Optimization of Transmissivity Measurement Networks for Head Uncertainty Reduction

by

Swen O. Magnuson

Submitted in Partial Fulfillment

of the Requirements for the

Degree of Master of Science

New Mexico Institute of Mining and Technology

Soccoro, New Mexico

March, 1988

Preface

This independent study paper is composed of two parts. The first and largest part is written for submittal as a paper to Water Resources Research. The reader looking for an abstract is referred to the beginning of the proposed paper. The second part consists of appendices which are not referred to in the proposed paper. In the appendices, an additional application of the network design algorithm is presented along with other issues not addressed in the proposed paper. Additionally, recommendations for future research, code listings, and data files for the results in the proposed paper are included.

Acknowledgments

I would like to thank my advisor and friend, Dr. John L. Wilson, for his support, guidance, and enthusiasm during the course of this work. His passion for worthwhile research was infectious, almost. The insight of Dr. Allan Gutjahr was also helpful at several stages.

Annette Schafer-Perini's help with coding, graphics, and humor is gratefully acknowledged. Tony Zimmerman's and Dave Peterson's interest in the work was also beneficial.

The National Science Foundation supported, in whole, the contents of this work through a Graduate Minority Fellowship. My sincerest thanks to my grandmother, Mary Holstein, for her wisdom and foresight which allowed me the opportunity to benefit from my heritage. My parents emotional and financial support throughout my education must be acknowledged; without it, nothing would have been possible.

Others are deserving of thanks as well for their friendship and support. These include: Earl Mattson, Alva Parsons, Steve Conrad, and Robert Knowlton. Lastly, I want to thank Deborah McElroy for giving me a reason to finally get finished with this degree.

TABLE OF CONTENTS

Optimization of Transmissivity Measurement Networks for Head Uncertainty Reduction

Swen Magnuson and John L. Wilson

Geoscience Department, New Mexico Institute of Mining and Technology

ABSTRACT

Transmissivity data can be processed using geostatistical methods and used to guide future measurement efforts. Often this analysis focuses on minimizing the kriging variance as the principle objective function of a network design optimization problem. Unfortunately decision makers have essentially no direct interest in transmissivity fields; therefore, more applicable objective functions are needed. These objective functions should directly represent quantities of interest to the decision makers. One such class of objective functions concerns head uncertainty, since uncertainty in heads can affect decisions regarding pumping lifts or contaminant concentrations and travel times. Head uncertainty can be represented through the head covariance matrix obtained from first order, second moment methods.

We present a network design algorithm which uses first order, second moment methods to optimize the location of additional transmissivity measurements in order to maximize the reduction in the trace of the head covariance matrix, which represents some overall measure of system uncertainty. The algorithm is tested on a hypothetical groundwater basin which has an underlying synthetic transmissivity field. The algorithm results show reasonable agreement with the optimal locations found by an alternative method which samples each location in order to determine the maximum trace reduction. Nonlinearity of the relationship between head and transmissivity necessitates the use of scaling to prevent the algorithm from resampling a location when additional measurements of the underlying field are made.

Introduction

Numerical models of groundwater flow are commonly used by groundwater managers to aid in making decisions. Predictions of heads are a main output of models upon which decisions are made. A variety of uncertain parameters need to be estimated before a model can be applied; for example, transmissivity, storativity, pumping rates and locations, initial conditions, and boundary conditions. We consider only transmissivity in this paper. Transmissivity estimates are uncertain due to the spatially varying nature of transmissivity, and can be made based on measurements at well locations or from geologic knowledge of the region being modeled. Kriging of transmissivity measurements in order to determine the regional transmissivity distribution is often done in modeling applications. Due to the uncertain nature of the transmissivity estimates, a network design problem addressed by managers is where to make additional measurements of transmissivity in order to improve decisions. A common objective is to reduce the uncertainty of the kriged transmissivity field [e.g. *Delhomme,* 1978; *Hughes and Lettenmaier,* 1981; *Bogardi et al., 1985*]. Unfortunately decision makers have essentially no direct interest in transmissivity fields; therefore, more applicable objective functions are needed.

When possible, network design objective functions should directly represent quantities of interest to the decision makers. One such class of objective functions concerns head

uncertainty, since uncertainty in heads can affect decisions regarding pumping lifts or contaminant concentrations and travel times (which depend on velocity fields). Head uncertainty can be represented through the head covariance matrix obtained from first order, second moment methods [*Dettinger and Wilson*, 1981].

There are several objective functions one could minimize in relation to the head covariance matrix. The objective function on which we focus in this paper is the trace of the head covariance matrix, which is the sum of the diagonal entries in the matrix and represents some overall measure of system uncertainty. Other objective functions relating to the head covariance matrix could be more applicable depending on the situation, and include minimizing the determinant, the spectral radius, or the maximum variance. The latter is tantamount to selecting the location with the maximum head variance as the sampling location.

Further extensions of the head uncertainty reduction approach could include transient flow or the influence of uncertain boundary conditions. Another extension would be to incorporate an inverse problem solution where additional direct transmissivity measurements would improve the initially estimated transmissivity field, i.e. the prior information. Likewise, head measurements could also be used to further condition the estimated transmissivity field. The methodology used in the trace analysis is not limited to the particular application presented. For example, uncertainty reduction in pumping lifts could be obtained by a simple linear transformation. Uncertainty reduction in travel times could be estimated by an additional chain rule operation. If we were interested in other decision variables such as pollutant concentration, rather than head, uncertainty in concentrations could be minimized. Determination of where to sample uncertain boundary conditions, heads, or concentrations are other possible extensions.

An important issue not addressed in this paper is that the objective function of reducing head uncertainty competes directly against objectives related to improving variogram identification. Variogram identification requires clustering of measurement locations while uncertainty reduction places measurements in zones away from existing measurements. *Warrick and Myers* [1987] present a sampling algorithm for optimizing the number of measurement couples in distance separation classes that are determined a priori. This objective function leads to a better estimate of the variogram parameters, and perhaps even the variogram model, but as a consequence tends to cluster the measurement locations.

We present a particular head uncertainty minimization approach based on derivatives of the trace of the head covariance matrix, which we call the "trace sensitivity approach." This method attempts to determine the optimum location to sample transmissivity in order to achieve the maximum reduction in the sum of head variances over the system. There are alternative approaches to this objective function. For example, co-kriging of transmissivity and heads including an inverse solution may be more successful in handling the nonlinearity of the relationship between transmissivity and heads [*Gutjahr and Wilson,* 1985]. A nonlinear search algorithm utilizing bordered matrix factorization [*Rouhani*, 1985] for estimating kriging covariances could be used to find the location of maximum trace reduction directly. While this approach would succeed in locating the optimal sampling location it would not lend itself to a better overall understanding of the system which is a benefit of the trace sensitivity approach.

The derivation of the trace sensitivity algorithm is presented first followed by numerical applications. Each application uses the same underlying transmissivity field which is generated synthetically. The first application is set up as a problem in determining a priori where to locate three wells in order to achieve the maximum reduction in head uncertainty, without actually making any measurements as an interim step. The kriging results from the existing measurements are used to estimate the values that would be obtained by measuring transmissivity at the new well locations. In the second application, after each well is located, the actual values from the underlying field is used as the measurement at the new well location. A scaled version of the trace sensitivity algorithm is introduced to avoid reselecting the same location. Lastly, an attempt is made to improve the algorithm's performance by increasing the grid discretization in two zones indicated as sensitive for the reduction of the head covariance matrix trace.

Trace Sensitivity Algorithm

The trace of a matrix is defined as the sum of the diagonal entries. Since the diagonal entries of the head covariance matrix are the head variances, the trace of the head covariance matrix is the sum of the head variances for each location in a discretized system. The derivative of the trace of the head covariance matrix with respect to a change in log transmissivity at one of the locations in the system is what we call the trace sensitivity. Logarithms of transmissivity to base ten are used throughout this derivation and the numerical application presented below.

By first order, second moment methods the steady state head variance at a node i in a discretized system can be written

$$Var\,(h_i) = \sum_{j=1}^{m}\sum_{k=1}^{m} \frac{\partial h_i}{\partial y_j}\,\frac{\partial h_i}{\partial y_k}\,Cov\,(y_j, y_k)\,, \tag{1}$$

where y represents log transmissivity, h represents head, and m is the number of parameters y [*Dettinger and Wilson*, 1981]. The objective function we choose to minimize is

$$Trace\,[\,Cov\,(\underline{h})\,] = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} \frac{\partial h_i}{\partial y_j}\,\frac{\partial h_i}{\partial y_k}\,Cov\,(y_j, y_k)\,, \tag{2}$$

where n is the number of nodes, $\underline{h}$ is the head vector, and $Cov(\underline{h})$ is the head covariance matrix.

The derivative of the trace with respect to log transmissivity at a parameter location l is

$$\frac{\partial\, Trace}{\partial\, y_l} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} \frac{\partial}{\partial\, y_l}\left(\frac{\partial h_i}{\partial y_j}\,\frac{\partial h_i}{\partial y_k}\right) Cov\,(y_j, y_k)\,, \tag{3}$$

where the derivative operator can be brought inside the summations since they are a linear combination. Rewriting, utilizing a product rule expansion, yields the equation for the derivative of the trace with respect to a change in log transmissivity at a location l,

$$\frac{\partial\, Trace}{\partial\, y_l} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} \left(\frac{\partial^2 h_i}{\partial y_l\,\partial y_j}\,\frac{\partial h_i}{\partial y_k} + \frac{\partial^2 h_i}{\partial y_l\,\partial y_k}\,\frac{\partial h_i}{\partial y_j}\right) Cov\,(y_j, y_k)\,. \tag{4}$$

We choose to treat all nodes as being equally important. An alternative method would allow subjective or areal weighting of each location in order to allow a decision maker to treat head uncertainty at some nodes as being more important than others or to handle non-uniform grids.

Calculating this derivative for each location in the domain produces a map of trace sensitivities. This map can be used to guide the search for the optimal location or locations for additional sampling. Areas of higher sensitivity indicate likely sample locations. If only one sample is going to be taken, it should presumably be at the location with the maximum absolute value of trace sensitivity. This trace sensitivity approach was first suggested by John L. Wilson at the 1979 Fall American Geophysical Union meeting.

The trace sensitivities calculated using the above method are punctual, i.e. they are based on derivatives of head with respect to log transmissivity at the expected value of log transmissivity at exact locations. As the estimated log transmissivities change with the addition of new measurements, these derivatives may behave in a nonlinear fashion; therefore, the trace sensitivity analysis may show nonlinear behavior due to the changing derivatives. In the hypothetical application discussed later, this nonlinear behavior allows the trace sensitivity algorithm to reselect a sampling location even after an additional transmissivity measurement is located there. Scaling the algorithm results by the transmissivity kriging covariance matrix prevents resampling. The expression for the scaled trace sensitivity at a location l is

$$Scaled\ \frac{\partial\ Trace}{\partial\ y_l} = \sum_{p=1}^{n} \frac{\partial\ Trace}{\partial\ y_p}\ Cov\ (y_l, y_p)\ . \qquad (5)$$

The trace sensitivity approach requires calculation of the first and second derivatives of head with respect to transmissivity. These derivatives are determined by differentiating the steady state flow equation twice and solving for the desired derivatives at the proper steps [*Townley,* 1984]. Time and expense is saved in computing both the first and second derivatives by taking advantage of the coefficient matrix sparsity [e.g. *Townley and Wilson,* 1985]. For the most part this approach is contained in the stochastic flow code CERT [*Townley,* 1983; *Townley and Wilson,* 1984]. We have also developed an adjoint method [see e.g., *Townley and Wilson,* 1985] for computing the trace sensitivities, but it does not appear to offer any advantage over the direct approach because the trace objective function requires calculation of all the first derivatives in order to build the load vector for

the adjoint problem. Since the second derivatives can be efficiently and directly computed using the same information, the adjoint approach does not offer any advantage.

