WATER RESOURCES RESEARCH, VOL. 39, NO. 6, 1158, doi:10.1029/2001WR000586, 2003

Numerical error in groundwater flow and solute transport simulation

Juliette A. Woods,[1,2] Michael D. Teubner,[1] Craig T. Simmons,[3] and Kumar A. Narayan[4]

Received 4 April 2001; revised 27 April 2002; accepted 16 July 2002; published 19 June 2003.

[1] Models of groundwater flow and solute transport may be affected by numerical error, leading to quantitative and qualitative changes in behavior. In this paper we compare and combine three methods of assessing the extent of numerical error: grid refinement, mathematical analysis, and benchmark test problems. In particular, we assess the popular solute transport code SUTRA [*Voss*, 1984] as being a typical finite element code. Our numerical analysis suggests that SUTRA incorporates a numerical dispersion error and that its mass-lumped numerical scheme increases the numerical error. This is confirmed using a Gaussian test problem. A modified SUTRA code, in which the numerical dispersion is calculated and subtracted, produces better results. The much more challenging Elder problem [*Elder*, 1967; *Voss and Souza*, 1987] is then considered. Calculation of its numerical dispersion coefficients and numerical stability show that the Elder problem is prone to error. We confirm that Elder problem results are extremely sensitive to the simulation method used. *INDEX TERMS:* 1832 Hydrology: Groundwater transport; 1894 Hydrology: Instruments and techniques; *KEYWORDS:* variable density, Elder problem, benchmarking, salinity, contaminant transport

Citation: Woods, J. A., M. D. Teubner, C. T. Simmons, and K. A. Narayan, Numerical error in groundwater flow and solute transport simulation, *Water Resour. Res.*, *39*(6), 1158, doi:10.1029/2001WR000586, 2003.

1. Introduction

[2] Groundwater modeling is now an important part of many hydrogeological investigations. Models are used to explore physical processes, to summarize what is known about field sites, and to investigate different management practices. In all of these applications, an understanding of model accuracy is essential.

[3] One factor affecting accuracy is numerical error, which occurs in all computational simulations. Numerical error can lead to quantitative and even qualitative changes in simulation results, potentially affecting the management of field sites [*Simmons et al.*, 1999; *Woods et al.*, 1998, 1999].

[4] There are many types of numerical error. If the chosen grid spacing and time step length are too large, small errors may grow to dominate part of a simulation, resulting in numerical instability [*Noye*, 1978; *Ferziger and Perić*, 1999]. This often leads to physically unreasonable results and problems with convergence. Another kind of numerical error is numerical dispersion [*Noye and Hayman*, 1985; *Gresho and Sani*, 1998]. Numerical dispersion is insidious because it mimics hydrodynamic dispersion (a heuristic description of various physical processes [*Bear*, 1972]) producing smooth results that may seem plausible. Round-off error occurs because a computer stores each number as a finite sequence of digits, but unless very fine grids or very small time steps are chosen, it is usually negligible for stable solution schemes [*May and Noye*, 1984]. Other numerical errors may be introduced through the use of iterative solvers, approximations of boundary conditions and the solution of nonlinear PDEs, such as those arising from variable-density flows.

[5] The amount of numerical error present in a given simulation depends on both the code developer and code user. The code developer chooses the numerical approximation of the PDE(s) while the code user chooses the grid-spacing and time step regime. The assessment of numerical error is important to both.

[6] This assessment is usually an implicit part of model verification, which is typically investigated through one of three main methods: (1) grid refinement, (2) mathematical analysis, and (3) test problems. Grid refinement studies are reasonably common [*Kolditz et al.*, 1997; *Holzbecher*, 1998; *Diersch*, 1996; *Frolkovič and Schepper*, 2000; *Schincariol et al.*, 1994] but tell us little about the general validity of a code. The second approach, that of mathematical analysis, is used fairly rarely for groundwater problems. Two recent papers on the analysis of numerical error are those of *Benson et al.* [1998] and *Lal* [2000]: *Benson et al.* [1998] consider the truncation error inherent in some types of velocity calculation, while *Lal* [2000] uses Fourier analysis to evaluate errors in MODFLOW. However, within the groundwater literature the most common method of verification is the test problem. Unfortunately, few satisfy what we believe to be the three criteria of a good test problem: that it is well-defined, well-understood and diffi-

[1] Department of Applied Mathematics, Adelaide University, Adelaide, South Australia, Australia.

[2] Now at Institute for Computational Engineering and Sciences, ACES 6.444, University of Texas at Austin, Austin, Texas, USA.

[3] School of Chemistry, Physics and Earth Sciences, Flinders University of South Australia and Centre for Groundwater Studies, Adelaide, South Australia, Australia.

[4] Davies Laboratory, CSIRO Land and Water, Townsville, Queensland, Australia.

0043-1397/03/2001WR000586$09.00

cult to simulate. By 'well-defined,' we mean that the initial conditions, boundary conditions and hydraulic parameters are specified exactly and unambiguously; by "well-understood," we mean that the resulting flow is known precisely; and by "difficult to simulate" we mean that numerical errors inherent in a code lead to large quantitative or qualitative differences between the expected and simulated results unless extremely accurate solution techniques are used.

[7] In this paper, we demonstrate that numerical error can best be assessed through a combination of approaches, as each approach sheds light on the others. We introduce a numerical analysis technique from finite difference methods (FDMs) which improves our understanding of grid refinement studies and test simulations. As an illustration, we consider a well-known code for the simulation of variable-density groundwater flow and solute transport: SUTRA [*Voss*, 1984]. SUTRA is chosen because it is a popular, well-documented, open source code that can be considered typical of many groundwater codes. However, results presented here have much broader application as most codes will probably suffer from the same problems. We show that even small (third order) changes to its solution technique can lead to large changes in simulation results.

2. Numerical Analysis

[8] In this section, we demonstrate how numerical analysis can provide insights into the accuracy of a simulation code. To make this analysis tractable, we need to make some simplifying assumptions. It is assumed that (1) transport is of an inert solute such as salt, (2) the domain being simulated is two-dimensional, (3) the porous medium is saturated, homogeneous and isotropic, (4) fluid viscosity is constant, (5) there is no hydrodynamic dispersion, only diffusion, and (6) the direction of gravity coincides with the y axis. These assumptions are fairly strict and few real-world simulations conform to them. However, our analysis should at least provide a guide to a code's behavior for more complicated simulations.

[9] Given these assumptions, the following equations for fluid flow and solute transport need to be solved:

$$\rho S\frac{\partial p}{\partial t}+\varepsilon\rho_{ref}\gamma\frac{\partial c}{\partial t}=-\nabla\cdot(\rho\mathbf{q}), \tag{1}$$

$$\rho\frac{\partial c}{\partial t}=-\frac{1}{\varepsilon}\rho\mathbf{q}\cdot\nabla c+D\nabla\cdot(\rho\nabla c), \tag{2}$$

where c is dimensionless concentration (mass of solute as a proportion of the total mass of brine), D is the molecular diffusivity (m^2s^{-1}), p is pressure ($kgm^{-1}s^{-2}$), S is storativity (Pa^{-1}), γ is the dimensionless coefficient of density change with concentration, ε is porosity, and ρ_{ref} is some reference density (kgm^{-3}). Darcy velocity $\mathbf{q}$ (ms^{-1}) and density ρ are given by [*Voss*, 1984]:

$$\mathbf{q}=-\frac{k}{\mu}(\nabla p-\rho\mathbf{g}), \tag{3}$$

$$\rho=\rho_{ref}(1+\gamma c), \tag{4}$$

where $\mathbf{g} = (0, -|g|)$ is acceleration due to gravity (ms^{-2}), k is permeability (m^2) and μ is the dynamic viscosity ($kgm^{-1}s^{-1}$). This formulation is expressed in terms of pressure and dimensionless concentration. Other formulations exist [*Diersch*, 1996; *Holzbecher*, 1998] but are generally regarded as equivalent.

2.1. SUTRA's Solution Method

[10] SUTRA solves equations (1) and (2) using a hybrid finite element method (FEM), the details of which are given by *Voss* [1984]. The standard FEM divides a modeled region into elements and nodes. Dependent variables are approximated by piecewise-continuous functions, i.e., a continuous variable $f(x, y, t)$ is replaced by a sum of discretized (nodal) values multiplied by weighting functions ϕ_l. The approximation is continuous over each element. That is,

$$f(x,y,t)\approx\sum_{l=1}^{N}f_l(t)\,\phi_l(x,y), \tag{5}$$

where the l subscript refers to a particular node and N is the total number of nodes.

