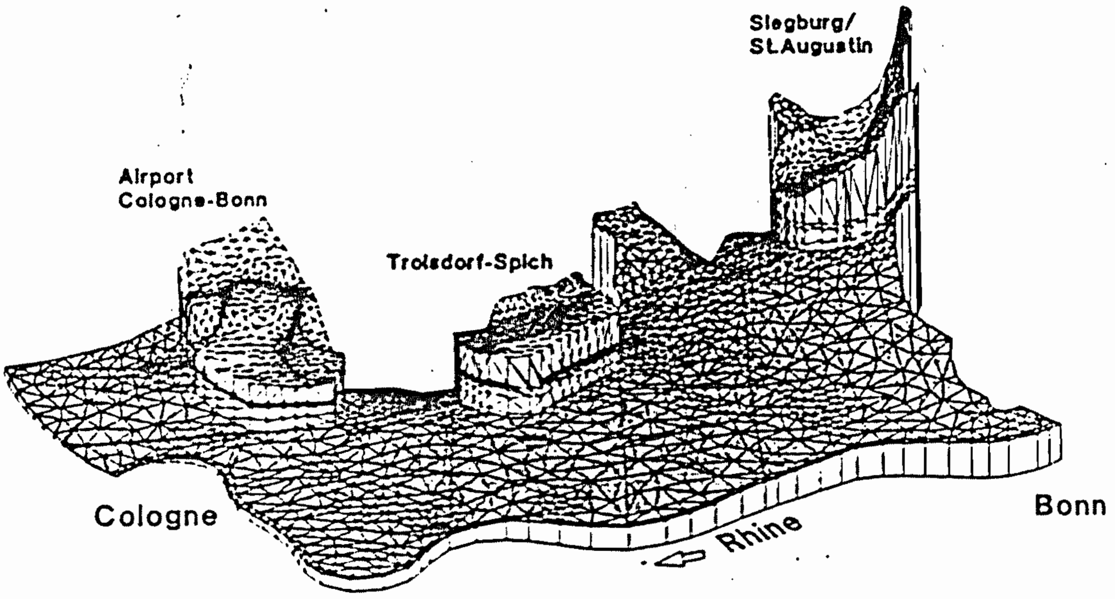
COMPUTATIONAL METHODS IN WATER RESOURCES X

Volume 1

[illegible]

[illegible]



SOLUTION OF FLOW AND TRANSPORT PROBLEMS USING A COLLOCATION DISCRETIZATION OF A DOMAIN DECOMPOSITION ALGORITHM

Joseph Guarnaccia*, Ismael Herrera** and George Pinder*
*University of Vermont, 109 Votey Bldg., Burlington, VT 05405
**Instituto de Geofisica, UNAM, Apdo Postal 22-582, 14000 Mexico, D. F.

INTRODUCTION

We consider herein the two-dimensional simulation of fluid flow and contaminant transport in porous media using the collocation finite element method in conjunction with the domain decomposition conjugate gradient (DD-CG) algorithm proposed by Herrera et al. (1994). The motivation for using domain decomposition is that it allows us to take advantage of parallel processing and to treat subdomains differently from a computational point of view.

As detailed in Herrera et al. (1994), the subdomains arising from the domain decomposition are linked together using a conjugate gradient iterative algorithm to enforce continuity of the normal flux across the interfaces separating subdomains. An important issue to be discussed below is the procedure used to transpose the theory, derived for the continuous variable case, into an algorithm applicable to the discrete system arising from the application of the collocation finite element method.

In this paper we will discuss the collocation discretization and the use of the Hermite cubic basis to represent the dependent variables. We then detail the procedure used to calculate the variables on the interfaces required for the conjugate gradient algorithm. Finally, we present some convergence results for a model flow and transport problem.