We have also developed an alternative straightforward method which we call perturbation trace reduction to evaluate the trace sensitivity algorithm results. We perturb the system by adding a transmissivity measurement and then calculate the resulting reduction in the trace of the head covariance matrix. Although used primarily to check the trace sensitivity results, the perturbation trace reduction method could be used by itself to solve the measurement network design problem posed herein. Perturbation trace reduction is defined on a nodal basis. It is the trace of the head covariance matrix determined from the original transmissivity measurements minus the trace of a new head covariance matrix determined with one additional transmissivity value at a node of the discretized system. We repeat this process for each node of the discretized system to produce a map which shows the optimal location at which to sample to achieve maximum trace reduction. The expression for the perturbation trace reduction at a location l is

$$Trace\ [\ Cov\ (\underline{h})\ ]_p - Trace\ [\ Cov\ (\underline{h})\ ]_{p+1}, \tag{6}$$

where p indicates the original number of transmissivity measurements and $p + 1$ indicates one additional measurement at location l. Alternatively this could be written as

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m}\frac{\partial h_i}{\partial y_j}\frac{\partial h_i}{\partial y_k}\left[\ Cov\ (y_j, y_k)_p - Cov\ (y_j, y_k)_{p+1}\ \right] \tag{7}$$

as long as the additional transmissivity measurement has no effect on the derivative terms, i.e. the additional measurement is the same as the kriged value at location l with p measurements.The perturbation trace reduction method is computationally intensive since it requires one run of the CERT code for each node in the system while the trace sensitivity algorithm requires only one run to produce a complete map.

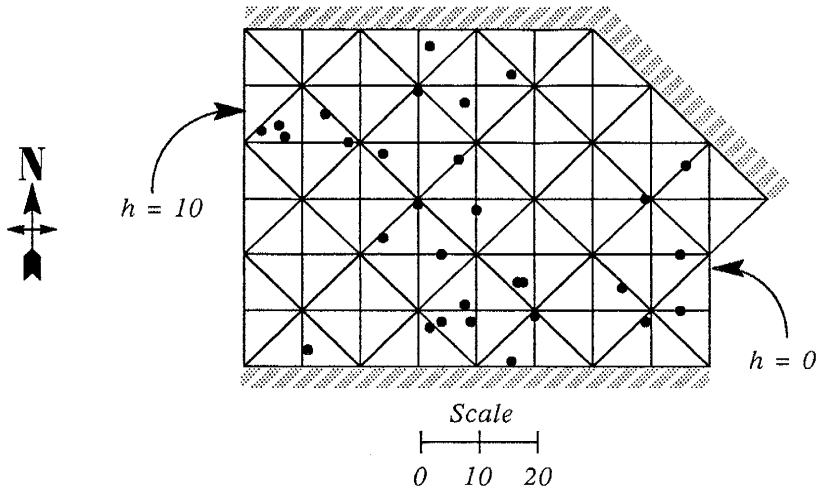

Figure 1. Finite element grid, boundary conditions, and thirty measurement locations.

Numerical Application

Consider the hypothetical grondwater basin shown in Figure 1 with known prescribed heads at each end and no-flow boundaries on the sides. In addition to the boundaries being certain, there are no local sources of recharge or pumping within the basin. Figure 2 represents the underlying log transmissivity field for this basin which is generated using the turning bands method with an exponential covariance structure [*Mantoglou and Wilson*,

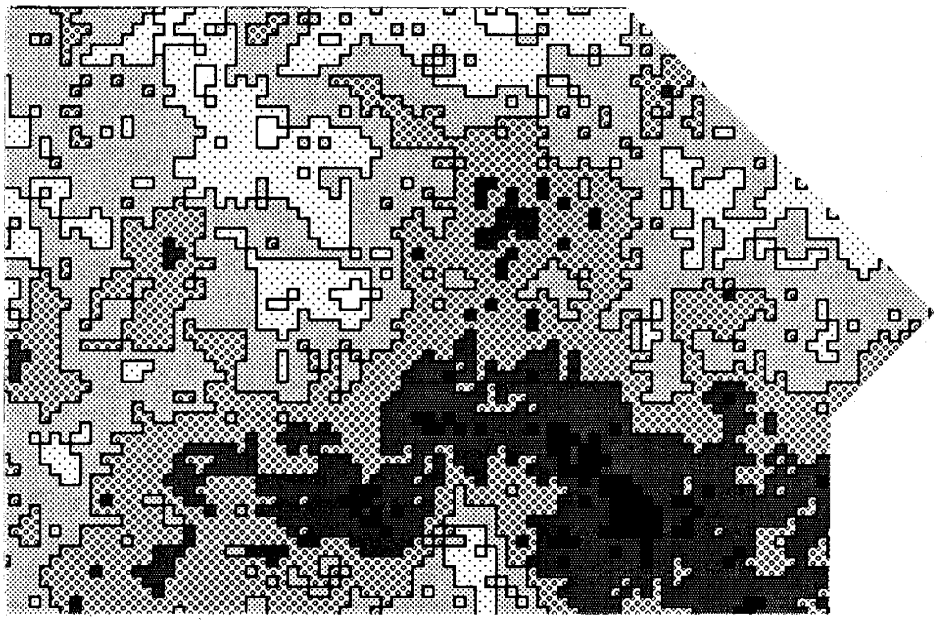

Figure 2. Synthetic field generated by turning bands method, exponential covariance, variance = 1.0, correlation length = 20. Different scale from Figure 1.

1982]. The prescribed theoretical mean for the synthetic field is 2.0 with a variance of 1.0 and a correlation length of 20. Each shading pattern represents values ranging within one sample standard deviation based on all the values from the synthetic field. The three darker patterns represent areas of log transmissivity which fall below the generated field population mean of 2.06 and vice versa for the lighter patterns. Pixel shading patterns are used for representation of the synthetic field because a contour plot using linear interpolation on a finely discretized finite element grid would be excessively noisy and difficult to interpret. The distance scale is different for Figure 2 so the detail in the underlying field can be seen. There are thirty irrigation wells with transmissivity measurements in the basin (see Figure 1). The measurement locations are randomly generated using a bivariate beta distribution with weights, $\alpha = \beta = 1.3$, in order to avoid putting a well on a boundary.

If a groundwater manager had the option to place additional wells for measuring transmissivity, where should they be located in order to maximize reduction in head uncertainty throughout the basin? A two-dimensional finite element grid for the basin is shown above in Figure 1. For convenience, we place transmissivity measurements at locations corresponding to nodes of the finite element grid although this in not strictly a requirement. This is equivalent to setting $n = m$ in equation (4). The flow field for the basin is simulated by kriging the existing thirty transmissivity measurements onto the nodes of the finite element grid (see Figure 1). Then using the first order, second moment, finite element approach contained in CERT, the heads and related standard deviations are calculated.

We choose to use a nodal rather than element parameterization of transmissivity for ease of data representation by contouring. Transmissivity is linearly interpolated between nodes. In any case, the solution strategy for linear elements calls for averaging the nodal values to come up with an equivalent element property. Both nodal and element parameterizations of transmissivity imply some kind of averaging. Unless the elements are significantly smaller than the correlation length or range of the underlying field, the variance of the transmissivities should be reduced. *Dettinger & Wilson* [1979] found in one-dimensional flow that variance reduction by areal averaging [*Vanmarke,* 1983] is appropriate. In two- or three-dimensional flow fields, however, areal or volumetric averaging is not necessarily appropriate [e.g., *Townley,* 1983; *Townley and Wilson,* 1984]. Variance reduction in numerical simulation models is an area of ongoing research and is not addressed in this paper. Variance reduction may affect the values of trace sensitivity calculated by the algorithm but it is not likely to affect the overall thrust of the approach--selecting the optimum transmissivity measurement location for head

uncertainty reduction. Current practice says that in order to adequately model the effect of the synthetic transmissivity field on the resulting heads, a discretization of at least five nodes per correlation length is necessary and preferably much more than that [*Dettinger and Wilson,* 1979; *Black and Freyberg,* 1987]. We come close to the minimum with three nodes per correlation length and justify this number in that our results for the most part are reasonable. This issue of discretization level is addressed later.

Original Situation with Thirty Log Transmissivity Measurements

Maximum likelihood parameter estimation [*Kitanidis and Lane,* 1984] with an exponential variogram as the chosen model is used to estimate the variogram from the thirty log transmissivity measurements. Measurement noise, N(0,0.1), is added to the measured value from the underlying field for each of the thirty well locations. The estimated variogram has a sill of 0.992 and a correlation length of 20.8. Interestingly, no nugget effect is estimated by the maximum likelihood procedure even in the presence of the additional noise. This estimated variogram is then used for kriging the thirty measurements onto the nodes of the discretized system. Results of the kriging exercise, the first order, second moment estimated heads, and their associated uncertainties are shown in Figure 3. All contouring in these figures is done using linear interpolation along the sides of the triangular finite element grid. The kriged log transmissivity field (Figure

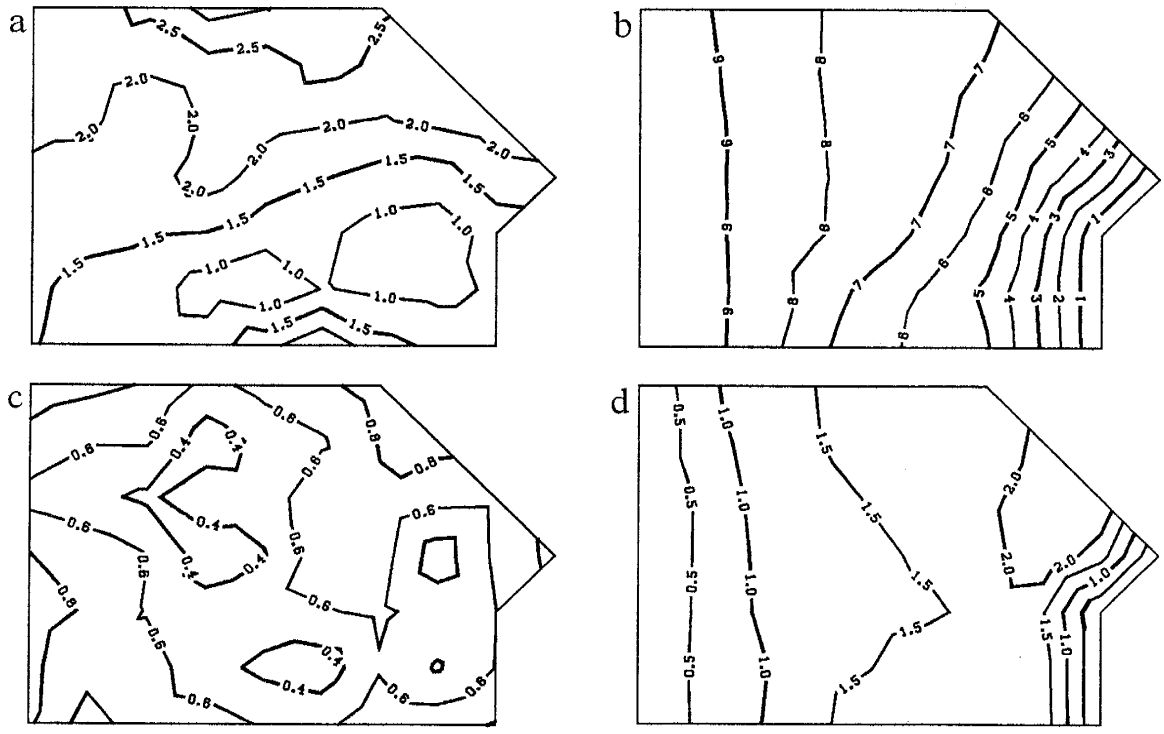

Figure 3. a) Kriged log transmissivity field, b) First order mean head field, c) Log transmissivity standard deviations, and d) First order, second moment head standard deviations.

