academic, San Diego, Calif., 1976.

Lee, C. H., and R. T. Cheng, On seawater encroachment in coastal aquifers, *Water Resour. Res., 10*(5), 1039–1043, 1974.

Liu, C. W., and T. N. Narasimhan, Redox-controlled multiple-species reactive chemical transport, 1, Model development, *Water Resour. Res., 25*(5), 869–882, 1989*a*.

Liu, C. W., and T. N. Narasimhan, Redox-controlled multiple-species reactive chemical transport, 2, Verification and application, *Water Resour. Res., 25*(5), 883–910, 1989*b*.

Lorensen, W. E., and H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *Comput. Graphics, 21*(4), 163–169, 1987.

MacFarlane, D. S., J. A. Cherry, R. W. Gillham, and E. A. Sudicky, Migration of contaminants in groundwater at a landfill: A case study, 1, Groundwater flow and plume delineation, *J. Hydrol, 63,* 1–29, 1983.

MacQuarrie, K. T. B., and E. A. Sudicky, Simulation of biodegradable organic contaminants in groundwater, 2, Plume behavior in uniform and random flow fields, *Water Resour. Res., 26,* 223–239, 1990.

MacQuarrie, K. T. B., E. A. Sudicky, and E. O. Frind, Simulation of biodegradable organic contaminants in groundwater, 1, Numerical formulation in principal directions, *Water Resour. Res., 26,* 207–222, 1990.

Mantoglou, A., and J. Wilson, The turning bands method for simulation of random fields using line generation by a spectral method, *Water Resour. Res., 18*(5), 1379–1394, 1982.

Marquardt, D. W., An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Ind. Appl. Math., 11*(2), 431–441, 1963.

McCormick, S. F. (Ed.), *Multigrid Methods,* 282 pp., SIAM, Philadelphia, Pa., 1987.

McCormick, S. F. (Ed.), Multigrid methods: Theory, applications, and supercomputing, *Lecture Notes in Pure and Applied Mathematics,* vol. 110, 644 pp., Marcel Dekker, New York, 1988.

Meyer, P. D., A. J. Valocchi, S. F. Ashby, and P. E. Saylor, A numerical investigation of the conjugate gradient method as applied to three-dimensional groundwater flow problems in randomly heterogeneous porous media, *Water Resour. Res., 25*(6), 1440–1446, 1989.

Mishra, S., and J. C. Parker, Parameter estimation for coupled unsaturated flow and transport, *Water Resour. Res., 25*(3), 385–396, 1989.

Mosteller, F., and J. W. Tukey, Data analysis, including statistics, in *Handbook of Social Psychology,* vol. 2, edited by G. Lindzey and E. Aronson, pp. 80–203, Addison-Wesley, Reading, Mass., 1968.

Naff, R. L., T.-C. J. Yeh, and M. W. Kemblowski, A note on the recent natural gradient tracer test at the Borden site, *Water Resour. Res., 24*(12), 2099–2103, 1988.

Narasimhan, T. N., A. F. White, and T. Tokunaga, Groundwater contamination from an inactive uranium mill tailings pile, 2, Application of a dynamic mixing model, *Water Resour. Res., 22*(13), 1820–1834, 1986.

Neuman, S. P., C. L. Winter, and C. M. Newman, Stochastic theory of field-scale Fickian dispersion in anisotropic porous media, *Water Resour. Res., 23*(3), 453–466, 1987.

Newport, B. D., Salt water intrusion in the United States, *Rep. 600/8-77-011,* 30 pp., U.S. Environ. Prot. Agency, Washington, D. C., 1977.

Noye, J., Finite difference methods for partial differential equations, in *Numerical Solutions of Partial Differential Equations,* edited by J. Noye, pp. 1–137, North-Holland, Amsterdam, 1982.

Ortega, J. M., and R. G. Voigt, Solution of partial differential equations on vector and parallel computers, *SIAM Rev., 27*(2), 149–240, 1985.

Pelka, W., and A. Peters, Finite element ground-water models implemented on vector computers, *Int. J. Numer. Meth. Fluids, 6,* 913–925, 1986.

Quenouille, M. H., Notes on bias in estimation, *Biometrika, 43*(3-4), 353–360, 1956.

Ripley, B. D., *Spatial Statistics,* 252 pp., John Wiley, New York, 1981.

Roberts, P. V., and D. M. Mackay (Eds.), A natural gradient experiment on solute transport in a sand aquifer, *Tech. Rep. 292,* 229 pp., Civil Eng. Dep., Stanford Univ., Stanford, Calif., 1986.

Saad, Y., Practical use of polynomial preconditionings for the conjugate gradient method, *SIAM J. Sci. Stat. Comput., 6*(4), 865–881, 1985.

Sameh, A. H., and D. J. Kuck, On stable parallel linear system solvers, *J. Assoc. Comput. Mach., 25*(1), 81–91, 1978.

Samper, F. J., and S. P. Neuman, Estimation of spatial covariance structures by adjoint state maximum likelihood cross validation, 1, Theory, *Water Resour. Res., 25*(3), 351–362, 1989.

Segol, G., and G. P. Pinder, Transient simulation of saltwater intrusion in southeastern Florida, *Water Resour. Res., 12*(1), 65–70, 1976.

Segol, G., G. F. Pinder, and W. G. Gray, A Galerkin finite element technique for calculating the transient position of the saltwater front, *Water Resour. Res., 11*(2), 343–347, 1975.