MODEL PROBLEM

Consider for this discussion the following flow and transport model. The first equation to be solved is the incompressible groundwater flow equation, written as:

$$Lh(\mathbf{x}) = -\nabla\cdot\left[\underset{\approx}{k}(\mathbf{x})\cdot\nabla h(\mathbf{x})\right] = Q(\mathbf{x}), \quad [\mathbf{x}] \in \Omega \tag{1}$$

where $\underset{\approx}{k}(\mathbf{x})$ is the hydraulic conductivity tensor, which for this discussion is assumed to be diagonal, $h(\mathbf{x})$ is the fluid pressure head, and $Q(\mathbf{x})$ represents a point source or sink function. If, given proper boundary conditions, equation (1) is solved for $h(\mathbf{x})$, we can employ Darcy's law to evaluate the fluid velocity vector, $\mathbf{v}(\mathbf{x})$. The fluid velocity in turn is used to solve the second equation describing contaminant transport:

$$\varepsilon\frac{\partial c(\mathbf{x},t)}{\partial t} + \mathbf{v}(\mathbf{x})\cdot\nabla c(\mathbf{x},t) - \nabla\cdot\left[\underset{\approx}{D}(\mathbf{x})\cdot\nabla c(\mathbf{x},t)\right] = Q(\mathbf{x})\left[c_Q - c(\mathbf{x},t)\right] \tag{2}$$

where ε is the porosity, $c(\mathbf{x},t)$ is the contaminant concentration, $\underset{\approx}{D}(\mathbf{x})$ is the dispersion-diffusion tensor which has components defined as:

$$D_{ij}(\mathbf{x}) = a_T\left|v(\mathbf{x})\right|\delta_{ij} + (a_L - a_T)\frac{v_i(\mathbf{x})\,v_j(\mathbf{x})}{\left|v(\mathbf{x})\right|} + D_m\,\delta_{ij} \tag{3}$$

where i and j symbolize spatial direction, a_L and a_T are the longitudinal and transverse dispersivities respectively, D_m is the coefficient of molecular diffusion, $v_i(\mathbf{x})$ and $v_j(\mathbf{x})$ are the components of the velocity vector, and $|v(\mathbf{x})|$ is the velocity magnitude. In addition, $Q(\mathbf{x})$ in (1) represents point sources and sinks of contaminant and c_Q is the concentration associated with Q.

COLLOCATION FINITE ELEMENTS

The domain is discretized into a finite number of rectangular elements with nodes located at element boundary intersections (figure 1). Since the dependent variables, h(**x**) and c(**x**,t), require continuous second order derivatives, the C^1 continuous Hermite cubic functions are chosen as the basis. The bicubic Hermite approximation of a general function, $u(x,y,t)$, is:

$$u(x,y,t) \approx \hat{u}(x,y,t) = \sum_{i=1}^{I} \sum_{j=1}^{J} \left[\begin{array}{c} U_{ij}(t)\,\phi_{ij}^{00}(x,y) + \partial U/\partial x_{ij}(t)\,\phi_{ij}^{10}(x,y) \\ + \partial U/\partial y_{ij}(t)\,\phi_{ij}^{01}(x,y) + \partial^2 U/\partial x \partial y_{ij}(t)\,\phi_{ij}^{11}(x,y) \end{array} \right] \tag{4}$$

where $\hat{u}(x,y,t)$ is the discretized approximation, the grid (figure 1) has I nodes in the x-direction, J nodes in the y-direction, (i,j) is the nodal index at which is defined four time dependent (if applicable) undetermined coefficients (the functional value, the two first order derivatives and the cross derivative), and four space dependent Hermite bicubic basis functions (ϕ^{00}, ϕ^{10}, ϕ^{01} and ϕ^{11}, see Lapidus and Pinder, 1982, page 83 for functional form). These approximations are then substituted into their respective balance equations (1) and (2) yielding two equations in 2[4(IJ)] nodal unknowns (2 dependent variables, 4 undetermined coefficients per node, and IJ nodes).