[11] SUTRA modifies this standard FEM in two main ways. Firstly, it treats the time derivatives in equations (1) and (2) differently. Instead of discretizing

$$\frac{\partial f}{\partial t}(x,y,t)\approx\sum_{l=1}^{N}\frac{\partial f_l}{\partial t}(t)\,\phi_l(x,y), \tag{6}$$

the following is used:

$$\frac{\partial f}{\partial t}(x,y,t)\approx\frac{\partial f_m}{\partial t}(t), \tag{7}$$

where m denotes the specific, single node at which $\partial f/\partial t$ is being evaluated. That is, the time derivative is evaluated at a single point (as if all the mass were at that point) rather than averaged over all adjoining nodes. This technique is known as "mass-lumping." Mass lumping enables marginally faster calculation of the matrix equation and theoretically leads to a more stable system, but doubts have been raised about its validity and accuracy [*Becker et al.*, 1986; *Gresho and Sani*, 1998].

[12] SUTRA also treats one of the velocity terms differently. The standard discretization of the velocity in terms of pressure and density leads to

$$q\approx-\frac{k}{\mu}\sum_{l=1}^{N}(p_l\nabla\phi_l-\rho_l\phi_l\mathbf{g}). \tag{8}$$

However, this means that the pressure term changes with $\nabla\phi_l$ while the density term changes with $\phi_l\mathbf{g}$. The two terms, when put together, do not vary consistently over an element [*Benson et al.*, 1998]. To overcome this, Voss introduces the following consistent velocity discretization:

$$q\approx-\frac{k}{\mu}\sum_{l=1}^{N}(p_l\nabla\phi_l-\mathbf{J}\rho_l|\nabla\phi_l|\mathbf{g}), \tag{9}$$

where $\mathbf{J}$ is the element's Jacobian:

$$\mathbf{J}=\begin{vmatrix}\dfrac{\partial x}{\partial\zeta} & \dfrac{\partial y}{\partial\zeta}\\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta}\end{vmatrix}, \tag{10}$$

given that (ζ, η) are an element's "local coordinates" (i.e., each FEM element can be mapped from global (x, y) space to local (ζ, η) space, where each element becomes a square such that $-1 < \zeta < 1$ and $-1 < \eta < 1$). The term "consistent velocity discretization" is perhaps unfortunate, as the standard discretization method (8) is consistent with usual finite element notions; what *Voss* [1984] wishes to convey is that equation (9) is consistent physically. *Benson et al.* [1998] outline other methods of achieving this physical consistency. Thus SUTRA is a hybrid finite element code employing mass-lumping and a "consistent-velocity" technique.

2.2. Numerical Dispersion

[13] Our first analytic technique is the calculation of the modified equivalent partial differential equation (MEPDE). The MEPDE is, in essence, the PDE which an approximate equation is actually solving [*Warming and Hyett*, 1974; *Noye and Hayman*, 1985]. However, the MEPDE technique was developed for finite difference methods and has not, to our knowledge, been applied before to a finite element code, such as SUTRA. This problem is circumvented as follows: the discretized, approximate SUTRA equations are applied to a uniform rectangular grid and the integral terms evaluated exactly. This reduces the finite element method to an equivalent finite difference method for which the MEPDE can be calculated. This process is developed in the following manner: each node within the rectangular finite element grid has eight additional nodes surrounding it that lead to non-zero links within the global finite element matrix. In finite difference terms, the node is designated (i, j) and the eight surrounding nodes will be $(i-1, j-1)$, $(i-1, j)$,......, to $(i+1, j+1)$. This means that the solution for any parameter τ at the nodal point (i, j), i.e., $\tau_{i,j}$, will be a function of the values of τ at each of the surrounding points $\tau_{i-1,j-1}$, etc. Mathematically, each of the values $\tau_{i-1,j-1}$, etc. can be represented as a Taylor series about the point (i, j), e.g.,

$$\begin{aligned}\tau_{i-1,j} =& \tau_{i,j} - \frac{\Delta x}{1!}\frac{\partial \tau}{\partial x}\bigg|_{i,j} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 \tau}{\partial x^2}\bigg|_{i,j} - \frac{(\Delta x)^3}{3!}\frac{\partial^3 \tau}{\partial x^3}\bigg|_{i,j} \\ &+ \sum_{k=4}^{\infty}\frac{(-\Delta x)^k}{k!}\frac{\partial^k \tau}{\partial x^k}\bigg|_{i,j}. \end{aligned} \tag{11}$$

By substituting the Taylor series expansion for each of the eight surrounding nodes into the equation that is used to solve for $\tau_{i,j}$, an equivalent partial differential equation (EPDE) is developed.

[14] The EPDE includes derivatives in both time and space. The unwanted time-derivatives are eliminated through the repeated differentiation of the original PDE so that the MEPDE is obtained.

[15] For the purpose of this calculation only, we assume that density is constant (i.e., $\gamma = 0$), so that equations (1) and (2) become:

$$S\frac{\mu}{k}\frac{\partial p}{\partial t} - \nabla^2 p = 0, \tag{12}$$

$$\frac{\partial c}{\partial t} + \mathbf{v}\cdot\nabla c - D\nabla^2 c = 0, \tag{13}$$

where $\mathbf{v} = (u, v) = \mathbf{q}/\varepsilon$ is the fluid velocity. Then, for example, differentiating Equation (13) with respect to x yields

$$\frac{\partial^2 c}{\partial x \partial t} = -u\frac{\partial^2 c}{\partial x^2} - v\frac{\partial^2 c}{\partial x \partial y} + D\frac{\partial^3 c}{\partial x^3} + D\frac{\partial^3 c}{\partial x \partial^2 y}, \tag{14}$$

and differentiating it with respect to y gives

$$\frac{\partial^2 c}{\partial y \partial t} = -u\frac{\partial^2 c}{\partial x \partial y} - v\frac{\partial^2 c}{\partial y^2} + D\frac{\partial^3 c}{\partial y^3} + D\frac{\partial^3 c}{\partial^2 x \partial y}. \tag{15}$$

Differentiating (13) with respect to t and substituting (14) and (15) yields:

$$\frac{\partial^2 c}{\partial t^2} = -u\frac{\partial^2 c}{\partial x \partial t} - v\frac{\partial^2 c}{\partial t \partial y} + D\frac{\partial^3 c}{\partial t \partial x^2} + D\frac{\partial^3 c}{\partial t \partial y}, \tag{16}$$

$$= u^2\frac{\partial^2 c}{\partial x^2} + 2uv\frac{\partial^2 c}{\partial x \partial y} + v^2\frac{\partial^2 c}{\partial y^2} + H.O.T. \tag{17}$$

These spatial equivalents of the time derivatives are substituted back into the EPDE to yield the MEPDE.

[16] The MEPDE for the pressure equation is

$$\begin{aligned} S\frac{\mu}{k}\frac{\partial p}{\partial t} - \nabla^2 p =& -\frac{(\Delta x^2)}{12}(6\sigma_x - 1)\frac{\partial^4 p}{\partial x^4} - \frac{(\Delta y)^2}{12}(6\sigma_y - 1)\frac{\partial^4 p}{\partial y^4} \\ 2 \qquad &+ \frac{\Delta x \Delta y}{2\sqrt{\sigma_x \sigma_y}}\left[\sigma_x\left(\sigma_y + \frac{1}{3}\right) + \sigma_y\left(\sigma_y + \frac{1}{3}\right)\right]\frac{\partial^4 p}{\partial x^2 \partial y^2} \\ &+ O\{5\}, \end{aligned} \tag{18}$$

where $\sigma_x = \frac{k}{\mu S}\frac{\Delta t}{(\Delta x)^2}$ and $\sigma_y = \frac{k}{\mu S}\frac{\Delta t}{(\Delta y)^2}$. The MEPDE for the concentration equation is

$$\begin{aligned} \frac{\partial c}{\partial t} + \mathbf{v}\cdot\nabla c - D\nabla^2 c =& D^N_{xx}\frac{\partial^2 c}{\partial x^2} + D^N_{xy}\Delta t\frac{\partial^2 c}{\partial x \partial y} + D^N_{yy}\frac{\partial^2 c}{\partial y^2} \\ & - \frac{1}{\Delta t}\left[\Delta x^3\left(\frac{1}{6}r_x + r_x s_x + \frac{1}{3}r_x^3\right)\frac{\partial^3 c}{\partial x^3}\right. \\ & - \Delta x^2 \Delta y\left(\frac{1}{2}r_y + r_y s_x + r_x^2 r_y\right)\frac{\partial^3 c}{\partial x^2 \partial y} \\ & - \Delta x \Delta y^2\left(\frac{1}{2}r_x + r_x s_y + r_x r_y^2\right)\frac{\partial^3 c}{\partial x \partial y^2} \\ & \left. -\Delta y^3\left(\frac{1}{6}r_y + r_y s_y + \frac{1}{3}r_y^3\right)\frac{\partial^3 c}{\partial y^3}\right] + O\{4\}, \end{aligned} \tag{19}$$