3a) can be compared to the synthetic log transmissivity field (Figure 2) since both represent a continuous field.

The location with the maximum log transmissivity kriging variance is in the extreme northwest corner of the domain and the second highest variance is along the junction of the north and northeast boundaries (see Figure 3c). The maximum head variance is along the slanted northeast side (Figure 3d). Already it appears that to estimate heads the optimum location to place additional measurements for transmissivity is not the same as that of the maximum kriging variance. In the next sections we present results from the trace sensitivity algorithm which are based on knowledge of the original situation as just described.

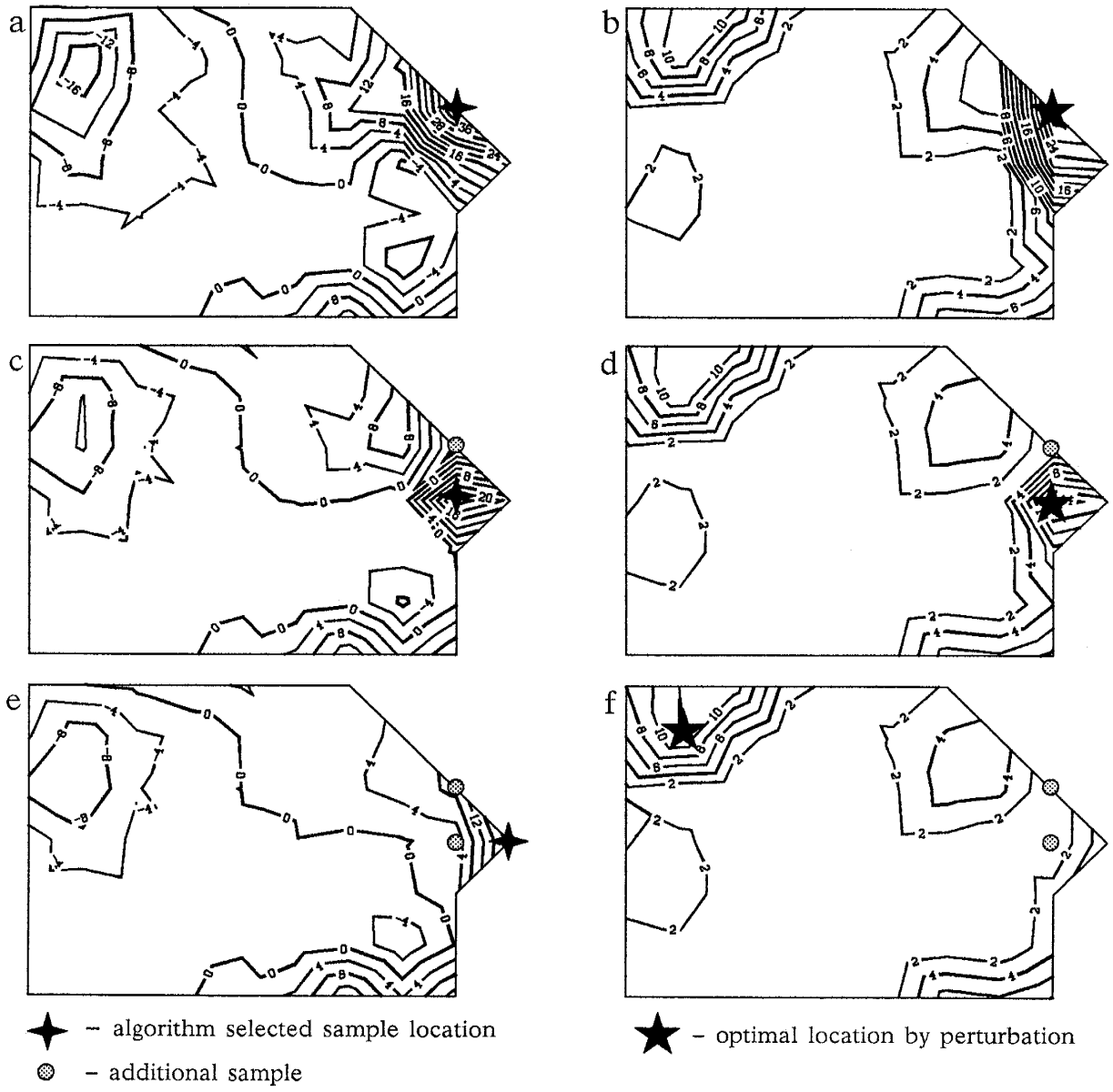

Figure 4. Comparison of trace sensitivity algorithm to perturbation trace reduction for a sequence of runs with kriging results added as measurements at each step. Thirty original measurements; a)trace sensitivity results and b)perturbation trace reduction. Thirty-one measurements with kriged value added as a measurement at the additional sample location; c)trace sensitivity and d)perturbation trace reduction. Thirty-two measurements with second kriged value added at second sample location; e)trace sensitivity results and f)perturbation trace reduction.

Sequential Design Without Sampling

For the first application of the trace sensitivity algorithm consider a problem where a manager needs to site three additional wells. Each well must be sited without the benefit of actually measuring the value of transmissivity at either of the other two sites. Instead these values are estimated by kriging the thirty existing measurements. Trace sensitivity results for the algorithm are shown if Figures 4a, 4c, and 4e, in which the three

measurement locations are selected sequentially. The maximum trace sensitivity location selected by the algorithm at each step can be compared to the perturbation trace reduction map (see Figures 4b, 4d, and 4f). We use perturbation trace reduction, defined above, as an alternative method to locate the optimal sampling location. The measurement values added at each node for the perturbation trace reduction maps in Figure 4 are the result of kriging the existing thirty measurements. Bordered matrix factorization is not currently used in the kriging process for each of these perturbation trace reduction runs, although considerable time and expense could be saved if it were implemented. The locations of the minimums and maximums in the contour maps are what is important when comparing trace sensitivity results to perturbation trace reductions, not the numerical value at the peaks even though they have the same units (L^2).

By comparing Figures 3a, 4a, and 4b it is obvious the optimal location to sample transmissivity in order to maximize head uncertainty reduction is not the same as the location of maximum kriging variance for the transmissivity field calculated from the original thirty measurements. The maximum kriging variance is at the northwest corner of the flow domain while the optimal location for sampling is along the slanted northeastern boundary of the domain. The optimal location for sampling in Figure 4a occurs at the same location as the maximum head variance determined with the original thirty measurements. This is circumstantial as becomes clear after an additional measurement is placed there (see Figure 4c and 4d). Although not shown, the location of the maximum head variance remains at the same location as the additional sample, and the optimal sampling location shifts southward one node. For two additional samples, the difference in location between maximum head variance and optimal sampling location is even greater (see Figure 4e and 4f). The head standard deviations after the addition of the first two kriged transmissivity measurements are shown below in Figure 7a. The maximum head variance occurs along the southern boundary while the maximum trace sensitivity location and the maximum perturbation trace reduction location occur elsewhere.

The locations of the maximum trace sensitivity and the maximum perturbation trace reduction show agreement for the first two applications of the algorithm, with thirty and thirty-one measurements respectively. But on the third application the locations selected differ substantially (see Figures 4e and 4f). While the source of this divergence is not known exactly, it is likely caused by the nonlinear relationship between transmissivity and head discussed earlier.

Positive and negative peaks in Figure 4a define sensitive areas in which changes in log transmissivity would have greater affects on the trace of the head covariance matrix. For the most part these peaks correspond well with those in the trace perturbation map for the same set of thirty measurements (see Figure 4b). Two possible explanations exist for the particular locations of these sensitive areas. First, there is a relatively low log transmissivity zone in the south central portion of the kriged transmissivity field (see Figure 3a). This low transmissivity zone actually extends to the southeastern corner of the domain but the original thirty transmissivity measurements fail to fully pick up this extension of the zone. The low transmissivity zone in the kriged field forces water to flow around it, increasing the importance or sensitivity of the trace to the value of transmissivity in those regions in which flow is focused. The area north of the low transmissivity zone is larger than the area south of it and therefore receives more of the flow. This makes the northern region more important relative to the southern region below the low transmissivity zone. This behavior is captured in both the trace sensitivity and perturbation reduction results as can be seen by the relative magnitude of the contours in both of these regions. The optimal location in the sensitive area occurs along the northern no-flow boundary. This is likely due to the influence of the existing log transmissivity measurements nearby which force the sampling location onto the boundary. Once an additional measurement is located on the no-flow boundary, the next algorithm selected location moves off the boundary into the interior of the basin.

The second reason for the location of the more sensitive zones can be explained by the similarity of our two-dimensional flow regime to a simple one-dimensional flow problem. Figure 5 shows the effect on head profiles for a one-dimensional flow field in an

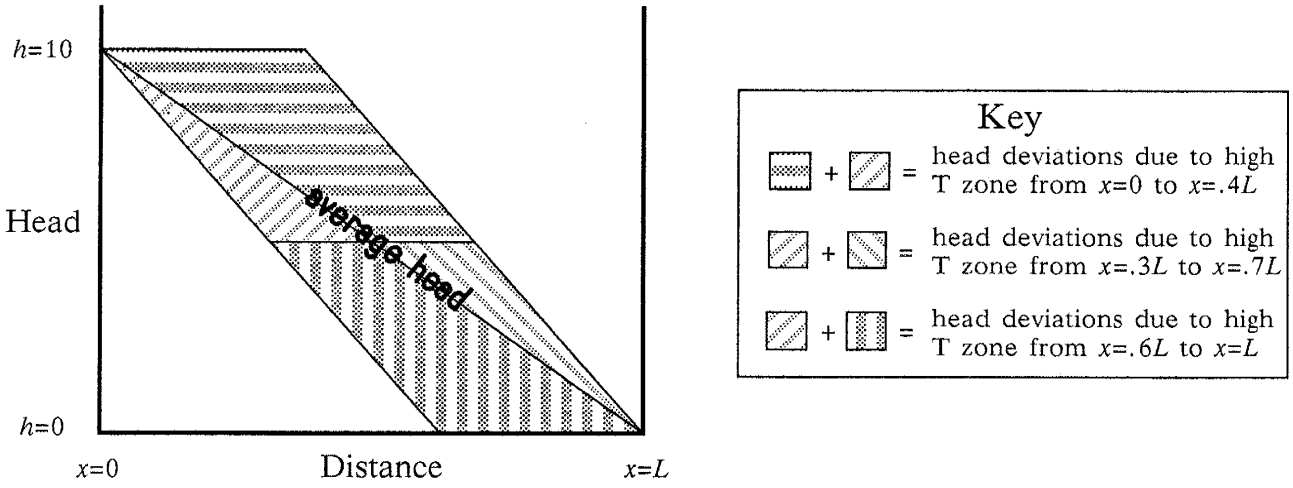

Figure 5. Head deviations due to the presence of a high transmissivity zone at different locations in a one-dimensional flow field.

otherwise homogeneous aquifer when a zone of high transmissivity or permeability is introduced at various locations in the flow field. The shaded area between the head lines and the average head line represent the head deviations due to the high transmissivity zone. The head deviation area is greater when the high transmissivity zone is located near the boundary than when it appears in the middle of the flow domain; therefore, a change in transmissivity at or close to the prescribed head boundary will have a greater effect on head deviations than a change in the middle of the flow field. Extending this concept to the two dimensional flow in the hypothetical problem explains why the sensitive zones tend to appear near the prescribed head boundaries.