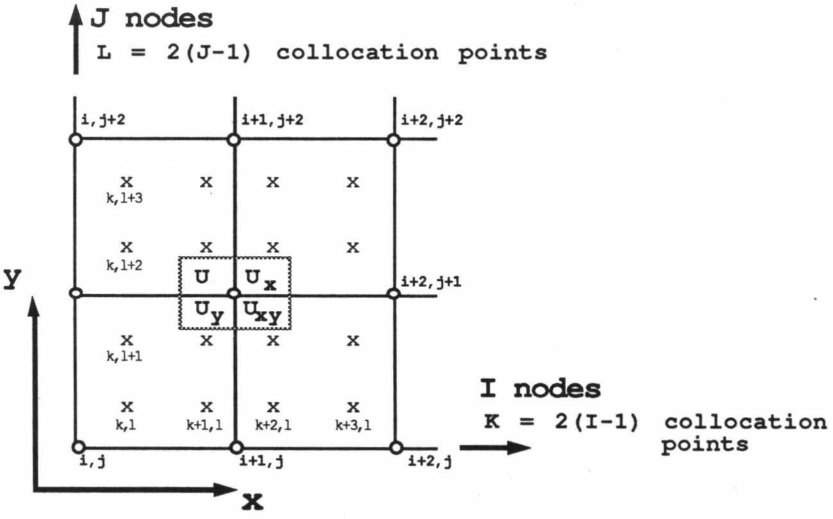


Figure 1 - Depiction of a collocation finite element mesh showing nodal locations (open circles with index i,j), approximate location of collocation points (x's with index k,l), and the degrees of freedom associated with each node (U's).

The nodal unknowns are determined by employing the collocation method: a method of weighted residuals where the weighting function is the displaced Dirac delta function (see Lapidas and Pinder [1982] for a detailed discussion). As a result, the residual errors incurred by using approximated dependent variables in the governing equations are driven to zero at specified points in the domain, called collocation points. If the Gauss quadrature points are chosen as the collocation points, the method is called orthogonal collocation. Using the general function $\hat{u}(x,y,t)$ as a surrogate for each approximated dependent variable, a system of linear algebraic equations is generated by imposing the interpolation constraints:

$$\hat{u}\,(x_k,y_l,t) = u\,(x_k,y_l,t),\ k=1,2\ldots,2(I-1);\ l=1,2,\ldots,2(J-1), \qquad (5)$$

where (x_k,y_l) are the locations of the collocation points (see figure 1). This results in four collocation points per element, and yields 2[4(I-1)(J-1)] equations (2 balance equations each written at 4(I-1) (J-1) collocation points). Note that equation generation requires no formal integrations, and therefore, collocation is computationally analogous to the finite difference method in that equations are written at points in the domain.

To augment the collocation equations such that the number of equations equals the number of unknowns requires that a sufficient number of boundary conditions be specified. Recall that we have four undetermined coefficients at each node. In order that the number of unknowns equal the number of equations, boundary conditions must be imposed such that two of the four undetermined coefficients be specified at midside nodes and three be specified at corner nodes. Therefore, for this discussion the following rule holds: a Dirichlet condition requires specification of both the function and the derivative tangent to the boundary, and a Neumann condition requires specification of both the normal and the cross derivatives. For example, along x-oriented boundary nodes, Dirichlet data includes U_{ij} and $\partial U/\partial x_{ij}$, while Neumann data includes $\partial U/\partial y_{ij}$ and $\partial^2 U/\partial x\partial y_{ij}$.

All other variables requiring spatial representation are defined nodally and interpolated into the element using bilinear Lagrange polynomials. This interpolation is shown for a general function g(x,y) as:

$$g\,(x,y) \approx \hat{g}\,(x,y) = \sum_{i=1}^{I} \sum_{j=1}^{J} G_{ij}\,\Gamma_{ij}(x,y) \qquad (6)$$

where $\hat{g}$ (x,y) is the approximated function, G_{ij} is the coefficient defined at node (i,j) and represents the value of the function at the node, and $\Gamma_{ij}(x,y)$ is the space dependent bilinear Lagrange basis defined at node (i,j) (see Lapidus and Pinder [1982] page 83 for functional form). The combination of using a continuous permeability field and a Hermite interpolated head variable has the advantage of yielding a continuous velocity field to be used in the transport equation.