where the $O\{4\}$ and $O\{5\}$ notation refers to the order of the largest of the remaining terms, $r_x = u\Delta t/\Delta x$, $r_y = v\Delta t/\Delta y$, $s_x = D\Delta t/(\Delta x)^2$ and $s_y = D\Delta t/(\Delta y)^2$. Note that the σ, r and s terms are the Courant and diffusion numbers for the equations. Also, $D^N_{xx} = \frac{1}{2}r_x^2\Delta x^2/\Delta t$, $D^N_{xy} = r_x r_y \Delta x \Delta y/\Delta t$, and $D^N_{yy} = \frac{1}{2}{r_y}^2\Delta y^2/\Delta t$ are the coefficients of numerical dispersion, i.e., these are unwanted terms within the approximate equation which induce artificial dispersion of solute. The numerical dispersion terms are of the same order as terms being solved for on the left-hand-side of the equation.

[17] From this it can be seen that the pressure equation matches the desired equation up to and including all third-derivative terms. However, there is significant numerical dispersion within the concentration equation. The extent of

numerical dispersion varies with the square of the velocity and with the time step length Δt.

[18] We also investigate the effect of mass-lumping on the MEPDE. When SUTRA is reformulated without mass-lumping of the time-derivatives, the MEPDE changes. The amount of numerical dispersion is unchanged, but the following third-order terms drop out:

$$\frac{1}{6\Delta t}\left[r_x(\Delta x)^3\frac{\partial^3 c}{\partial x^3}+r_x(\Delta x)^2\Delta y\frac{\partial^3 c}{\partial^2 x\partial y}\right.$$
$$\left.+r_y\Delta x(\Delta y)^2\frac{\partial^3 c}{\partial x\partial^2 y}+r_y(\Delta y)^3\frac{\partial^3 c}{\partial y^3}\right], \tag{20}$$

suggesting that SUTRA is less accurate because of the mass-lumping.

2.3. Numerical Stability

[19] We also need to consider the likely range of grid-spacing and time stepping required to ensure SUTRA's numerical stability.

[20] The numerical stability range of a solution technique can sometimes be determined analytically; for complicated techniques the stability range can be more difficult to establish [*Hindmarsh et al.*, 1984; *Noye*, 1991]. *Voss* [1984] suggests a provisional stability constraint of $\Delta x \leq 4\beta$, where β is the longitudinal dispersivity. This implies that SUTRA is numerically unstable whenever the hydrodynamic dispersivity is zero. However, the code has been used successfully for simulations without hydrodynamic dispersivity, such as the Henry and Elder problems [*Voss*, 1984; *Voss and Souza*, 1987].

[21] We suggest that SUTRA's approximate stability range in one dimension is in fact

$$\frac{u\Delta x}{D+u\beta}<4, \tag{21}$$

as this becomes identical to Voss's criterion if $D \ll \beta$, but also permits purely diffusive simulations. This stability range is suggested simply by analogy with Voss' criterion, except that diffusion is taken into account. The denominator commonly arises as a measure of total dispersive forces in one-dimensional analyses of constant-velocity flow and transport, such as those of *Simmons* [1997]. The term on the left-hand side is the grid Peclet number, *Pe*. For one-dimensional simulations, the grid Peclet number is simply the ratio of Courant and diffusion numbers, i.e., $Pe = r/s$. In the absence of a formal calculation of SUTRA's stability range, we will employ equation (20) throughout this paper.

2.4. A Test Code

[22] To illustrate how these numerical issues affect simulation results, a test code was developed. The test code permits easy switching between a number of solution techniques for the concentration equation. It simulates variable-density groundwater flow and solute transport in saturated, isotropic and homogeneous aquifers. It always uses a uniform rectangular mesh, so both finite element and finite difference techniques can be directly compared. Its pressure solver is identical to SUTRA's as its MEPDE, equation (18), suggests that this is sufficiently accurate.

[23] The main solution options are (1) SUTRA-type, (2) SUTRA without mass-lumping, (3) SUTRA with numerical dispersion terms subtracted, (4) SUTRA without mass-lumping and with numerical dispersion terms subtracted, (v) SUTRA with an "inconsistent velocity" formulation, (5) a Crank-Nicolson finite difference method, and (6) a higher-order finite difference method. The third option is suggested by SUTRA's MEPDE. The numerical dispersion coefficients are calculated within the code and are then subtracted from the molecular diffusion coefficient to cancel out the numerical dispersion. That is, instead of the code calculating

$$D\left(\frac{\partial^2 c}{x^2}+\frac{\partial^2 c}{\partial y^2}\right), \tag{22}$$

it calculates

$$\left(D-D^N_{xx}\right)\frac{\partial^2 c}{\partial x^2}+\left(D-D^N_{yy}\right)\frac{\partial^2 c}{\partial y^2}-D^N_{xy}\frac{\partial^2 c}{\partial x\partial y}. \tag{23}$$

This is similar to the "correction factor" approach used by *Noye and von Trojan* [1998].

[24] The fifth option does not employ the "consistent velocity" formulation of *Voss* [1984]. Instead, all velocities are calculated according to equation (8).

[25] The two finite difference methods are given by J. A. Woods (Numerical accuracy of variable-density groundwater flow and solute transport simulations, Ph.D. thesis, submitted to University of Adelaide, Adelaide, South Australia, Australia, 2003). These were developed for the (linear) constant-coefficient advection-diffusion equation. To use them here, we linearize the PDE for concentration as follows: substituting Darcy's law (equation 3) into equation (2) yields

$$\frac{\partial c}{\partial t}+\left(\frac{1}{\varepsilon}\mathbf{q}-\frac{D}{\rho}\nabla\rho\right)\cdot\nabla c-D\nabla^2 c=0. \tag{24}$$

To linearize this at each node and time step, we compare the nonlinear equation with its linear counterpart, the advection-diffusion equation,

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}+V\frac{\partial f}{\partial y}-\alpha_x\frac{\partial^2 f}{\partial x^2}-\alpha_y\frac{\partial^2 f}{\partial y^2}=0, \tag{25}$$

where U and V can be considered to be locally linearized velocities, and α_x and α_y are locally linearized diffusion coefficients.

[26] Thus the linearized coefficients for the concentration equation are $U = u - (D/\rho)(\partial\rho/\partial x)$, $V = v - (D/\rho)(\partial\rho/\partial y)$, and $\alpha_x = \alpha_y = D$.

[27] These finite difference methods are included because neither incorporates any numerical dispersion when their coefficients are constant. Under these circumstances, the Crank-Nicolson method is second order and unconditionally stable, while the higher-order method is third order and numerically stable provided the Courant numbers remain less than 1. They are likely to be less accurate and numerically stable for other types of flow [*Noye*, 2000].

[28] In the following sections, we apply these techniques to two specific test problems, the Gauss pulse test for constant-density flow and the Elder problem for variable-density flow.

3. A Constant-Density Test Problem: The Gauss Pulse

[29] Our first test problem is the Gauss pulse test, which has been extensively used by Noye in the analysis of finite difference methods [*Leonard and Noye*, 1990; *Noye*, 1990,

Table 1. Input Parameters for the Gauss Pulse Test Problem

	Symbol	Value	Units
Grid Geometry			
Grid spacing	Δx	0.025	m
Grid spacing	Δy	0.025	m
Time step length	Δt	0.0125	d
Domain width	X	2	m
Domain height	Y	2	m
Simulation length	T	1.25	d
Parameters			
permeability	k	1×10^{-9}	m^2
porosity	ε	1	(—)
dynamic viscosity	μ	1	$kg\ m^{-1}\ d^{-1}$
storativity	S	0	Pa^{-1}
diffusivity	D	0.01	$m^2\ d^{-1}$
density change with concentration	γ	0	(—)
acceleration due to gravity	g	0	$m\ d^{-2}$
Initial Conditions			
$\mathbf{q}(\mathbf{x}, 0) = (0.8, 0.8)$			
$c(\mathbf{x}, 0) = \exp\left[-\frac{1}{D}\left\{(x-x_0)^2+(y-y_0)^2\right\}\right]$			
Boundary Conditions			
$\mathbf{q} = (0.8, 0.8)$			
$c(0, y, t) = \frac{1}{4t+1}\exp\left[-\frac{(ut+x_0)^2+(y-vt-y_0)^2}{D(4t+1)}\right]$			
$c(X, y, t) = \frac{1}{4t+1}\exp\left[-\frac{(X-ut-x_0)^2+(y-vt-y_0)^2}{D(4t+1)}\right]$			
$c(x, 0, t) = \frac{1}{4t+1}\exp\left[-\frac{(x-ut-x_0)^2+(vt+y_0)^2}{D(4t+1)}\right]$			
$c(x, Y, t) = \frac{1}{4t+1}\exp\left[-\frac{(x-ut-x_0)^2+(Y-vt-y_0)^2}{D(4t+1)}\right]$			
Convergence Criteria			
Maximum number of iterations per time step		1	
Maximum pressure difference between iterations		NA	
Maximum concentration difference between iterations		NA	

1991; *Noye and Hayman*, 1992]. It is a type of contaminant slug problem, similar to those studied by *Prickett et al.* [1981] and *Javandel et al.* [1984]. However, unlike these others, our version includes molecular diffusion but no hydrodynamic dispersion. This makes any numerical dispersion easier to see.