As the kriging values from the original thirty measurements are added at the indicated locations, the trace sensitivity and perturbation reduction results both decrease in absolute value (see Table 1 and Figure 4). Although intuitively appealing, the trace sensitivity reduction is coincidental. The addition of a kriged transmissivity value at a node does not affect the resulting transmissivity field since kriging preserves measurement values in the absence of a nugget effect. Therefore, the flow field does not change and the heads and head derivatives stay the same. Only the kriged transmissivity covariance matrix is affected by the addition of a measurement location. Entries in the matrix which feel the affect of the measurement location get reduced. The heads and head derivatives would not stay the same if we used a second order estimate for heads. The reduced covariance matrix would affect the second order correction and therefore the resulting estimated heads and head derivatives [see *Townley,* 1984]. By referring to (4) it can be seen that the resulting trace sensitivities calculated using this "reduced" covariance matrix will not necessarily be smaller since the sign on the first and second derivatives can be positive or negative. If, for example, the reduced covariance terms match with mostly positive derivative products, the resulting trace sensitivities could increase in absolute value. This

Table 1. Global Minimum and Maximum Trace Sensitivity for Sequential Design with Kriged Values and Trace Reduction at Each Step

	Trace Sensitivity with Kriged Values		Trace Reduction after adding Kriged Value	
	Minimum (L^2)	Maximum (L^2)	Trace (L^2)	Reduction (L^2)
Initial Thirty Measurements	-19.9	46.2	114.1	-
First Measurement Added	-12.3	26.0	84.5	29.6
Second Measurement Added	-11.9	22.1	69.4	15.1

would result in the trace sensitivity algorithm reselecting a location even though it already has a kriged measurement added. In comparison, no further perturbation trace reduction can be achieved at the node where a measurement is added (see Figures 4d and 4f). Adding the same kriged measurement at a node again has no effect on the head derivatives or the transmissivity covariance matrix and therefore, a location cannot be reselected.

Sequential Design With Sampling of Underlying Field

In this application of the trace sensitivity approach we consider the same design problem outlined in the previous section with one difference. The manager is able to measure the actual value of transmissivity at each additional site before the next location is determined. The sequence of trace sensitivity results for this application are represented in Figure 6. The initial trace sensitivity results with the original thirty measurements are not shown since they are the same as Figure 4a. The sample values from the underlying field have measurement noise N(0.,0.1) added to them in order to be consistent with the initial noisy thirty measurements.

The first and second locations selected in this application are the same as before. The difference in the third selected location (compare Figures 6b and 4e) is explained partly because of the transmissivity value measured from the underlying field at the second selected site. After a log transmissivity measurement from the underlying field is added, the initial estimated variogram is used to krige the transmissivity field at each step. Since the estimated transmissivity field can change after each sample is taken in this case, the first and second derivatives of head can also change. This explains why the algorithm selected the same location twice, once in Figure 6a and again in Figure 6b. The value of

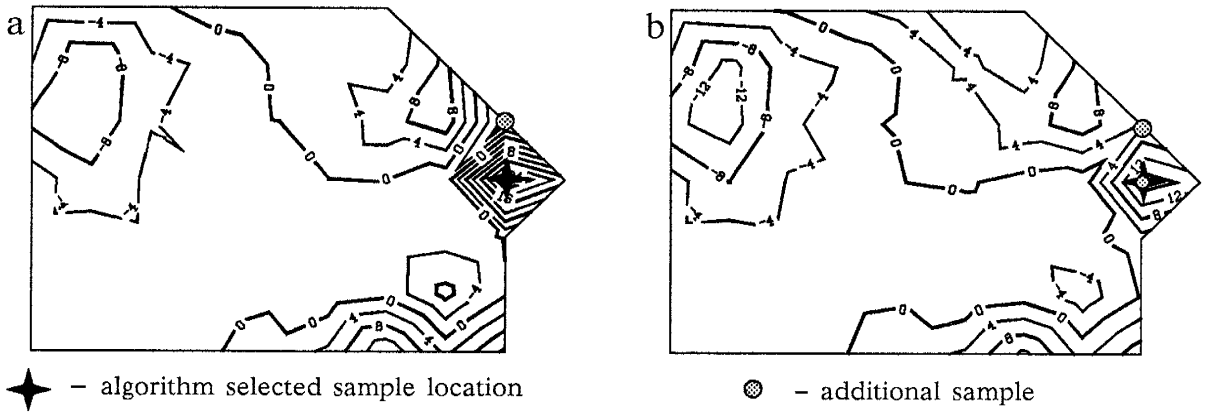

Figure 6. Sequential design with trace sensitivity algorithm. Value added at each step is a measurement from the underlying field. a)Trace sensitivity after adding first measurement and b)after adding second measurement. Compare to Figures 4 c and e.

Table 2. Global Minimum and Maximum Trace Sensitivity for Sequential Design with Measurements from the Underlying Field and Trace Reduction at Each Step

	Trace Sensitivity with Field Measurements		Trace Reduction after adding Measurement	
	Minimum (L^2)	Maximum (L^2)	Trace (L^2)	Reduction (L^2)
Initial Thirty Measurements	-19.9	46.2	114.1	-
First Measurement Added	-11.8	28.7	83.3	30.8
Second Measurement Added	-14.0	18.6	76.4	6.9

the transmissivity measured from the underlying field selected at the location shown in Figure 6a was higher than the value from the kriging results on the original thirty measurements. When this new value was included in the kriging exercise for the second step in this application, the value of transmissivity in the region of the sample location was higher. This further focused the flow field through the northern region and increased the sensitivity of the trace to transmissivity values in this region.

The trace sensitivity results from adding measured values from the underlying field and the resulting trace reductions are presented in Table 2. The transmissivity dependent nonlinear behavior of the head derivatives can be seen in the minimum trace sensitivity at each step. The minimum value increases between adding the first and second measurements.

The head standard deviations that result after addition of two values at sample locations selected by the trace sensitivity approach are presented in Figure 7. The effect of using kriging results from the original thirty measurements and of using actual measurements from the underlying field at the selected locations are shown in Figures 7a and 7b, respectively. For our numerical example, the addition of actual measurements from the underlying field decreases the overall head variances more than the addition of kriging values. Since the kriging covariance is the same for both these cases, the differences are due to the nonlinear changes in head derivatives. Figure 7b contains the additional information of two actual transmissivity measurements and is a better estimate of the resulting head uncertainty than Figure 7a. This would be the case even if the variances in Figure 7b were larger.

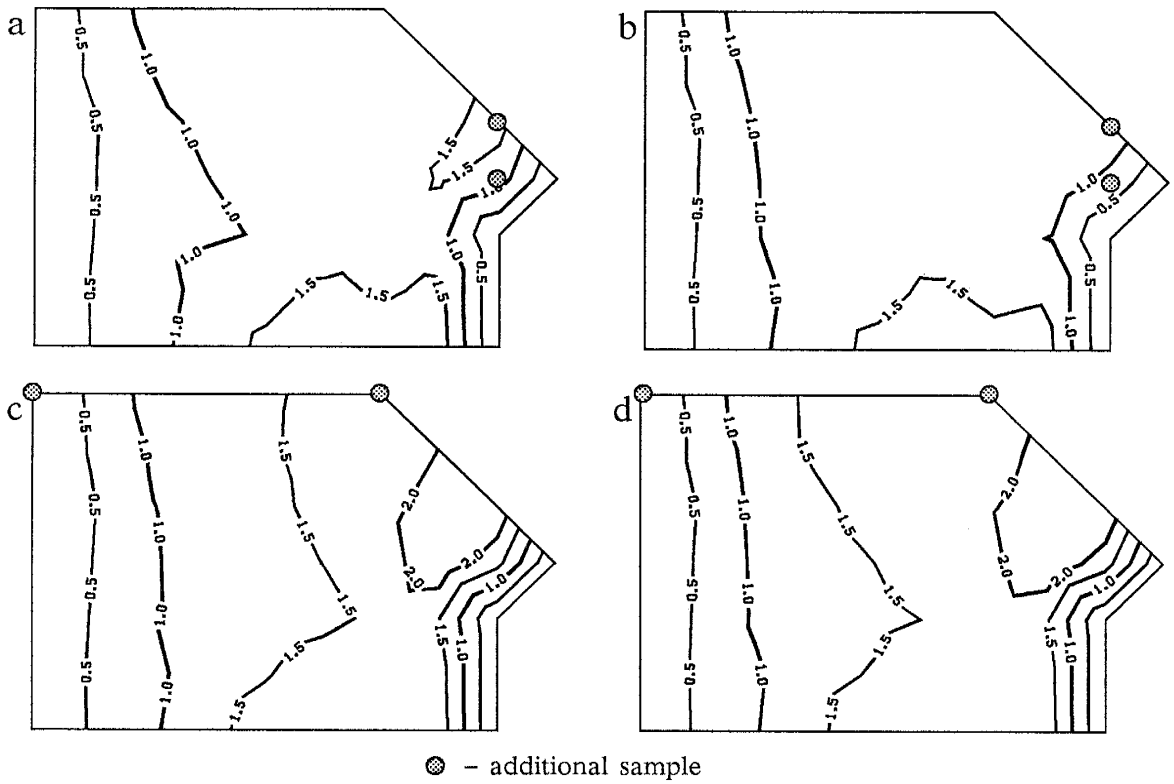

Figure 7. Head standard deviations after addition at two algorithm selected sample locations of values from a)the kriging results and b)the underlying transmissivity field. Head standard deviations after addition at the two maximum kriging variance locations of values from c)the kriging results and d)the underlying transmissivity field. Compare to Figure 3d.

As stated in the introduction, a standard approach to the network design approach for sampling transmissivity is placing additional measurements at the location of greatest uncertainty in the transmissivity field. The head standard deviations resulting from adding measurements at the two locations of maximum variance (Figure 3c) in the kriged transmissivity field are shown in Figures 7c and 7d. Again, the values added are either the results of kriging the original thirty transmissivity measurements or actual values from the underlying transmissivity field. Two points become evident upon considering these last two uncertainty maps. First, there is barely any difference between the head deviations when kriging results or actual measurements are added. The actual measurements from the underlying field are close to the kriging results at these two locations; therefore, the head derivatives do not change significantly. Second, there is greater reduction in head uncertainty when the sample locations are chosen by the trace sensitivity algorithm (see Figures 7a and 7b). This is not surprising since our objective function is directly related to that goal.

Sequential Design with Scaled Trace Sensitivity

The sampling results from the first application of the trace sensitivity algorithm shown in Figure 4 compare well with the optimum locations selected by perturbation trace reduction. However, the trace sensitivity algorithm fails to select the sensitive region in the northwestern portion of the flow domain within a reasonable number of runs (compare Figures 4e and 4f). The algorithm also fails in the second application when it reselects the same sampling location (e.g., Figure 6b). Although this problem of reselecting the same location did not occur with the first application when kriged measurements were added there is nothing to prevent it from happening, as discussed previously. Scaling of the trace sensitivity results is necessary to prevent resampling the same location. There are several methods of scaling which work. The first is scaling the results by the kriging covariance matrix, similar to a self-scaling quasi-Newton nonlinear search method [*Luenberger*, 1984]. The scaled trace sensitivity is calculated by multiplying the trace sensitivities by the covariance matrix and is shown above in (5). Other possible scaling methods include weighting vectors composed of either kriging variances or column sums of the kriging covariance matrix. We scale the trace sensitivity results by the entire covariance matrix in order to incorporate as much information as possible.