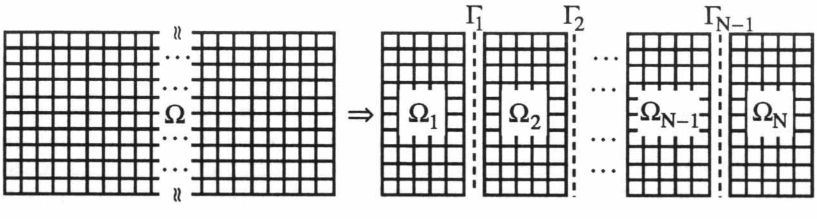


Figure 2: Definition sketch of the domain decomposition algorithm. The domain is partitioned into N subdomains such that, $\Omega = \Omega_1 \cup \Omega_2 \ldots \cup \Omega_{N-1} \cup \Omega_N$, separated by interfaces Γ_i, i = 1,2,...,N-1. Note that the partitioning cuts Ω in one dimension only, and that horizontal partitioning is equally feasible.

APPLICATION OF THE DD-CG ALGORITHM

In this section we present a procedural summary of the implementation of the DD-CG algorithm proposed by Herrera et al. (1994) using the collocation discretization presented above. Consider the domain decomposition shown in figure (2). For this discussion we require that the partitioning cuts Ω in one direction only so that the interfaces Γ_i have no nodes in common. This is not a necessary requirement, however, it allows each interface problem to be computed independently.

Let us present the solution of the flow equation (1) for illustrative purposes and present later the specific features of the transport equation (2) which require special attention. As described in Herrera et al. (1994), we seek the solution to (1) as the linear combination of an inhomogeneous and a homogeneous problem. Using an initial guess for the solution of equation (1) as Dirichlet data for interface nodes, the solution to the inhomogeneous problem (equation 2.4 in Herrera et al., 1994) is obtained in each independent subdomain using the collocation method and a direct solver for each system of equations. This calculation results in a solution which has a continuous functional value but which in general has a jump in the mass flux across the interfaces.

The next step is to solve the homogeneous problem (equation 3.1 in Herrera et al., 1994) requiring that its solution has a jump in mass flux across the interfaces which is equal and opposite to that obtained for the inhomogeneous problem. This is referred to as the jump condition in Herrera et al. (1994, equation 2.6). From this we define the jump in mass flux across an interface Γ_i:

$$f(\mathbf{x}) = -\underset{\approx}{k}(\mathbf{x}) \, [\![\partial h/\partial n]\!] \tag{7}$$

where the double bracket denotes the jump operator: $[\![\bullet]\!]_\Gamma \equiv (\bullet)_{\Gamma^+} - (\bullet)_{\Gamma^-}$, and the derivative is with respect to the positive normal (n) direction. This "boundary value problem with prescribed jumps on Γ" will be solved using the CG procedure proposed in Herrera et al. (1994).

During the CG procedure, equation (7) is evaluated using the jump in the Neumann data after the solution of each set of subdomain boundary value problems. $f(\mathbf{x})$ then becomes the Dirichlet data on each Γ_i for the solution of each subdomain homogeneous problem and is also used for the evaluation of the weights α and β (defined in Herrera et al. [1994]). Because the dependent variables are represented by the space spanned by Hermite cubics, the Dirichlet data and thus $f(\mathbf{x})$ must also be represented by the same space. This Dirichlet data can be thought of as the one-dimensional Hermite representation of $f(\mathbf{x})$ along the interface Γ_i. For example, if the interface is along the y-axis, the discrete approximation of f(y) is:

$$f(y) \approx \hat{f}(y) = \sum_{j=1}^{J} \hat{F}_j \, \phi_j^{\,0}(y) + \left(\partial\hat{F}/\partial y\right)_j \phi_j^{\,1}(y) \tag{8}$$

where $\hat{f}(y)$ is the approximate discrete representation. To apply rigorously the theory presented in Herrera et al. (1994), requires projecting the function f on the space spanned by Hermite cubics. However, as in many other numerical applications, the projection operation can be replaced by a simpler procedure. In particular, since we are using Hermites, the use of nodal values is a natural choice. Also, the coefficient **k** on Γ can be approximated in different manners. Other options can be considered, and in the numerical experiments presented here, three different procedures for calculating the Hermite coefficients in (8) have been applied.

<u>Method 1</u>: The Hermite coefficients are calculated using nodal values of $\underset{\approx}{k}(\mathbf{x})$ and the coefficients of the Hermite representation for $h(\mathbf{x})$. That is, if for example, the x-direction is normal to Γ and the y-direction is tangential, then the coefficients in (8) are:

$$\hat{F}_j = -\left(k_{xx} \, [\![\partial h/\partial x]\!] \right)_j \tag{9a}$$

$$\left(\partial\hat{F}/\partial y\right)_j = -\left\{ \left(\partial k_{xx}/\partial y \, [\![\partial h/\partial x]\!] \right)_j + \left(k_{xx} \, [\![\partial^2 h/\partial x \partial y]\!] \right)_j \right\} \tag{9b}$$

where in equation (9b) we have used the chain rule. Note that since $\underset{\approx}{k}(\mathbf{x})$ is represented by linear Lagrange along Γ, the term $(\partial k_{xx}/\partial y)$ at the node can be discontinuous. If in the heterogeneous case $(\partial k_{xx}/\partial y)$ at node j is discontinuous then the value is approximated by the average of the values defined on either side of the node. We also note that even though $h(\mathbf{x})$ is represented by Hermites, the product k_{xx} $[\![\partial h/\partial x]\!]$, is not. Therefore, this method of calculating the coefficients for $f(\mathbf{x})$ is not optimal from a theoretical point of view.

Method 2: Here we apply a more rigorous approach where we project the function $f(\mathbf{x})$ on the space spanned by Hermite polynomials as discussed in Herrera et al.(1994). The coefficients in equation (8) are calculated by solving the following system of equations:

$$\left\{\begin{array}{l} \langle \hat{f}, \phi_j^0 \rangle = \langle f, \phi_j^0 \rangle \\ \langle \hat{f}, \phi_j^1 \rangle = \langle f, \phi_j^1 \rangle \end{array}\right\}; \quad j = 1, 2,..., J \tag{10}$$

where $\hat{f}$ is defined by equation (8), the operation described by $\langle \bullet, \bullet \rangle$ is for example, $\langle g, h \rangle = \int_\Gamma g(\mathbf{x})\, h(\mathbf{x})\, d\mathbf{x}$, and the integrations required in (10) are evaluated using two point Gauss quadrature. In addition, evaluation of f on the right hand side of (10) is achieved by evaluating equation (7) at the Gauss points. Note that nodal values of f and its derivative need not be evaluated. The system of equations defined by (10) is of order 2*(J-1), and the matrix is symmetric with a half bandwidth of four.