[30] In groundwater terms, the Gauss pulse test simulates the release of a narrow slug of solute. The solute is released into a horizontal, homogeneous, and isotropic aquifer. Flow within the aquifer is constant, moving from the southwest to the northeast. Thus the solute slug moves with the flow, spreading out as it does so because of molecular diffusion. The mathematical specification of the problem is given in Table 1. The parameters shown are for illustrative purposes only and does not correspond to realistic groundwater flow.

[31] The Gauss pulse test has an exact solution [*Noye and Hayman*, 1992]:

$$c(x,y,t) = \frac{1}{4t+1}\exp\left[-\frac{(x-ut-x_0)^2+(y-vt-y_0)^2}{D(4t+1)}\right], \quad (26)$$

where (x_0, y_0) is the initial location of the center of the solute slug. Thus, for our parameters, the center of the pulse should shift to (1.5, 1.5) and have a concentration 1/6 of its original value by the end of the simulation at $t = T = 1.25$ days.

3.1. Numerical Analysis

[32] The results from Section 2 can be applied here. Firstly, the amount of numerical dispersion for this problem and parameter set can be estimated for SUTRA. The numerical dispersion coefficients are calculated to be $D_{xx}^N = D_{yy}^N = \frac{1}{2}u^2\Delta t = 4 \times 10^{-3} m^2 d^{-1}$ and $D_{xy}^N = uv\Delta t = 8 \times 10^{-3} m^2 d^{-1}$. Thus the numerical dispersion is of a similar magnitude to the molecular diffusion coefficient $D = 0.01$, suggesting that numerical dispersion will be significant.

[33] We also check the numerical stability of the problem. The Courant numbers are $r_x = r_y = 0.4$ and the diffusion numbers are $s_x = s_y = 0.2$. Thus the grid Peclet number $Pe = 2 < 4$, which suggests that SUTRA will be numerically stable for this problem.

3.2. Results

[34] Concentration contours for the final time step are given in Figure 1. Only the northeast quarter of the domain is shown. The exact solution, Figure 1a, is perfectly circular, with a maximum concentration of 1/6. By contrast, the results of SUTRA, shown in Figure 1b, are elongated in the direction of flow. The peak concentration is a much lower

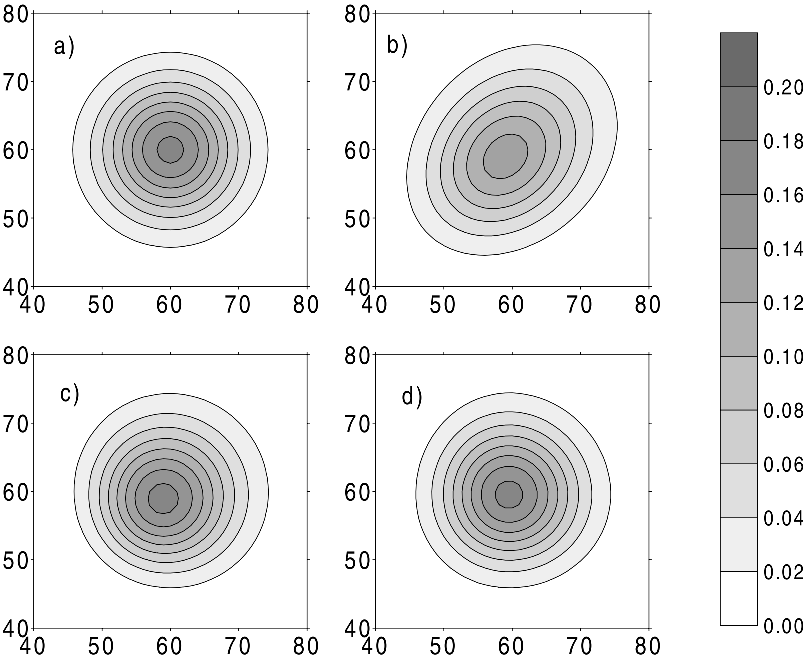

Figure 1. Comparison of concentration plots for the final time step of the Gauss pulse test as calculated (a) exactly, (b) by SUTRA, (c) by SUTRA without numerical dispersion, and (d) by the higher-order method. Concentration is in units of kg/kg. Distances are in terms of nodes.

0.1308. To test whether this inaccuracy is indeed caused by numerical dispersion, as suggested by our numerical analysis, the problem is run again but with the numerical dispersion removed, as described in section 2.4. The results for this run are shown in Figure 1c. The contours are much more circular and the maximum concentration is 0.1687, which is much closer to the exact value. The contours are slightly off-center, possibly because of residual third-order errors. More accurate still, however, is the higher-order method, as shown in Figure 1d. Its maximum concentration is 0.1662. Figure 2 shows how the maximum concentration calculated by SUTRA compares with the exact solution. This suggests that at a small number of time steps, the difference between the exact and the SUTRA solution is significant. As the number of time steps increases, the difference is smaller, but it does remain at approximately 20% for all time steps.

[35] As a check and comparison, the Gauss problem was run with different velocities. The exact solution states that the maximum concentration depends only on the time elapsed, and not on the velocity. However, the numerical dispersion terms found in the MEPDE do depend on velocity. Thus SUTRA should match the exact solution if the velocity is set to zero. A numerical experiment confirmed this. With $\mathbf{q} = \mathbf{0}$, SUTRA produced circular contours and a maximum concentration of 0.1667 at the final time step; i.e., SUTRA is accurate when there is no fluid velocity present.

[36] Thus our numerical analysis and model results are clearly in agreement. This shows that SUTRA induces potentially significant numerical dispersion into its solute-transport simulations. Despite almost two decades of SUTRA's use and popularity, this is seldom commented on explicitly in the literature, possibly because numerical dispersion alters flow patterns in a manner resembling hydrodynamic dispersion (an exception is the work by *Watson and Barry* [2001], who report problems with SUTRA's numerical dispersion in a cross-verification study). Modelers expect dispersive-type flows and so this type of numerical error distorts results in a plausible way. The Gauss pulse test makes numerical dispersion visible because of its simplicity and circular contours. It satisfies all the requirements of a good benchmark: it is well-defined, well-understood and difficult to simulate accurately because of its sensitivity to numerical dispersion.

[37] We have demonstrated that our analysis techniques can successfully suggest ways of improving a code. Sub-

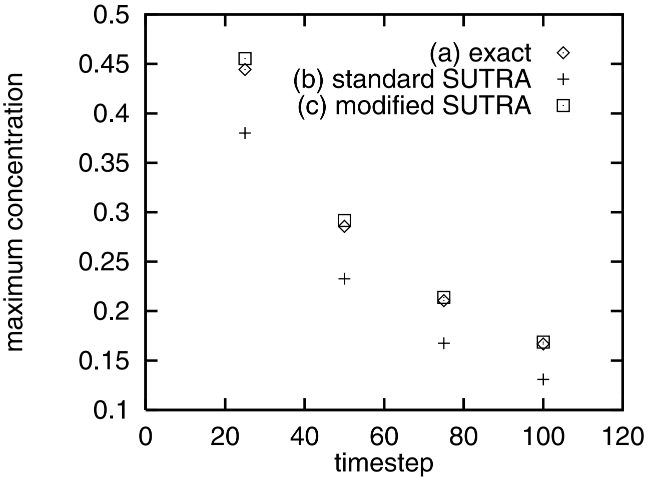

Figure 2. Maximum concentration for the Gauss pulse problem, as calculated exactly (diamonds), by SUTRA (pluses), or by the modified SUTRA method in which numerical dispersion is subtracted (squares).