Seitz, C. L., and J. Matisoo, Engineering limits on computer performance, *Phys. Today, 37*(5), 38–45, 1984.

Sharp, J. A., *An Introduction to Distributed and Parallel Processing,* 174 pp., Blackwell Scientific, Oxford, 1987.

Shinozuka, M., and C.-M. Jan, Digital simulation of random processes and its applications, *J. Sound Vibrat., 25*(1), 111–128, 1972.

Sposito, G., and D. A. Barry, On the Dagan model of solute transport in groundwater: Foundational aspects, *Water Resour. Res., 23*(10), 1867–1875, 1987.

Sposito, G., V. K. Gupta, and R. N. Bhattacharya, Foundational

theories of solute transport in porous media: A critical review, *Adv. Water Resour., 2*, 59–68, 1979.

Sposito, G., W. A. Jury, and V. K. Gupta, Fundamental problems in the stochastic convection-dispersion model of solute transport in aquifers and field soils, *Water Resour. Res., 22*(1), 77–88, 1986.

Stone, H. S., An efficient parallel algorithm for the solution of a tridiagonal linear system of equations, *J. Assoc. Comput. Mach., 20*(1), 27–38, 1973.

Sudicky, E. A., A natural gradient experiment on solute transport in a sand aquifer: Spatial variability of hydraulic conductivity and its role in the dispersion process, *Water Resour. Res., 22*(13), 2069–2082, 1986.

Sudicky, E. A., The Laplace transform Galerkin technique: A time-continuous finite element theory and application to mass transport in groundwater, *Water Resour. Res., 25*(8), 1833–1846, 1989*a*.

Sudicky, E. A., The Laplace transform Galerkin technique for large-scale simulation of mass transport in discretely fractured porous formations (abstract), *Eos Trans. AGU, 70*(43), 1082, 1989*b*.

Sudicky, E. A., J. A. Cherry, and E. O. Frind, Migration of contaminants in groundwater at a landfill: A case study, 4, A natural-gradient dispersion test, *J. Hydrol., 63*, 81–108, 1983.

Sudicky, E. A., S. L. Schellenberg, and K. T. B. MacQuarrie, Assessment of the behaviour of conservative and biodegradable solutes in heterogeneous porous media, in *Dynamics of Fluids in Hierarchical Porous Formations*, edited by J. H. Cushman, Academic, San Diego, Calif., 1990.

Taylor, C., and P. S. Huyakorn, Three-dimensional groundwater flow with convective dispersion, in *Finite Elements in Fluids*, vol. 3, edited by R. H. Gallagher, pp. 311–321, Wiley, New York, 1978.

Tichelaar, B. W., and L. J. Ruff, How good are our best models? *Eos Trans. AGU, 70*(20), 593, 605–606, 1989.

Tompson, A. F. B., E. G. Vomvoris, and L. W. Gelhar, Numerical solution of solute transport in randomly heterogeneous porous media: Motivation, model development, and application, *Rep. 316*, 114 pp., Ralph M. Parsons Lab., Civil Eng. Dep., Mass. Inst. of Technol., Cambridge, 1988.

Tompson, A. F. B., R. Ababou, and L. W. Gelhar, Implementation of the three-dimensional turning bands random field generator, *Water Resour. Res., 25*(10), 2227–2243, 1989.

Traub, J. F., Iterative solution of tridiagonal systems on parallel or vector computers, in *Complexity of Sequential and Parallel Numerical Algorithms*, edited by J. F. Traub, pp. 49–82, Academic, San Diego, Calif., 1973.

Tukey, J. W., Bias and confidence in not-quite large samples (abstract), *Ann. Math. Stat., 29*(2), 614, 1958.

van Kampen, N. G., *Stochastic Processes in Physics and Chemistry*, 419 pp., North-Holland, Amsterdam, 1981.

Voigt, R. G., The influence of vector computer architecture on numerical algorithms, in *High Speed Computer and Algorithm Organization*, edited by D. J. Kuck, D. H. Lawrie, and A. H. Sameh, pp. 229–244, Academic, San Diego, Calif., 1977.

Voigt, R. G., D. Gottleib, and M. Y. Hassaini (Eds.), *Spectral Methods for Partial Differential Equations*, 276 pp., SIAM, Philadelphia, Pa., 1984.

Wallis, J. R., Hydrology—The computer revolution continues, *Rev. Geophys., 25*(2), 101–105, 1987.

Walsh, M. P., S. L. Bryant, S. R. Schechter, and L. W. Lake, Precipitation and dissolution of solids attending flow through porous media, *AIChE J., 30*(2), 317–328, 1984.

Winter, C. L., C. M. Newman, and S. P. Neuman, A perturbation expansion for diffusion in a random velocity field, *SIAM J. Appl. Math., 44*(2), 411–424, 1984.

Yeh, W. W.-G., Review of parameter identification procedures in groundwater hydrology: The inverse problem, *Water Resour. Res., 22*(1), 95–108, 1986.

Yeh, W. W.-G., and C. Wang, Identification of aquifer dispersivities: Methods of analysis and parameter uncertainty, *Water Resour. Bull., 23*(4), 369–580, 1987.

Young, D. M., and R. T. Gregory, *A Survey of Numerical Mathematics*, vol. II, 659 pp., Addison-Wesley, Reading, Mass., 1972.

D. A. Barry, Centre for Water Research, University of Western Australia, Nedlands, Australia 6009.