Method 3: Here we approximate the permeability on Γ by a constant. Actually in applying the CG procedure the value of the constant is irrelevant and $f(\mathbf{x})$ becomes:

$$f(\mathbf{x}) = -\,[\![\partial h/\partial n]\!] \tag{11}$$

which has the advantage that $f(\mathbf{x})$ is in the space spanned by Hermite cubics. That is, the coefficients for $f(\mathbf{x})$ are calculated directly from the coefficients of the Hermite representation for $h(\mathbf{x})$, where for example if the x-direction is normal to Γ and the y-direction is tangential, then the coefficients in equation (8) are:

$$\hat{F}_j = [\![\partial h/\partial x]\!]_j \text{ and } \left(\partial \hat{F}/\partial y\right)_j = [\![\partial^2 h/\partial x \partial y]\!]_j \tag{12}$$

Finally, with respect to the evaluation of the scalar weights α and β (in Herrera et al., 1994), given the Hermite representation for $f(\mathbf{x})$, the α weight is for example:

$$\alpha^k = \sum_{i=1}^{N-1} \langle \hat{R}^k, \hat{R}^k \rangle_{\Gamma_i} / \langle \hat{P}^k, \widehat{AP}^k \rangle_{\Gamma_i} \tag{13}$$

where the integrations are computed using two point Gauss quadrature, and β is defined in an analogous way. Note that the integrations in equations (10) and (13) require four point Gauss quadrature to be exact, however, computational experiments show that a two point scheme provides the best convergence results. The reason for this result is currently being investigated.

RESULTS FOR THE FLOW PROBLEM

Using a series model flow problems, we will compare the convergence attributes of the DD-CG algorithm using the three methods of evaluating equation (8).

Let us now consider an example from the oil industry, the repeated five-spot well field pattern. Borrowing a problem from Russell et al. (1986), a square domain, 304.8m on edge, is discretized into 48 by 48 6.35m elements. As shown in figure 3, a source well is located in the lower left corner of the mesh, and a sink well is located in the upper right corner; they each pump $18.6 m^3 d^{-1} m^{-1}$. No flow boundary conditions prevail. The permeability field is in general isotropic and heterogeneous. It is generated in the following manner. Each dimension is divided into seven equal sections of seven nodes each. A patch is defined by the inner product of each dimension section. As shown in figure (3a), to each of the resulting 49 patches is assigned a constant isotropic k value defined by: $k = 10^s k_0$, where s is a random number, $0<s<1$, and for these examples $k_0 = 8.6 \times 10^{-7}$ m/d. Thus, the maximum variation in k between patches is an order of magnitude.

For the results shown, the grid is partitioned into four 48 by 12 element subdomains. Results were obtained for both vertical (constant in x) and horizontal (constant in y) partitioning. The location of the partitions is shown in figure 3. For the homogeneous case and for four different realizations of the k field, we note the number of iterations, m, for which the convergence condition, $\| R^m \|_2 \leq 0.001 \| R^0 \|_2$, is satisfied. Table 1 summarizes the results.

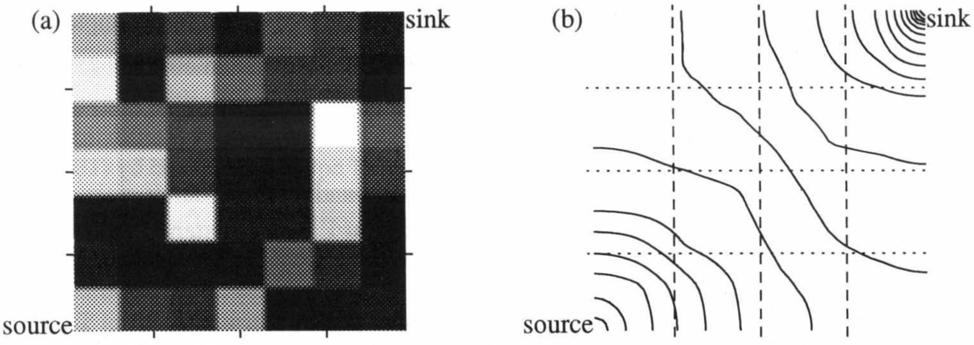


Figure 3: a) plot of the domain showing permeability patch distribution for one realization (light patches reflect higher permeabilities), and tic marks on the boundary representing domain partition locations by connecting marks on opposite faces. b) solution of the head field for the permeability distribution in (a), vertical partition locations are shown by dashed lines and horizontal by dotted lines.