A sampling sequence using the scaled algorithm is shown in Figure 8. The sequence again begins with the original thirty measurements. The contour intervals are different for the maps in Figure 8 from those presented previously because scaling reduces the magnitude of the trace sensitivities. The results, however, can be compared with those previous since it is only the relative minimums and maximums in which we are interested. In comparison to the scaled trace sensitivity results, the optimal location found by perturbation trace reduction stays the same after adding kriging measurements at the indicated locations in Figure 8. At each step the perturbation trace reduction optimal location is the same as the location indicated in Figure 8c (or Figure 4b). It is not until the third run that the scaled trace sensitivity algorithm selects the optimum location determined by the perturbation trace reduction approach. The scaled trace sensitivity algorithm does select nearby nodes to the optimum location at both the initial and one additional measurement runs (see Figure 8b and 8c). Discretization errors may be a source of the problem which causes the scaled algorithm to slightly miss the optimum sample location. In additional runs (not presented) the scaled trace sensitivity algorithm fails to select the important zone in northwest region (see Figure 4f) within a reasonable number of runs. Again, the nonlinear relationship between transmissivity and head is a possible cause for this failure. Scaling the trace sensitivity results by the kriging

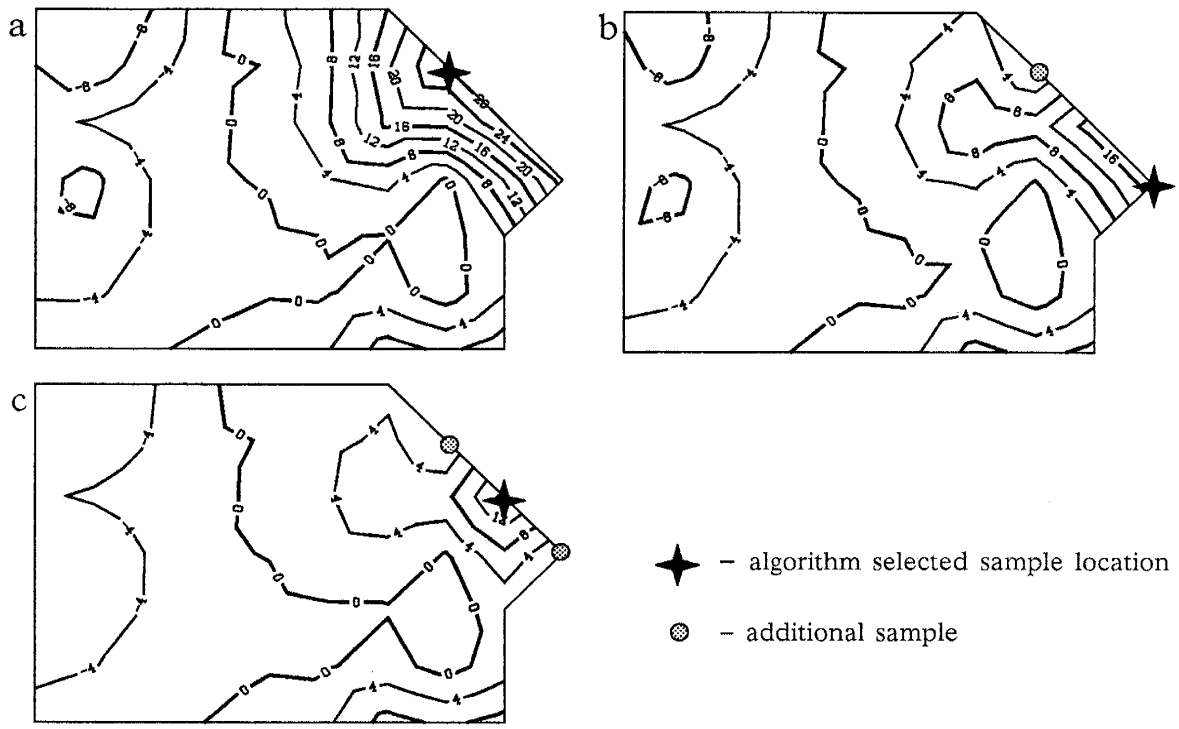

Figure 8. Sequential design with scaled trace sensitivity algorithm. Value added at each step is a result from kriging the initial thirty measurements. Scaled trace sensitivity results with a)original thirty measurements, b)after adding first value, and c) after adding second value. Compare to Figures 4 a,c,&e.

covariance matrix does, however, prevent the algorithm from reselecting the same location, whether using kriged data for measurements or actual measurements. In the next section we look at the effect of increasing the grid discretization in the two zones indicated as most sensitive in Figure 4a.

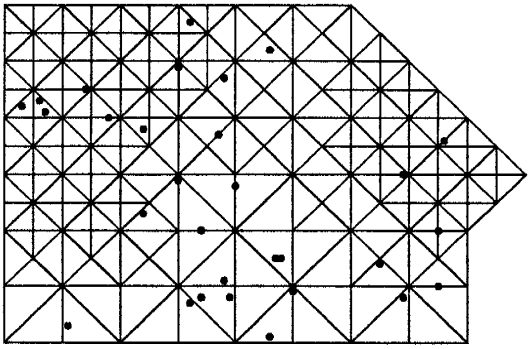

Figure 9. Finite element grid and measurement locations after increasing discretization in sensitive zones.

Increased Discretization

In an attempt to improve the performance of the trace sensitivity algorithm the two predominantly sensitive areas of the flow domain are discretized further to reduce truncation error and allow the estimated head surface to better conform to the actual head surface (see Figure 9). Since we choose to weight each node equally in the trace sensitivity analysis, increasing the grid density in a region results in increasing the weight or importance of that region. An alternative method would be to restrict the trace to include only those nodes common to the original finite element grid. Figure 10 depicts the kriged log transmissivity field based on the original thirty measurements (same estimated

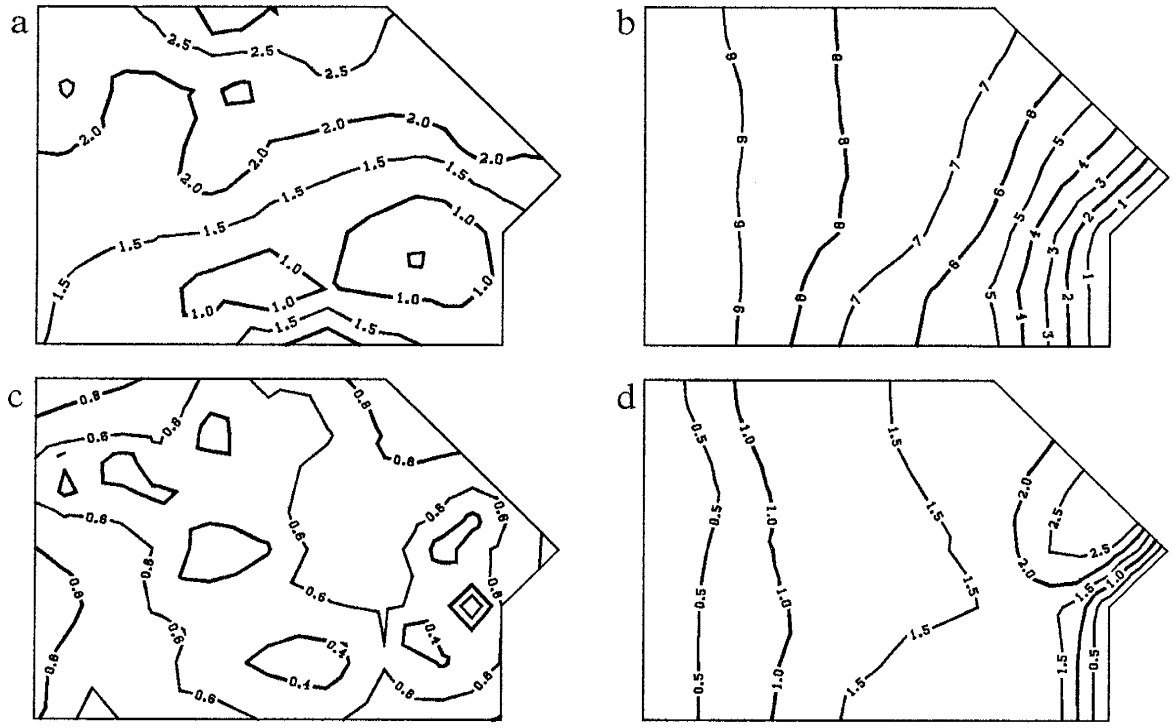

Figure 10. a) Kriged log transmissivity field, b) Head field, c) Log transmissivity standard deviations, and d) First order, second moment estimated head standard deviations.

variogram), the corresponding first order head field, and the related uncertainty associated with each when the increased discretization is used. As in the previous example, the contours are created with linear interpolation on the sides of the finite elements of the finer grid. The kriged log transmissivity field has the same overall character as that of Figure 3a but shows more detail due to the finer elements. The low log transmissivity zone is more pronounced with a further lowering of log transmissivity at the center of the southeastern region. The log transmissivity kriging standard deviations show this same increase in detail, especially where one of the measurement locations is coincident with a finite element node. While the first order estimated head field is virtually unchanged, the head standard deviations in the northeastern zone increase. Increased discretization should have the opposite effect, decreasing the head uncertainties in this region due to the neglect of variance reduction. Instead, the additional lowering of the log transmissivity in the southeastern region, combined with the decrease in kriging variance because of the coincident node and measurement location further focus flow through the northeast region. This results in an increase in head uncertainty in the northeast region which more than offsets the decrease in head uncertainty due to reduced discretization errors.

Trace sensitivity results, unscaled and scaled, for the increased discretization are shown in Figures 11a and 11b. Only results from applying the algorithm to the initial thirty measurements are shown. With the increased discretization, the trace sensitivity algorithm (scaled and unscaled) misses the optimum sampling location determined by perturbation trace reduction, which is calculated by two different methods in this example. The first method, Figure 11c, is the same as before. It uses the kriging results from the original thirty measurements as the value added at each node. The second method, Figure 11d, adds the true (no noise added) value from the underlying field at each nodal location. This was done in an attempt to more closely approximate the optimal measurement location for maximum trace reduction. It is possible to use this method in the hypothetical case only because we know the values in the underlying synthetic field.