Table 2. Input Parameters for the Elder Problem

	Symbol	Value	Units
Grid Geometry			
Grid spacing	Δx	14.28	m
Grid spacing	Δy	6	m
Time step length	Δt	2.628×10^6	s
Domain width	X	600	m
Domain height	Y	150	m
Simulation length	T	20	years
Parameters			
Permeability	k	4.845×10^{-13}	m^2
Porosity	ε	0.1	(—)
Dynamic viscosity	μ	10^{-3}	kg m^{-1} s^{-1}
Storativity	S	0	Pa^{-1}
Diffusivity	D	3.565×10^{-6}	m^2 s^{-1}
Density change with concentration	γ	0.2	(—)
acceleration due to gravity	g	−9.81	m s^{-2}
Initial Conditions			
Hydrostatic pressure			
$c(x, y, 0) = 0$			
Boundary Conditions			
$v(0, y, t) = 0$			
$v(600, y, t) = 0$			
$u(x, 0, t) = 0$			
$u(x, 150, t) = 0, x \neq 0, x \neq 600$			
$p(0, 150, t) = 0$			
$p(600, 150, t) = 0$			
$\partial c/\partial x(0, y, t) = 0$			
$\partial c/\partial x(600, y, t) = 0$			
$\partial c/\partial y(x, 0, t) = 0$			
$\partial c/\partial y(x, 150, t) = 0, 0 \leq x \leq 150, 450 \leq x \leq 600$			
$c(x, 150, t) = 1, 150 \leq x \leq 450$			
Convergence Criteria			
Maximum number of iterations per time step		20	
Maximum pressure difference between iterations		500	kg $m^{-1}s^{-2}$
Maximum concentration difference between iterations		0.01	(—)

tracting numerical dispersion leads to substantial improvement in code accuracy for this type of simple, linear problem.

4. A Variable-Density Test Case: The Elder Problem

[38] In this section we no longer assume that fluid density is constant. Instead, we let fluid density vary linearly with solute concentration and solve nonlinear PDEs (1) and (2). This type of solute transport is very common. Examples include seawater intrusion [*Bear et al.*, 1999], leachate plumes [*Koch and Zhang*, 1992] and saline disposal basins [*Narayan and Armstrong*, 1995].

[39] Unfortunately, the equations required to simulate this type of flow are nonlinear. This has three main effects. Firstly, there are only a few known problems of this type for which there are any analytic results, such as those of *van Duijn and Schotting* [1997], *Nield and Bejan* [1992], and *Bear et al.* [1999]. Most of these are either unsuitable as benchmark problems because of their unrealistic (e.g., infinite) boundary conditions or are yet to be considered as potential test problems. This means that benchmark tests need to be derived from laboratory experiments or through comparison with other numerical codes, introducing doubts about the accuracy of the results against which a code is being compared. Secondly, this makes accurate numerical simulation much more prone to error. Finally, analysis techniques used to gauge numerical error are often no longer strictly valid. All of these factors combine to make simulation of variable-density flow much more difficult than constant-density flow. Nevertheless, we wish to show in this section that useful information about nonlinear PDEs can be found through a combination of numerical analysis, grid resolution studies and test problems.

[40] The test problem in this case is Elder's "short-heater" problem, a heat flow problem [*Elder*, 1967] which was converted for use with solute-transport codes by *Voss and Souza* [1987]. It is one of the standard benchmark problems for variable-density flow and transport. The problem is described in Table 2 and Figure 3.

[41] There has been much discussion in the literature about the results of the Elder problem. Elder's original numerical solution shows complicated fingering which develops into convection cells separated by a downwelling of solute and SUTRA matches this reasonably well [*Voss and Souza*, 1987]. However, *Oldenburg and Pruess* [1995] note that the Elder's numerical results seem to contradict the published photographs of his laboratory work [*Elder*, 1967]. These show no indication of any central downwelling. Moreover, they perform a grid resolution study. When using the same grid as Voss and Souza's, the results from the two

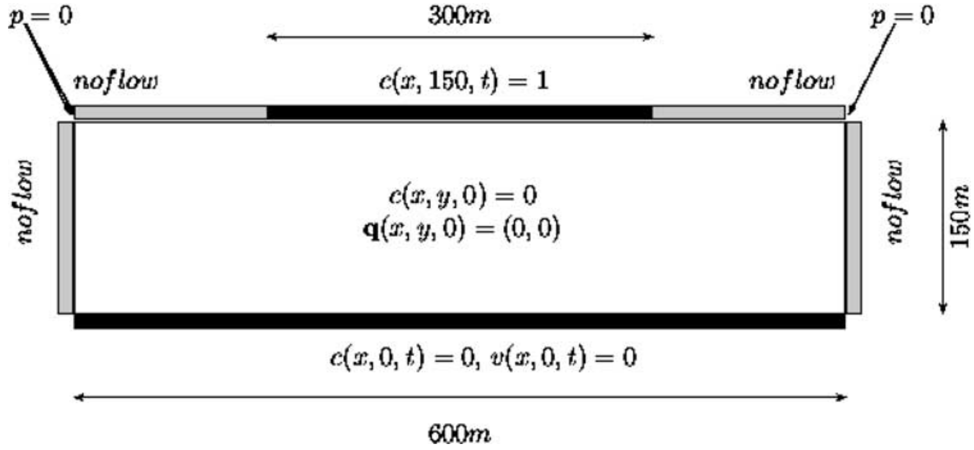

Figure 3. The initial and boundary conditions of the Elder problem.

numerical codes match well. Using a finer grid produces results more like Elder's laboratory photographs, in that there is an upwelling of fluid in the center of the flow and hence no central plume. *Oldenburg and Pruess* [1995] conjecture that their finer grid simulation is more accurate than either Elder's semianalytic results or Voss and Souza's numerical results.

[42] These results have since been confirmed by others, e.g., *Kolditz et al.* [1997], *Holzbecher* [1998], *Ackerer et al.* [1999], *Prasad* [2000], and *Oltean and Buès* [2001]. However, *Frolkovič and Schepper* [2000] consider even finer grids. They show that for even higher grid resolutions a small downwelling of solute occurs in the center. An unpublished study (H.-J. Diersch, The Elder problem, unpublished results, 2000) using another code shows similar results. This does seem to contradict Elder's laboratory photographs, but it is worth noting that the photographs show a quite asymmetric flow, despite the symmetry of the problem. This is presumed to be due to experimental variability and demonstrates the difficulty in using laboratory results as a benchmark. Thus the extent to which either the laboratory work or Elder's numerical solution can be trusted remains an open question.

[43] A very fine grid is also used in another recent paper [*Boufadel et al.*, 1999]. Unlike *Frolkovič and Schepper* [2000], Boufadel et al.'s results do not show a central downwelling, possibly because their numerical technique seems to be fairly inaccurate. At lower grid resolutions their code produces distinctly asymmetric results.

[44] Thus results to date show that coarse grids and very fine grids generally yield a downwelling, while finer grids yield an upwelling. It is possible that yet finer grids might yield other results, although eventually the grid size would become so small that roundoff error would dominate [*May and Noye*, 1984].

[45] What is clear is that simulation results for the Elder problem change with grid size, time stepping scheme and numerical solution method. Thus one can presume that the problem is sensitive to numerical error. This is supported by *Frolkovič and Schepper* [2000] who convincingly demonstrate that even a small change in the problem's initial conditions may lead to qualitative differences in long-term behavior. However, sensitivity to to numerical error is a useful quality in a benchmark problem only if the problem has a verified and precise solution. The Elder problem lacks this at present, which limits its usefulness as a benchmark. This may change if the very fine mesh results of *Frolkovič and Schepper* [2000] are replicated.

[46] In fact, there is a good reason why the simulation results should be expected to vary with grid size, as the grid size alters both initial and boundary conditions. Say that a numerical scheme assumes that concentration varies vertically across an element according to function $g(y, t)$ (typically a linear or quadratic function). The amount of solute initially present in an element under the upper constant-concentration boundary is

$$\Delta x \int_{150-\Delta y}^{150} g(y,0)\, dy. \tag{27}$$

If, for example, concentration varies linearly across an element, then $g(y, 0) = 1 + (y - 150)/\Delta y$. Thus the amount of solute in the element is

$$\Delta x \int_{150-\Delta y}^{150} \left(1 + \frac{y-150}{\Delta y}\right) dy = \frac{1}{2}\Delta x \Delta y. \tag{28}$$

That is, the initial amount of solute present in the model domain depends on the grid size. Also, whenever a discontinuity exists in the boundary conditions, the implementation of the boundary condition at the point of discontinuity will be grid dependent. *Konikow et al.* [1997] have noted similar discrepancies in the specification of initial conditions for the "salt-dome" problem.