Table 1- Iterations required to solve of the flow problem using four subdomains.

k field	method (1) horizontal partitions	method (1) vertical partitions	method (2) horizontal partitions	method (2) vertical partitions	method (3) horizontal partitions	method (3) vertical partitions
homogeneous	20	20	20	20	20	20
realization 1	82	82	65	69	52	39
realization 2	85	79	69	66	58	54
realization 3	92	97	65	77	58	53
realization 4	88	84	69	71	60	54

Upon review of Table 1 the following conclusions can be drawn. Use of the projection operation [method(2)] substantially improves convergence relative to method (1) and appears to be a cost effective procedure. In most cases that we have treated method (3) provided the fastest convergence. However, in some cases method (3) failed to converge, while the convergence of method (2) is always granted. Thus, a reasonable strategy is to use method (3) and the to use method (2) only when necessary.

SOLUTION OF CONTAMINANT TRANSPORT BY DD-CG

If we discretize equation (2) in time with an implicit backward difference, it can be rewritten in the form:

$$[\varepsilon/\Delta t + Q(\mathbf{x})]\, c^{n+1}(\mathbf{x}) + \mathbf{v}(\mathbf{x})\, \nabla c^{n+1}(\mathbf{x}) - \nabla \bullet \left[\underset{\approx}{D}(\mathbf{x})\, \nabla c^{n+1}(\mathbf{x})\right] = g(\mathbf{x}) \qquad (11)$$

where $\Delta t = (t^{n+1} - t^n)$, n is the time level, $c^{n+1}(\mathbf{x}) = c(\mathbf{x}, t^{n+1})$ and $g(\mathbf{x}) = (\varepsilon/\Delta t)\, c^n(\mathbf{x}) + c_Q\, Q(\mathbf{x})$ is the forcing function. Due to the presence of the first order term this equation is non-symmetric, and the method proposed by Herrera et al. (1994) is not directly applicable. The modifications required are being developed at present. Thus, in this paper we apply a procedure that was quite successful in our numerical experiments. It consists of applying the CG algorithm as discussed in Herrera et al. (1994), realizing that the algorithm will not provide optimal convergence properties, nor will convergence be guaranteed. Also, the definition of the jump condition on Γ_i is modified to be:

$$f(\mathbf{x}) = [\![\partial c/\partial n]\!] \qquad (12)$$

which is in the space spanned by Hermite cubics. Note that the advective part of the flux does not appear in (12) because by definition, $[\![c]\!] = 0$.

Results

We test the application of the DD-CG procedure on the transport equation (11) using the results from the flow problem to define the velocity field. Additional problem definition includes: $\varepsilon = 0.1$, $a_L = 10\, a_T$ (a_L is a variable for analysis below), $D_m = 0$, and c_Q at the source = 1.0. We note that with respect to finite element discretizations there is a constraint on time step size, determined from the grid Courant number (Co), generally Co $\leq$ 1 (Huyakorn and Pinder, 1983). Since mass is transported a relatively short distance per time step, we choose for these examples an extreme domain decomposition which cuts the domain into one element strips. This results in 48 subdomains of dimension 48 by 1 elements. An example of the solutions for the homogeneous case and a heterogeneous case are presented in figure 4.

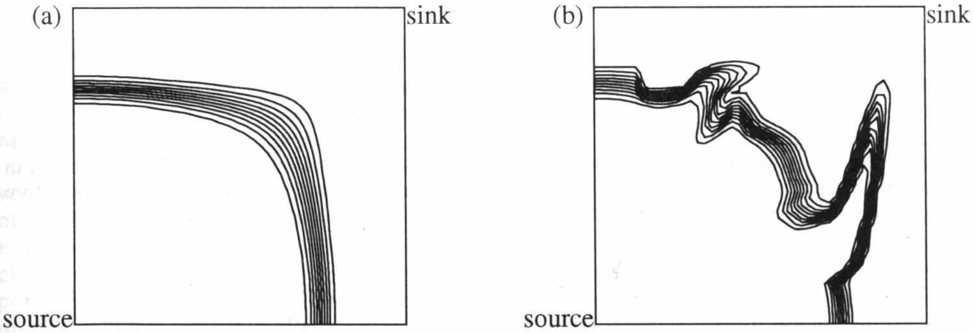


Figure 4: Contour plots for the solution of contaminant transport at 250 days (Δt = 1 day, a_L=0.635m) for (a) the homogeneous k-field and (b) the heterogeneous k-field shown in figure 3a.