The optimum measurement location differs slightly with the two perturbation trace reduction methods. Additional and, in this case, complete information from the underlying field affects the perturbation trace reduction results. The optimum sampling location in either method shifts closer to the prescribed head boundary than was seen in the case with less discretization (see Figure 4b). This follows the concept relating to one-dimensional flow discussed previously. Some negative perturbation trace reductions occur when the measurements are taken from the underlying field (see Figure 11d). This is caused by nonlinear changes in head derivatives due to changes in the estimated

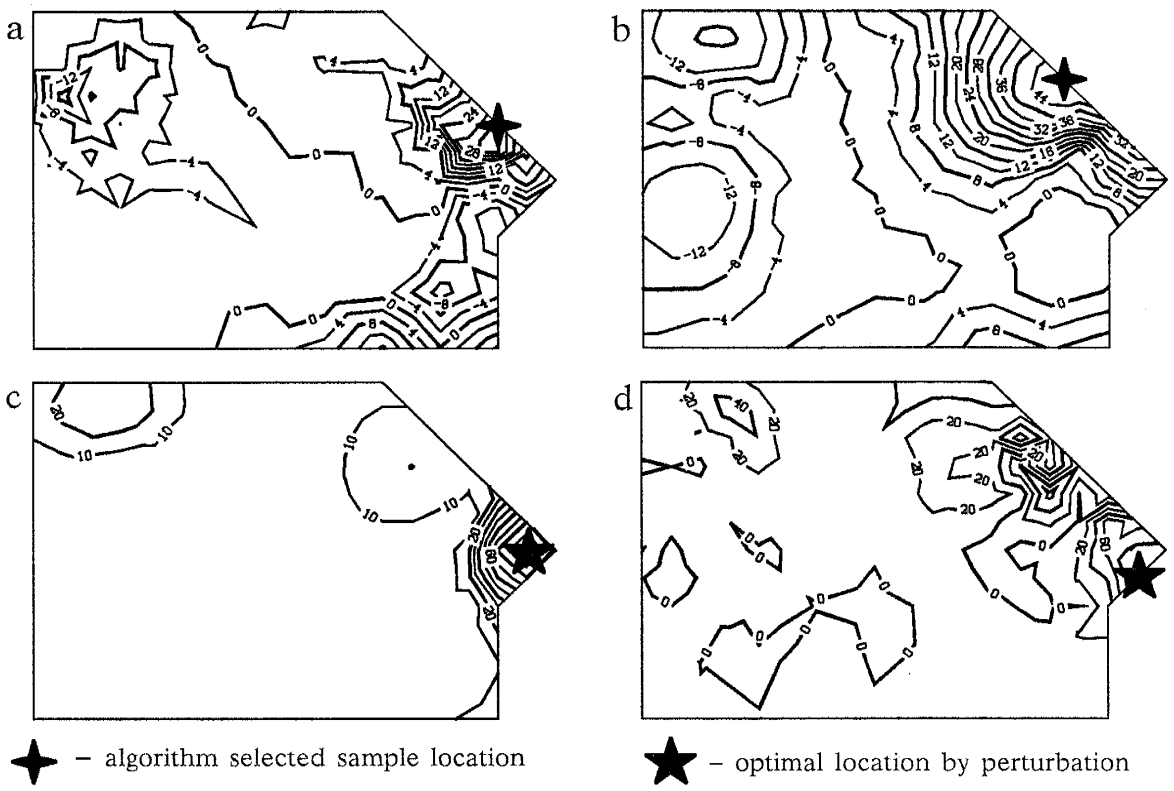

Figure 11. With the increased discretization; a)Trace sensitivity results, b)Scaled trace sensitivity results, c)Trace perturbation reductions with kriged values, and d) Trace perturbation reductions with values from underlying field. Compare to Figures 4a, 4b and 8a.

transmissivity field from additional measurements, and thereby increasing the resulting head variances.

The attempt to improve algorithm performance by increasing discretization within the sensitive zones does not produce desirable results. The trace sensitivity algorithm selected location in both the scaled and unscaled case is more than one node away from the optimal location as found by perturbation trace reduction. The unscaled algorithm selects the same location as with the original grid (Figures 4a and 11a). Scaling the trace sensitivity results by the covariance matrix moves the measurement selected location away from the existing measurements (Figures 8a and 11b). There is at least one possible explanation why increased discretization fails to improve the trace sensitivity algorithm performance. The increased importance due to more nodes in the sensitive zone may be affecting the results.

Even with the failure of the trace sensitivity algorithm to select the actual optimal sampling locations, the approach shows promise. Even though the trace sensitivity algorithm does not necessarily select the optimal location it does succeed in locating the sensitive zones of the flow domain for the given objective function of minimizing head uncertainty reduction.

Conclusions

A new approach using the sensitivity of the head covariance matrix trace to changes in transmissivity in order to design transmissivity measurement networks has been developed. While the usefulness of the approach per se is limited, it does point out the need for managers to be aware that the network design approach of locating additional measurements at locations of maximum uncertainty of the transmissivity field may not necessarily be the optimal location with a different objective function. In the hypothetical example presented the location of maximum kriging variance was in an entirely different zone of the flow domain from the measurement location that yielded the greatest reduction in head uncertainty.

The results of the trace sensitivity algorithm compare well with those from perturbation trace reduction. Even though the actual location selected often differs between the two methods, zones or regions important for trace reduction agree overall.

Nonlinearity of the relationship between head and transmissivity creates problems when using the algorithm, such as reselecting the same sample location after already sampling there. Scaling of trace sensitivities by kriging covariance matrix successfully prevents reselecting same location. Discretization errors may be a source of the problem which causes the scaled trace sensitivity algorithm to slightly miss the optimum sample location found by perturbation trace reduction.

The trace sensitivity approach may work best when used to guide a subjective search using the perturbation trace reduction method in order to find the optimal sampling location or when used as a guide to designate zones of a finite element mesh need to be discretized further. The algorithm can also be extended to include transient flow, uncertain boundaries, and head measurements. Additionally, the algorithm methodology could be used for other applications, for example, design of monitoring networks for pollutant concentrations or travel times.

Acknowledgments

This material is based upon work supported under a National Science Foundation Graduate Fellowship.

References Cited

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

Bogardi, I., Bardossy, A., and Duckstein, L., Multicriterion network design using geostatistics, *Water Resources Research, 21*(2), 199–208, 1985.

Delhomme, J.P., Kriging in the hydrosciences, *Advances in Water Resources, Vol. 1, No.* 5, 251–266, 1978.

Dettinger, M.D., and Wilson, J.L., Numerical modeling of aquifer systems under uncertainty: a second moment analysis, Technical note, Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics, Department of Civil Engineering, MIT, 1979.

Dettinger, M.D., and Wilson, J.L., First order analysis of uncertainty in numerical models of groundwater flow: Part 1. Mathematical development, *Water Resources Research, 17*(1), 149–161, 1981.

Gutjahr, A.L., and J.L. Wilson, Co-kriging and stochastic differential equations, *Proceedings of Symposium on The Stochastic Approach to Subsurface Flow*, Montvillargenne, France, published by Ecole des Mines, Paris, June, 1985.

Hughes, J.P., and Lettenmaier, D.P., Data requirements for kriging: estimation and network design, *Water Resources Research, 17*(6), 1641–1650, 1981.

Kitanidis, P.K., and R.W. Lane, Maximum likelihood parameter estimation of hydrologic spatial processes by the Gauss–Newton method, Iowa Institute of Hydraulic Research Technical Report No. 275, University of Iowa, May, 1984.

Luenberger, D.G., *Linear and Nonlinear Programming*, Second Edition, Addison–Wesley Publishing Company, Menlo Park, California, 1984.

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

Rouhani, S., Variance reduction analysis, *Water Resources Research, 21*(6), 837–846, 1985.

Townley, L.R., Numerical models of groundwater flow: prediction and parameter estimation in the presence of uncertainty, Ph.D. thesis, Department of Civil Engineering, MIT, 1983.

Townley, L.R., Second order effects of uncertain transmissivities on prediction of piezometric heads, Proceedings of 5th International Conference on Finite Elements in Water Resources, University of Vermont, June, 1984.

Townley, L.R., and J.L. Wilson, CERT documentation and user's manual, Open-file report at the New Mexico Institute of Mining and Technology, Hydrology Program. 1984.

Townley, L.R., and J.L. Wilson, Computationally efficient algorithms for parameter estimation and uncertainty propagation in numerical models of groundwater flow, *Water Resources Research, 21*(12), 1851-1860, 1985.

Vanmarke, E., *Random Fields Analysis and Synthesis,* MIT Press, 1983.

Warrick, A.W., and Myers, D.E., Optimization of sampling locations for variogram calculations, *Water Resources Research, 23*(3), 496-500, 1987.

Appendix A

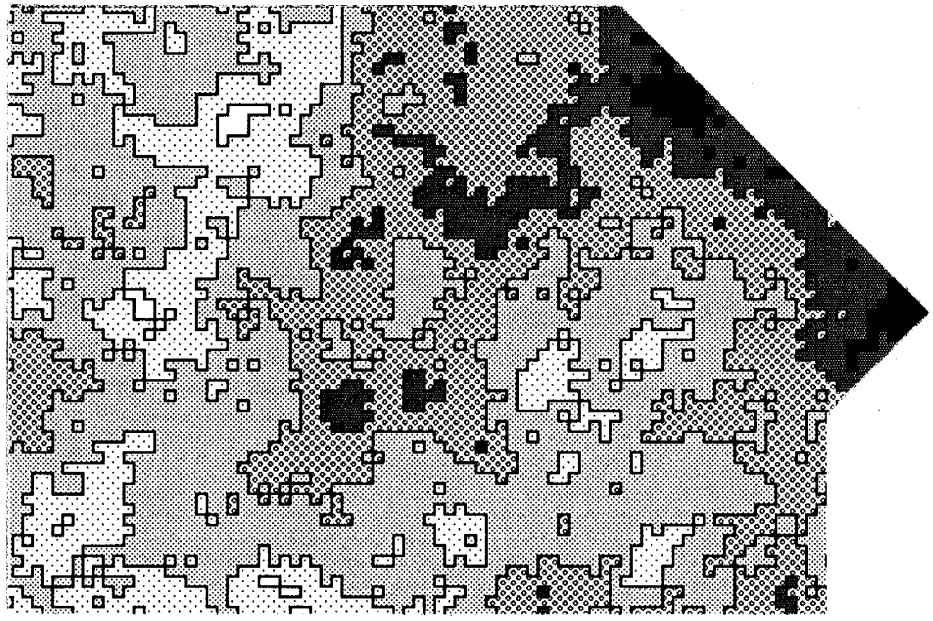

Figure A1. Exponential covariance synthetic field, correlation length = 40., variance = 1.0.

Additional Numerical Application: Increased Correlation Length

In order to further illustrate the application of the trace sensitivity algorithm, a new synthetic field is generated with the turning bands method. The new synthetic field is shown in Figure A1 in the same manner as before, each shading pattern represents log transmissivity values falling within the range of one sample standard deviation. This field had the same prescribed variance (1.0) as the field in the previous numerical application but a longer correlation length. The new correlation length is forty units, twice that of the previous example. This length is chosen so there are five nodes per correlation length, the currently accepted minimum discretization with which to model the variability in the underlying log transmissivity field. The variogram estimation step is left out of the kriging exercise for this case because variogram estimation is not the emphasis of this paper. Instead, the theoretical variogram parameters used for the turning bands method with the addition of a nugget of 0.1, are used for all kriging associated with the new synthetic field. The same thirty measurement locations as in the previous example are used with the same measurement noise N(0.,0.1) added to them. The results of the kriging exercise, the first order estimated heads, and the related uncertainties are shown in Figure A2. All results in Appendix A are calculated and contoured using the originally discretized finite element grid shown in Figure 1.