[47] The discontinuous pressure boundary at the top also presents a problem. Depending on the grid size and the interpolation method used, a code's interpolation between the constant-pressure and the neighboring no-flow node will differ. Thus every time a modeler varies the grid spacing in a grid resolution study, the physical problem is altered, leading to variations in the simulated solution. These variations may occur anywhere in the solution regime.

4.1. Numerical Analysis

[48] To examine SUTRA's numerical stability and dispersion, we need to know what velocities occur in the Elder simulation. From a preliminary model run, the maximum

magnitude of the velocity is approximately $3 \times 10^{-6} ms^{-1}$ while a more typical velocity is $5 \times 10^{-7} ms^{-1}$.

[49] Thus for a month-long time step ($2.628 \times 10^6 s$), as used by *Voss and Souza* [1987], the maximum numerical dispersion coefficients will be $D^N_{xy} = \pm 2.37 \times 10^{-5} m^2 s^{-1}$ and $D^N_{xx} = D^N_{yy} = 1.18 \times 10^{-5} m^2 s^{-1}$ which are an order of magnitude larger than the molecular diffusion coefficient, as $D = 3.565 \times 10^{-6} m^2 s^{-1}$. At more typical velocities, $D^N_{xy} = \pm 6.570 \times 10^{-7} m^2 s^{-1}$ and $D^N_{xx} = D^N_{yy} = 3.285 \times 10^{-7} m^2 s^{-1}$, which is smaller than the diffusion coefficient. This suggests that numerical dispersion is important wherever the flow velocity is particularly high.

[50] SUTRA's numerical stability depends on the grid and time stepping schemes used. Here we follow *Frolkovič and Schepper* [2000], dividing the problem domain into 2^{2L+1} elements for the half-domain, where L denotes the "grid level." We consider $L = 4, 5, 6$. The time step length is either 1 month or 0.1 months.

[51] Based on the maximum velocity, $Pe = u\Delta x/D = 0.842\Delta x$, so to keep $Pe < 4$ as required for stability, $\Delta x < 4.75\ m$ (e.g., a grid level of 5 or higher is required). On the basis of a more typical velocity of 5×10^{-7}, $\Delta x < 28.52\ m$, i.e., for all but the coarsest grids (grid levels 0–2).

[52] However, this simple stability analysis must be modified by what we know about numerical dispersion, which effectively changes the molecular diffusion coefficient and thus the grid Peclet number. For a worst-case (maximum velocity, larger time step) scenario, the effective diffusion coefficient $D^E_{xx} = 3.565 \times 10^{-6} + 1.18 \times 10^{-5} = 1.54 \times 10^{-5} m^2 s^{-1}$. This makes the stability criterion less severe, so that $\Delta x < 20.53\ m$ and not $\Delta x < 4.75\ m$, so the numerical dispersion distorts the flow but promotes numerical stability. It suggests that grid level 4 will be numerically stable for SUTRA after all. In addition, for grid levels >4 the effective diffusion number is greater than the Courant number, so the simulation becomes diffusion-dominated and not advection-dominated.

[53] For a more typical velocity of $5 \times 10^{-7} ms^{-1}$ the effective coefficients are less severe, i.e., $D^E_{xx} = 3.894 \times 10^{-6} m^2 s^{-1}$ ($\Delta x < 31.15\ m$), so there is comparatively little change to SUTRA's stability criterion.

[54] Thus for Elder's problem run with time steps 1 month long, SUTRA's numerical dispersion is only important at points of high velocity, where it increases the effective diffusion coefficient but permits a larger stability range and thus coarser grids.

4.2. Results

[55] The numerical stability of the simulations is much as expected. SUTRA is indeed numerically stable for all the grid levels considered here. The Crank-Nicolson method is almost always stable, but does not converge for $\Delta t = 0.1$ month and grid level 6. The higher-order method, being less numerically stable than SUTRA, only converges when both grid steps and time steps are small ($\Delta t = 0.1$ month and grid levels 6 and 7). The "inconsistent velocity" method always leads to nonconvergence, vindicating the decision of *Voss* [1984] to use the "consistent" version.

[56] Regrettably, SUTRA with numerical dispersion subtracted proves to be numerically unstable, failing to converge for $\Delta t = 1$ month. For smaller time steps, its results are theoretically very close to the standard SUTRA method and our simulation results agree with this. Thus, while this method proved very successful for the Gauss pulse problem, it is much less satisfactory for complex, nonlinear problems.

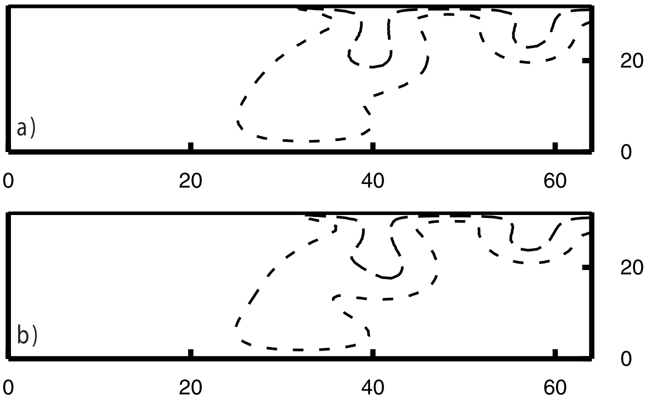

Figure 4. The 0.2 and 0.6 concentration contours for the Elder problem after 4 years of simulated time at grid level 5 using SUTRA with time steps of (a) 1 month or (b) 0.1 month. Axes are labeled in terms of nodes.

[57] Instead, the extent of numerical dispersion can be gauged through varying the time step length, as the numerical dispersion coefficients vary with Δt. A comparison is given in Figure 4. The main difference is that the outer plume (representing the 0.2 concentration contour) is shaped differently, which suggests that the greatest velocities occur in this area. However, numerical dispersion has minimal impact throughout most of the simulation, where the velocities are reasonably low. The SUTRA method without mass-lumping and the Crank-Nicolson method show similar behavior.

[58] Importantly, none of our methods replicate the results of *Frolkovič and Schepper* [2000] and H. J. Diersch (unpublished results, 2000), in their switch from a central downwelling (grid level 4) to an upwelling (level 5) and thence to a downwelling again (level 6). Figure 5 summarizes some of our results. At level 4, only SUTRA shows a downwelling at the center (right-hand side of Figure 5a). Without mass lumping, SUTRA produces a central upwelling (Figure 5d), as does the Crank-Nicolson method (Figure 5g). At level 5, SUTRA with and without mass lumping show an upwelling (Figures 5b and 5e), although the plumes are further apart without mass-lumping. However, the Crank-Nicolson method still shows a downwelling (Figure 5h). The direction of flow does not change as we move to grid level 6 (Figures 5c, 5f, and 5i); indeed SUTRA without mass lumping may have achieved grid convergence. Only the higher-order method produces results similar to those of H. J. Diersch (unpublished results, 2000) (Figure 5j). There still does not appear to be a unique solution to the Elder problem, even at higher grid levels and smaller time step sizes. As *Frolkovič and Schepper* [2000] suggest, the Elder problem may be truly indeterminate.