Figure 5 presents two computational experiments. In figure 5a, using the homogeneous flow problem as an example, we show the average number of iterations to convergence (using the same convergence criterion as for the flow problem) per Δt as a function of grid Peclet number, Pe=(grid spacing)/a_L. Observe that as dispersion decreases, the number of iterations also decreases. This is because the coupling between subdomains becomes weaker as dispersion is reduced.

In figure 5b, we compare results that were obtained applying this method to the transport problems shown in figure 4. It can be seen that the number of iterations required for the heterogeneous problem is slightly larger than for the homogeneous problem, but the method is still highly efficient. Also note in that figure, that the number of iterations increases in a more or less linear fashion with Δt.

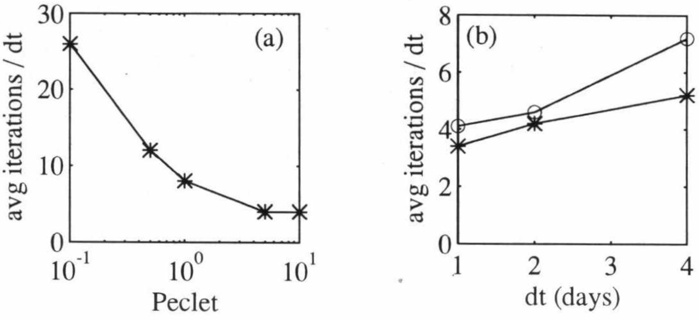


Figure 5 - (a) the relationship between Pe and iteration count for the homogeneous problem (Δt=1d). (b) the relationship between time step size and iteration count for the solution to the problems in figure 4 (Pe = 10): the homogeneous problem (*'s) and the heterogeneous problem (o's).

SUMMARY

In this paper we applied the DD-CG algorithm proposed by Herrera et al. (1994) using a collocation discretization of the equations describing groundwater flow and contaminant transport. For the solution of the flow problem we applied the theory directly and found that approximating the permeability on the interface by a constant reduces the number of iterations required for convergence, in most cases. However, in order to grant convergence it is necessary to use a more precise representation of the interface flux.

The non-symmetric character of the contaminant transport problem precludes a direct application of the DD-CG algorithm, in its present state. However, a modification of it was applied and quite satisfactory numerical results were obtained. This suggests that the proposed algorithm is both robust and efficient.

ACKNOWLEDGMENT. The authors wish to acknowledge the financial support of the U.S. Environmental Protection Agency (EPA) assistance I.D. No. CR-820499. Although the research described in this article has been funded by the U.S. EPA, it has not been subjected to agency review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

REFERENCES

Herrera, I., J. Guarnaccia and G. F. Pinder (1994), "Domain decomposition methods for collocation procedures", in A. Peters et al. (eds.), Computational Methods in Water Resources, Kluwer Academic Publishers, Dordrecht, this volume.

Huyakorn, P. S., and G. F. Pinder (1983), Computational Methods in Subsurface Flow, Academic Press, New York.

Lapidus, L., and G. F. Pinder (1982), Numerical Solution of Partial Differential Equations, John Wiley, New York.

Russell, T., M. Wheeler and C. Chiang (1986), "Large-scale simulation of miscible displacement by mixed and characteristic finite element methods", W. Fitzgibbon (ed.), SIAM, Philadelphia, pp 85-107.