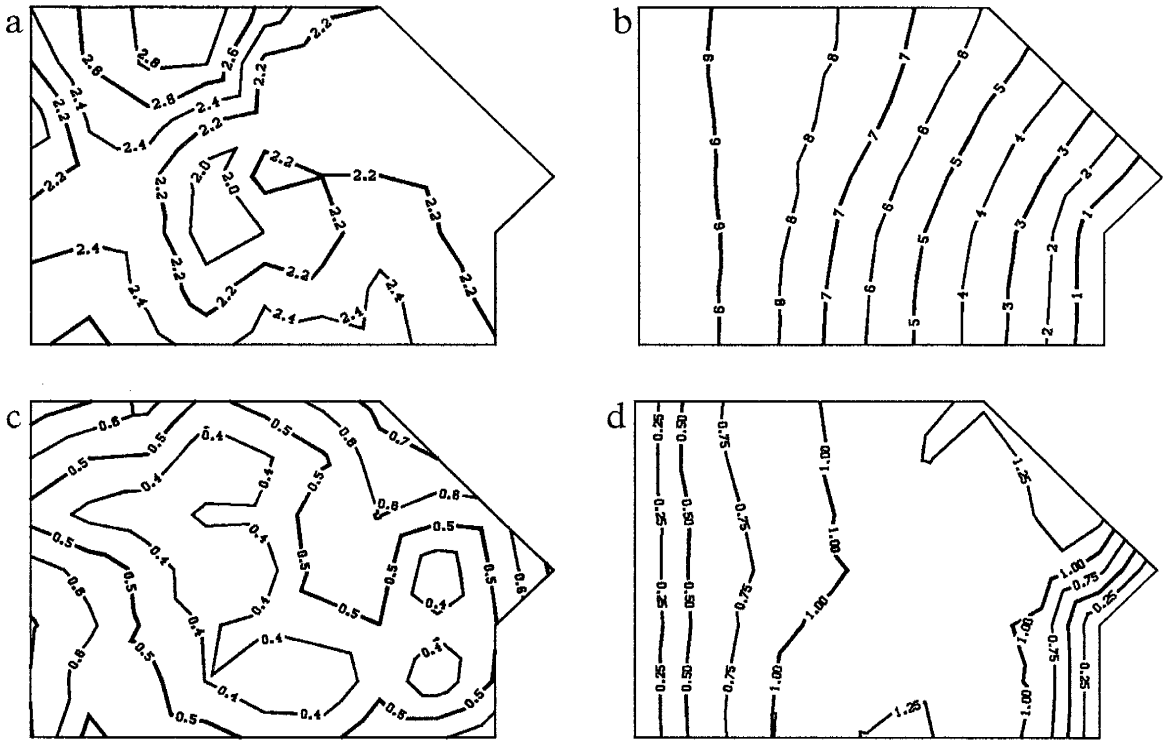

Figure A2. a) Kriged log transmissivity field, b) First order estimated head field, c) Log transmissivity standard deviations, and d) Head standard deviations.

The new kriged field has a low log transmissivity zone almost centered in the flow domain. The flow rate is focused above and below this low transmissivity zone which favors creation of zones sensitive for trace reduction north and south of the low zone, similar to the previous numerical application. We also expect sensitive areas to occur near each boundary to a greater extent in this example because the transmissivity field is more uniform and hence more like a one-dimensional flow problem.

The trace sensitivity algorithm results and perturbation trace reduction results for the initial thirty measurements are shown in Figure A3. As expected, the indicated sampling location(s) are close to the prescribed head boundaries with the optimum location(s) indicated in Figures A3e and A3f. The effects of three different scaling methods [*Magnuson and Wilson,* 1988] are shown in Figures A3b, A3c, and A3d. When comparisons are drawn between these contour plots it is the location of the peaks and not the units that are important. Scaling by the entire kriging covariance matrix produces the smoothest plot and shows the most influence of the original measurement locations (Figure A3b). Scaling by sums of columns from the covariance matrix uses less information from the matrix and creates a slightly rougher plot (Figure A3c). Accordingly, scaling by the kriging variances uses the least information from the

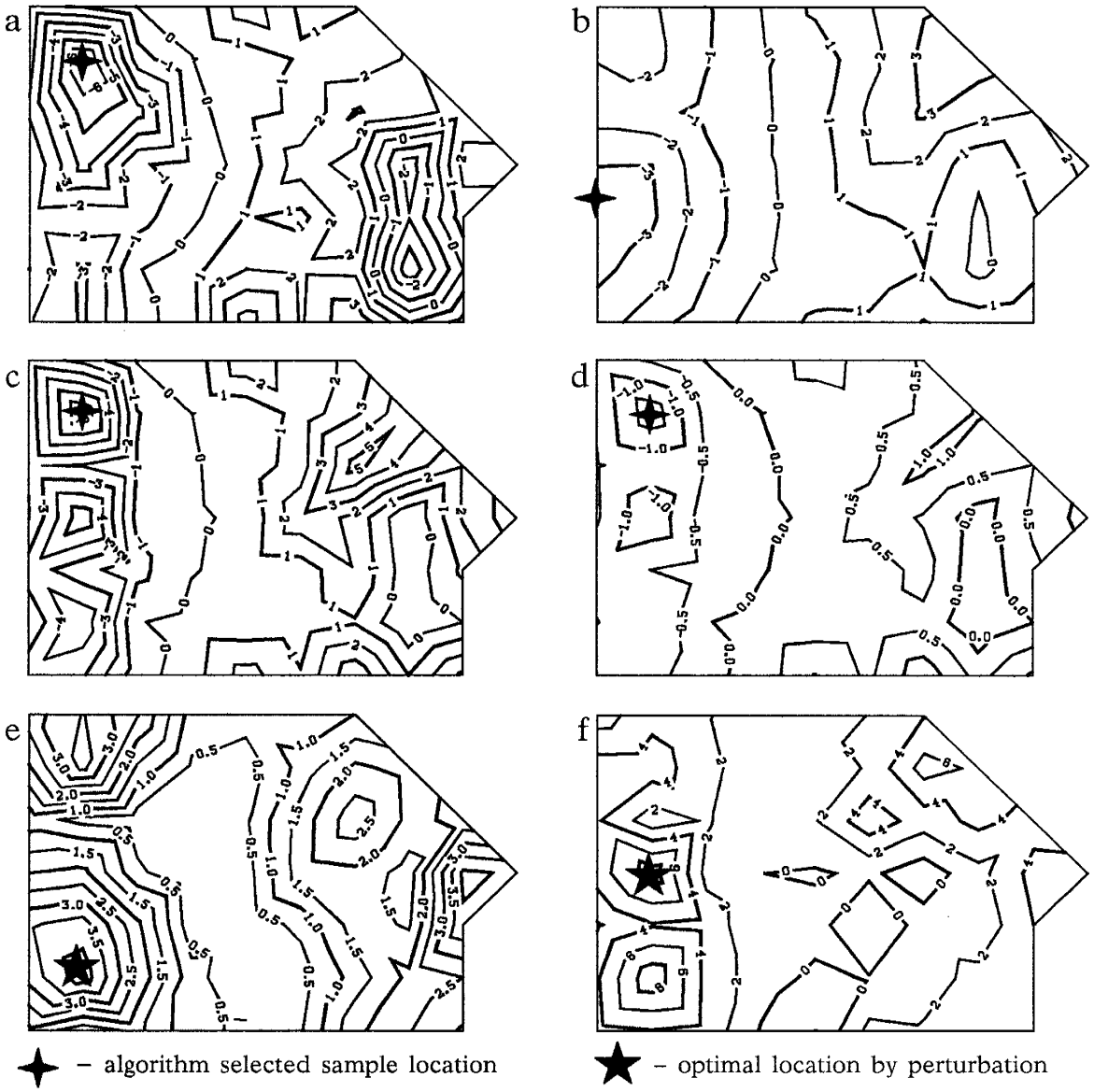

Figure A3.Trace sensitivity results on the original thirty measurements: a) Unscaled, b) Scaled by full covariance matrix, c) Scaled by column sums of covariance matrix, d) Scaled by kriging variances. Perturbation trace reductions with e) kriging results, and f) true values from underlying transmissivity field.

covariance matrix, creates the roughest plot, and shows the least impact from the measurement locations (Figure A3d). Each method does, however, prevent resampling the same location which was the original intent of scaling

As in the previous example with a different underlying field, the trace sensitivity algorithm results succeed in locating zones of higher trace sensitivity but fail to select the particular node that would yield the greatest trace reduction (see Figure A3e). Actually, the location selected by the full matrix scaled trace sensitivity algorithm is not to far from

that found by perturbation trace reduction with the kriging results (compare Figures A3b and A3e). Likely causes of the failure to select the optimal measurement location are again either discretization error or nonlinearity of the head/transmissivity relationship.

Due to the location of the kriging measurements, the algorithm is unable to pick out the true global optimum sampling location shown by trace perturbation with measurements taken from the underlying field (see Figure A3f). None of the original thirty measurements pick up a local high transmissivity zone near the center of the western prescribed head boundary (see Figure A1), which when sampled results in the global maximum in the perturbation trace reduction map. Two of the alternative scaling methods each show a local negative peak which corresponds with this true global optimum (see Figures A3b and A3d). Since the kriging result does not predict the correct transmissivity value in this region, it is likely coincidental that scaling the trace sensitivity results creates a local negative peak in this region. Possible explanations for this weigh heavily on the previous discussion of sensitive zones near prescribed head boundaries in one-dimensional flow fields.

One strong argument for the use of the scaled rather than the unscaled trace sensitivity algorithm is the local negative peaks that occur near the eastern prescribed head boundary in the unscaled trace sensitivity results (see Figure A3a). These peaks likely occur for the previously discussed reason relating to one-dimensional flow fields. But in either of the perturbation trace reduction contour plots there are no corresponding local peaks for this same region. There are numerous measurement locations in the eastern region in the original set of thirty so it is not surprising that a measurement here has no great affect on reducing head uncertainty. By comparison, all three of scaled trace sensitivity algorithms show a low sensitivity to measurements in this location--more in agreement with what intuition would say for this region due to the presence of existing measurements.

Addition of Value at Measurement Location

The indicated sample location in Figure A3a is selected as the sample point in order to illustrate what happens to the unscaled trace sensitivities when a location at a negative peak is used after being selected as the next measurement location. This same location is selected by two out of the three scaled trace sensitivity results. Trace sensitivity results with the addition of one kriged value as a measurement and the corresponding perturbation trace reductions are shown in Figure A4. In general when kriged values are added as samples at negative peaks and no scaling is used, the subsequent trace sensitivity results at the additional measurement location become more negative. This behavior of reselecting the same location occurs at positive peaks also (see Figure 6).

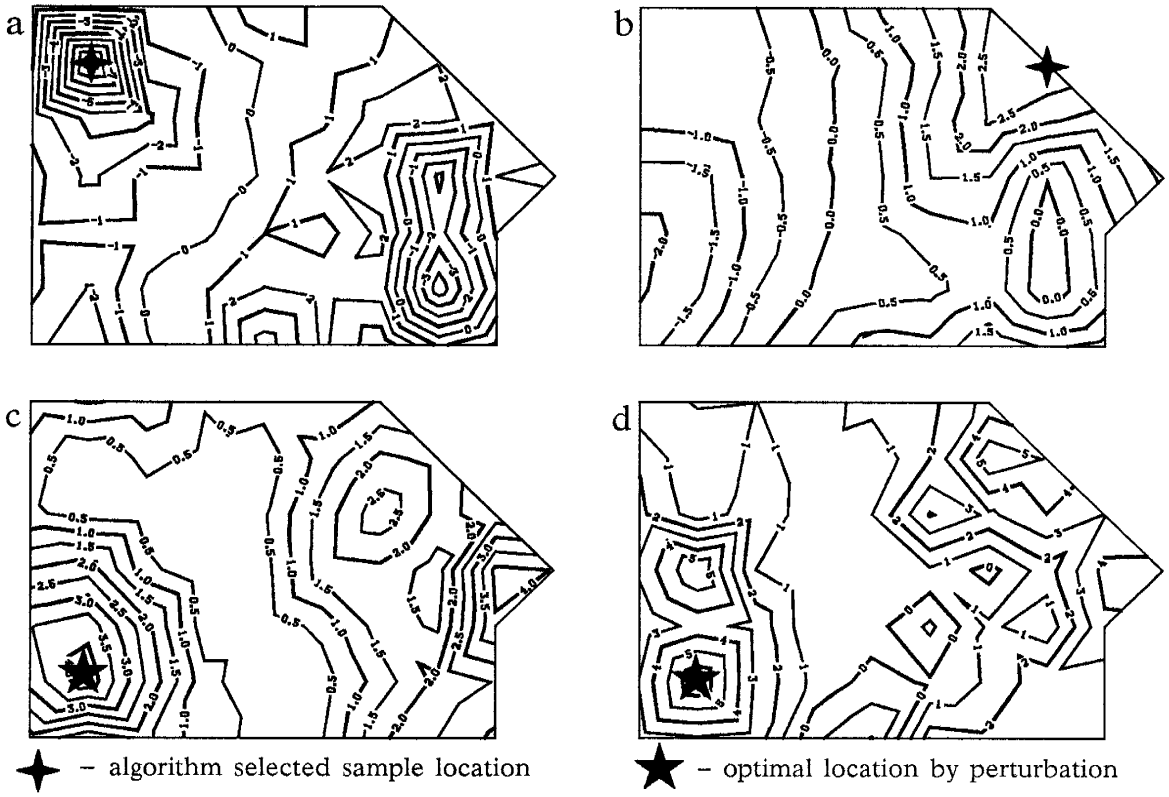

Figure A4. After one sample: a) Trace sensitivity results, b) Scaled trace sensitivity using full covariance matrix, c) Trace perturbation reductions with kriged values, and d) Trace perturbation reductions with values from the underlying field.