[59] If this is true, then what does this imply about the use of the Elder problem as a benchmark? While it is well-defined and difficult to simulate, it does not have well-understood and consistent results. It is undeniably a much more stringent test than the comparable Henry problem

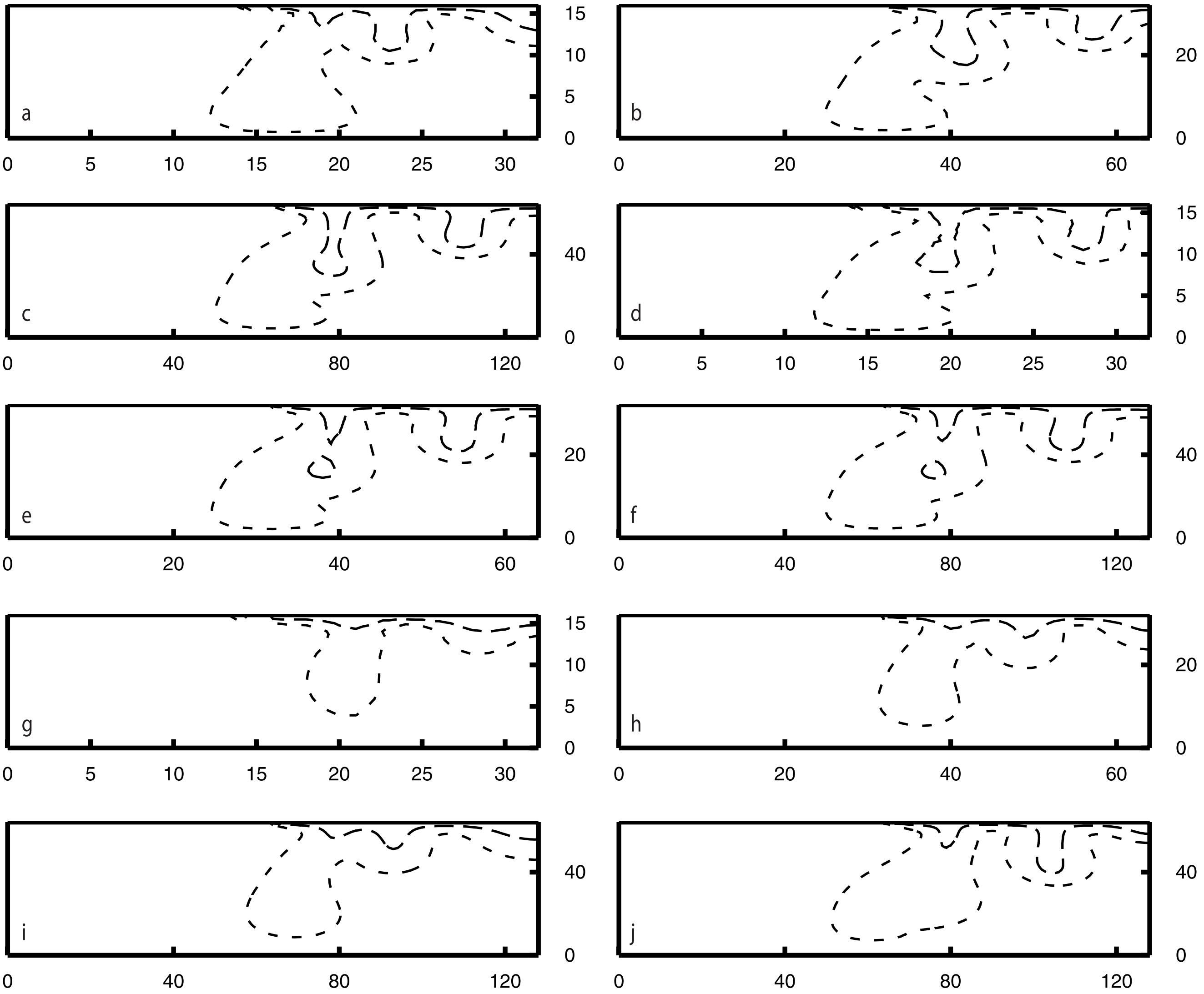

Figure 5. The 0.2 and 0.6 concentration contours for the Elder problem after 4 years of simulated time with $\Delta t = 0.1$ month for (a–c) SUTRA at grid levels 4, 5, 6, (d–f) SUTRA without mass-lumping at grid levels 4, 5, 6, (g–i) Crank-Nicolson method at grid levels 4, 5, 6, and (j) the higher-order method at grid level 6. Axes are labeled in terms of nodes. Because of the symmetry of the results, only the left half of the domain is shown.

[*Henry*, 1964; *Croucher and O'Sullivan*, 1995; *Benson et al.*, 1998; *Voss and Souza*, 1987] and so remains valuable, but if we expect a benchmark to distinguish between good and better methods of simulation, then the Elder problem is of limited usefulness. Codes capable of simulating the Elder problem do not necessarily agree on the results of other problems [*Simmons et al.*, 1999; *Woods et al.*, 1999], so it is clear that there is a need for even more rigorous test problems.

[60] Relating our results back to the numerical analysis, we note several important points. Firstly, compare the qualitatively different results of Figures 5a and 5d. They differ in the number of plumes and in the plumes' shape and extent. Yet the only difference in the solution methods is that Figure 5a has used mass-lumping and 5d has not. According to our linear analysis of their MEPDEs, these techniques differ only in their third order error terms. Third order error terms are usually dismissed as too small to have an effect on groundwater flow simulations, yet for a complicated and nonlinear problem such as Elder's, they evidently can have a substantial effect.

[61] Second-order errors (i.e., numerical dispersion) usually increase with time step length, so if second-order errors are more important than third-order ones, then we would expect different solution methods to yield similar results provided that the time step length is kept small. This does not occur with the Elder problem. The SUTRA and Crank-Nicolson methods differ in their second-order errors, but their results do not resemble each other's even for small time steps. We suggest that this is due to the sensitivity of the problem to plume-triggering processes and the nonlinear nature of the problem. Even small perturbations at the upper boundary may trigger a plume or a local vortex.

[62] Thus it seems that for nonlinear problems, even seemingly negligible errors can alter the shape of a flow.

This has important implications, especially when model results are part of an environmental management plan.

5. Conclusions

[63] Numerical analysis, grid resolution studies and test problems can be fruitfully combined in the investigation of numerical error in simulation codes. A technique adapted from FDMs (the MEPDE) can suggest ways of improving existing codes, such as SUTRA. In this paper we have shown that SUTRA can be improved for simple linear problems and regular grids when numerical dispersion is subtracted. Deriving an appropriate expression for numerical dispersion for irregular grids is more difficult and it is likely that there are more efficient ways of addressing this computationally. For more complex simulations, removing mass lumping may be advantageous for little computational cost. Similarly, calculation of numerical stability criteria and their interaction with numerical dispersion may improve prediction of suitable grid and time stepping regimes. While our analysis is of SUTRA, similar insights may be achieved through the analysis of other codes.

[64] Some of this information may be usefully incorporated into a simulation code, warning the user of probable numerical instability (as some codes already do) or specifying the amount of numerical dispersion present—although one hopes that through the use of the MEPDE, code developers might avoid numerically dispersive methods altogether.

[65] However, some prefer numerically dispersive methods because of their frequently enhanced numerical stability (as demonstrated in this paper). For example, *Frolkovič and Schepper* [2000] argue that their exponential upwind scheme is superior partly because their calculated concentration never strays out of bounds (i.e., $0 \leq c \leq 1$ everywhere). While this makes physical sense, *Zoppou and Roberts* [1996] have elegantly demonstrated that some higher-order methods must be expected to go out of bounds to a small degree at sharp interfaces but otherwise show a much better match to exact solutions. Upwind schemes, almost by definition, artificially increase the dispersion in a model: that is, they alter the physical problem being simulated.

[66] This behavior may be critical in complex, nonlinear simulations. We have demonstrated that even third order error terms can substantially affect simulation results.

[67] As well as numerical analysis, new benchmark problems, such as the Gauss pulse test described here, allow a modeler to better gauge the accuracy of a code. Nevertheless, much still needs to be done on the verification of variable-density flow simulators. This is likely to require the development of further mathematical tools and the selection of better benchmark tests. This paper has outlined a few starting possibilities.

[68] **Acknowledgments.** Juliette Woods was supported by an Australian Postgraduate Award and through the CRC for Catchment Hydrology.

References

Ackerer, P., A. Younes, and R. Mosé, Modeling variable density flow and solute transport in porous medium: 1. Numerical model and verification, *Transp. Porous Media*, *35*, 345–373, 1999.

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

Bear, J., A. H.-D. Cheng, S. Sorek, D. Ouazar, and I. Herrera, *Seawater Intrusion in Coastal Aquifers: Concepts, Methods, and Practices*, *Theory Appl. Transp. Porous Media*, vol. 15, Kluwer Acad. Press, Norwell, Mass., 1999.

Becker, E. B., G. F. Carey, and J. Tinsley, *Finite Elements*, vol. 6, *Fluid Mechanics*, Prentice-Hall, Old Tappan, N. J., 1986.

Benson, D. A., A. E. Carey, and S. W. Wheatcraft, Numerical advective flux in highly variable velocity fields exemplified by saltwater intrusion, *J. Contam. Hydrol.*, *34*, 207–233, 1998.

Boufadel, M., M. Suidan, and A. Venosa, Numerical modeling of water flow below dry salt lakes: Effect of capillarity and viscosity, *J. Hydrol.*, *221*, 55–74, 1999.

Croucher, A., and M. O'Sullivan, The Henry problem for saltwater intrusion, *Water Resour. Res.*, *31*, 1809–1814, 1995.

Diersch, H.-J., Interactive, graphics-based finite-element simulation system FEFLOW for modeling groundwater flow, contaminant mass and heat transport processes, FEFLOW user's manual version 4.5., WASY Ltd., Berlin, 1996.

Elder, J., Transient convection in a porous medium, *J. Fluid Mech.*, *27*, 609–623, 1967.

Ferziger, J. H., and M. Perić, *Computational Methods for Fluid Dynamics*, Springer-Verlag, New York, 1999.

Frolkovič, P., and H. D. Schepper, Numerical modelling of convection dominated transport coupled with density driven flow in porous media, *Adv. Water Resour.*, *24*, 63–72, 2000.

Gresho, P. M., and R. Sani, *Incompressible Flow and the Finite Element Method: Advection-Diffusion and Isothermal Laminar Flow*, John Wiley, 1998.

Henry, H., Effects of dispersion on salt encroachment in coastal aquifers, *U. S. Geol. Surv. Water Supply Pap.*, *1613-C*, 1964.

Hindmarsh, A., P. Gresho, and D. Griffiths, The stability of explicit Euler time-integration for certain finite difference approximations of the multi-dimensional advection-diffusion equation, *Int. J. Numer. Methods Fluids*, *4*, 853–897, 1984.

Holzbecher, E., *Modeling Density-Driven Flow in Porous Media: Principles, Numerics, Software*, Springer-Verlag, New York, 1998.

Javandel, I., C. Doughty, and C. Tsang, *Groundwater Transport: Handbook of Mathematical Models*, *Water Resour. Monogr.*, vol. 10, AGU, Washington, D. C., 1984.

Koch, M., and G. Zhang, Numerical simulation of the effects of variable density in a contaminant plume, *Ground Water*, *30*, 731–742, 1992.

Kolditz, O., R. Ratke, H.-J. Diersch, and W. Zielke, Coupled groundwater flow and transport: 1. Verification of variable density flow and transport models, *Adv. Water Resour.*, *21*, 27–46, 1997.

Konikow, L., W. Sanford, and P. Campbell, Constant-concentration boundary conditions: Lessons from the HYDROCOIN variable-density groundwater benchmark problem, *Water Resour. Res.*, *33*, 2253–2261, 1997.

Lal, A. W., Numerical errors in groundwater and overland flow models, *Water Resour. Res.*, *36*, 1237–1247, 2000.

Leonard, B., and B. J. Noye, Second and third order two-level implicit FDMs for unsteady one-dimensional convection-diffusion equation, in *Computational Techniques and Applications: CTAC-89*, edited by H. Hogarth and B. J. Noye, pp. 311–317, Taylor and Francis, Philadelphia, Pa., 1990.

May, R., and B. J. Noye, The numerical solution of ordinary differential equations: Initial value problems, in *Computational Techniques for Differential Equations*, edited by B. J. Noye, pp. 1–94, North-Holland, New York, 1984.

Narayan, K. A., and D. Armstrong, Simulation of groundwater interception at Lake Ranfurly, Victoria, incorporating variable density flow and solute transport, *J. Hydrol.*, *165*, 161–184, 1995.

Nield, D. A., and A. Bejan, *Convection in Porous Media*, Springer-Verlag, New York, 1992.

Noye, B. J., Finite difference techniques, in *Numerical Simulation of Fluid Motion*, edited by B. J. Noye, pp. 1–112, North-Holland, New York, 1978.

Noye, B. J., A new third-order finite-difference method for transient one-dimensional advection-diffusion, *Comput. Appl. Numer. Methods*, *6*, 279–288, 1990.

Noye, B. J., A compact unconditionally-stable finite-difference method for transient one-dimensional advection-diffusion, *Comm. App. Numer. Meth. Eng.*, *7*, 501–512, 1991.

Noye, B., Explicit finite difference methods for variable velocity advection in the presence of a source, *Comput. Fluids*, *29*, 385–399, 2000.

Noye, B. J., and K. Hayman, Accurate finite difference methods for solving the advection-diffusion equation, in *Computational Techniques and Ap-*

plications: CTAC-85, edited by B. J. Noye and R. May, pp. 137–157, North-Holland, New York, 1985.

Noye, B. J., and K. Hayman, Explicit two-level finite-difference methods for the two-dimensional diffusion equation, *Int. J. Comput. Math.*, *42*, 223–236, 1992.

Noye, B. J., and A. von Trojan, An explicit finite-difference method for variable velocity advection, in *Computational Techniques and Applications: CTAC-97*, edited by B. J. Noye, M. D. Teubner, and A. W. Gill, pp. 513–520, World Sci., River Edge, N. J., 1998.

Oldenburg, C. M., and K. Pruess, Dispersive transport dynamics in a strongly coupled groundwater-brine flow system, *Water Resour. Res.*, *31*, 289–302, 1995.

Oltean, C., and M. A. Buès, Coupled groundwater flow and transport in porous media: A conservative or non-conservative form?, *Transp. Porous Media*, *44*, 219–246, 2001.

Prasad, A., Density-driven groundwater flow and solute transport in heterogeneous porous media, M.S. thesis, Flinders Univ. of South Australia, Adelaide, 2000.

Prickett, T. A., T. G. Naymik, and C. G. Lonnquist, A "random-walk" solute transport model for selected groundwater quality evaluation, technical report, Ill. Dep. of Energy and Nat. Resour., Champaign, 1981.

Schincariol, R. A., F. W. Schwartz, and C. A. Mendoza, On the generation of instabilities in variable density flow, *Water Resour. Res.*, *30*, 913–927, 1994.

Simmons, C. T., Density-induced groundwater flow and solute transport beneath saline disposal basins, Ph.D. thesis, Flinders Univ. of South Australia, Adelaide, 1997.

Simmons, C. T., K. A. Narayan, and R. A. Wooding, On a test case for density-dependent groundwater flow and solute transport models: The salt lake problem, *Water Resour. Res.*, *35*, 3607–3620, 1999.

van Duijn, C., and R. Schotting, Brine transport in porous media: On the use of Von Mises and similarity transformations, *Modell. Anal. Simulation Rep. MAS–R9724*, Cent. voor Wiskundeen Inf., Amsterdam, 1997.

Voss, C. I., SUTRA: A finite-element simulation model for saturated-unsaturated fluid-density-dependent ground-water flow with energy transport or chemically-reactive single-species solute transport, *U.S. Geol. Surv. Water Resour. Invest. Rep.*, *84-4369*, 1984.

Voss, C. I., and W. R. Souza, Variable density flow and solute transport simulation of regional aquifers containing a narrow freshwater-saltwater transition zone, *Water Resour. Res.*, *23*, 1851–1866, 1987.

Warming, R., and B. Hyett, The modified equation approach to the stability and accuracy analysis of finite difference methods, *J. Comput. Phys.*, *14*, 159–179, 1974.

Watson, S. J., and D. A. Barry, Numerical analysis of stable brine displacements for evaluation of density-dependent flow theory, *Phys Chem. Earth, Part B, Hydrol. Oceans Atmos.*, *26*, 325–331, 2001.

Woods, J., C. T. Simmons, and K. A. Narayan, Verification of black box groundwater models, in *Proceedings of the Third Biennial Engineering Mathematics and Applications Conference: EMAC98*, edited by E. Tuck and J. Stott, pp. 523–526, Inst. of Eng., Barton, Australia, 1998.

Woods, J., M. D. Teubner, C. T. Simmons, and K. A. Narayan, Numerical inaccuracy in groundwater modelling: diagnosis and pathology, in *Water 99 Joint Congress: 25th Hydrology and Water Resources Symposium, 2nd International Conference on Water Resources and Environment Research: Handbook and Proceedings*, pp. 532–538, Inst. of Eng. Aust., Brisbane, 1999.

Zoppou, C., and S. Roberts, Behaviour of finite difference schemes for advection diffusion equations, *Tech. Rep. Math. Res. Rep. MRR 062-96*, Aust. Natl. Univ., Canberra, 1996.

K. A. Narayan, Davies Laboratory, CSIRO Land and Water, PMB PO Aitkenvale, Townsville, Queensland 4814, Australia.

C. T. Simmons, School of Chemistry, Physics and Earth Sciences, Flinders University of South Australia and Centre for Groundwater Studies, Adelaide, South Australia 5042, Australia.

M. D. Teubner, Department of Applied Mathematics, Adelaide University, Adelaide, South Australia 5005, Australia.

J. A. Woods, Institute for Computational Engineering and Sciences, ACES 6.444, University of Texas at Austin, Austin, TX 78712, USA. (jwoods@cfdlab.ae.utexas.edu)