However, the trace sensitivity results when positive peaks are selected at least trends towards zero (see Table 1), while at the negative peaks the absolute value of trace sensitivity increases (compare Figures 3a and 4a). The possibility of this behavior is discussed previously [*Magnuson and Wilson,* 1988]. Coding errors in the algorithm have been most rigorously removed and are not a likely explanation for this behavior. In the extreme it is obvious from equation (4) that adding enough data points at nodes in the negative trace sensitivity regions eventually will force the trace sensitivity in the area to zero. Although not shown, this was done and the proper behavior was observed.

The full matrix scaled trace sensitivity results select a positive peak on the northeast side of the domain (Figure A4b), showing the effect of the existing kriging measurement locations. The overall pattern of results is much more like those of the trace perturbation reductions (Figure A4c) than those of the unscaled trace sensitivity (Figure A4a), although the selected optimal locations are different. The perturbation trace reductions by either the kriging values or measurements from the underlying field point to the same location in this situation. The kriging perturbation trace reductions miss a local optimum

near the center of the western prescribed head boundary that does appear in the underlying field trace perturbation reductions (compare Figures A4c and A4d).

In summary, this application of the trace sensitivity algorithm yields the same results as the previous application discussed in *Magnuson and Wilson,* 1988. The algorithm fails to select the optimum location identified by the trace perturbation approach but it does succeed in locating the general areas which are more sensitive to trace reduction. Increasing the correlation length of the underlying transmissivity field does not significantly effect the algorithm results. It is interesting to note that none of the selected sample locations are in the zones of focused flow due to the low log transmissivity zone in the center of the domain. The increased correlation length makes the field more uniform and therefore, more like a one-dimensional flow problem. Scaling the trace sensitivity results is shown to be even more important than in the example presented in *Magnuson and Wilson,* 1988. Spurious local trace sensitivity peaks which appear near prescribed head boundaries are eliminated by scaling.

References Cited

Magnuson, S.O., and Wilson, J.L., Optimization of transmissivity measurement networks for head uncertainty reduction, submitted to *Water Resources Research,* 1988.

Appendix B

Trace Perturbation Comparison of True Values with Noisy Values

The use of perturbation trace reductions with values from the underlying field has been used in both *Magnuson and Wilson*, 1988 and in Appendix A as an approximation to the truth in order to find *the* optimal sampling location. In realistic terms a measurement taken at a point will have measurement noise associated with it. Figure B1 shows six different perturbation trace reduction results based on the original thirty measurements

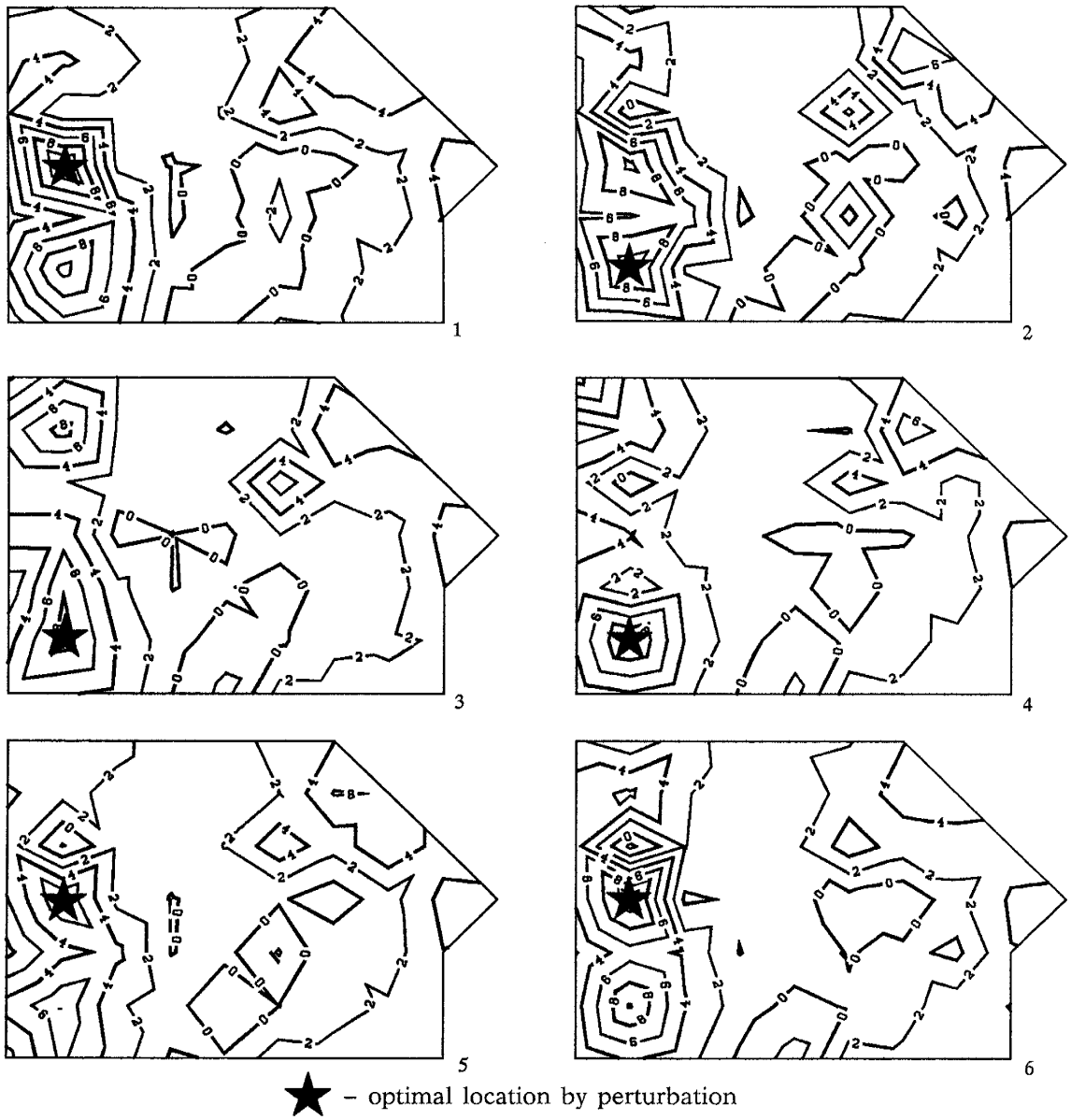

Figure B1. Six perturbation trace reductions with measurement noise N(0.,0.1) added to the true value from the underlying field. Compare to Figure A3f.

(from the Appendix A underlying field) with white noise, N(0,0.1), added to the values from the underlying field. By comparing the contour plots in Figure B1 to Figure A3f, it can be seen that perturbation trace reduction with true or non–noisy samples from the underlying field (Figure A3f) represents an ensemble of perturbation trace reductions with noisy values (Figure B1); therefore, the perturbation trace reductions with true values from the underlying field used in *Magnuson and Wilson,* 1988 and Appendix A paper can be used to approximate the truth in determining optimal sampling locations for maximizing the reduction of the head covariance matrix trace.

References Cited

Magnuson, S.O., and Wilson, J.L., Optimization of transmissivity measurement networks for head uncertainty reduction, submitted to *Water Resources Research,* 1988.

Appendix C

Recommendations for Further Research

The field of optimization related to transmissivity and head measurement networks has received much attention in the past and continues to receive attention. One reason for the attention is the practical application of results from research in this area. Because of the practical applications, continued effort is needed in order to determine more applicable objective functions for the decision makers who apply the research results. Selecting sample locations in order to reduce the kriging variance of the transmissivity field *does not* meet the needs of groundwater managers. The objective function presented in this paper is only a first step in the right direction. Reducing head uncertainty is a proper way to proceed but the objective function chosen in this paper, the trace of the head covariance matrix, is not likely to be a usable objective function. Perhaps narrowing the region of concern for the head uncertainty by implementing a weighting scheme (as discussed in *Magnuson and Wilson,* 1988) would produce usable results. For example, selecting a zone out of the entire flow domain and placing transmissivity measurements in order to reduce the sum of head variances within that zone.

A different approach to the problem which uses some of the same ideas may be more applicable. Use of the inverse problem to incorporate more information into the design process would be more realistic since head information is neglected totally in this paper. Co-kriging of head and transmissivity with an inverse solution could be a promising alternative and could be used as an extension of the concept relating to minimizing head uncertainty presented herein. From a manager's standpoint, an objective function which determines transmissivity and head measurement locations in order to minimize head uncertainty throughout a domain, or reduce it below some prescribed uncertainty level and which incorporates spatial information regarding head and transmissivity would be a valuable contribution and deserves further research effort.

Appendix D

Trace Sensitivity Code Listings

Two FORTRAN77 codes are included which when used in combination with CERT will calculate trace sensitivities, both unscaled and scaled. The first code, KINV.F is "crude" in that it takes unformatted output relating to the compactly stored coefficient matrix and solves for the inverse matrix using an IMSL routine. The inverse matrix is dumped in unformatted form for use in the second code, TRACE.F. This code uses the inverted matrix along with additional unformatted output from CERT to calculate the second derivatives of head w.r.t. log transmissivity. Then using equation (4) from *Magnuson and Wilson,* 1988, the trace sensitivities are calculated.

In addition to these two codes, a dimensioning file, DIMENS.INC is listed. All the dimensions for both the FORTRAN77 codes are contained in DIMENS.INC. This same file is used for dimensioning CERT. For information on redimensioning CERT, and thus KINV.F and TRACE.F, the reader is reffered to the CERT User's Manual.

KINV.F Listing

```
c*****************************************************************************
c
c           code to take gk dumped from doubt,put in into full upper triangular
c           storage, use IMSL to get an inverse matrix ,gkfinv, and lastly build
c           a 2dsquare matrix gkibig which is filled from gkfinv.  the big matrix
c           is dumped in binary form for use in the John Wilson's network design
c           algorithm. the file name is gkinv.out
c
c           written by : swen magnuson   8/87
c
c*****************************************************************************
c
      include 'DIMENS.INC'
           dimension gkf(j09op),gkfinv(j09op),gkibig(j02g,j02g),wk(14703)
           character*4 dum,bgk,sinv,symf,dmp,tot
           data bgk/' GKF'/,sinv/'INVT'/,symf/'SFUL'/,dmp/'DUMP'/,
     *     tot/'TOTL'/
           open(93,file='run.time')
           iout=93
           ktime=0
           call elapse(ktime,0,dum)
           ktot=ktime
c
c          get number of node points interactively
           write(6,*)' enter number of node points for this problem'
```

```
            read(5,*)numnp
c
c           calculate the number of terms in full symmetric storage
            ngk = nupper(numnp,numnp)
            if(ngk.gt.j09op) then
              write(6,*)' error, numnp is too large'
              stop
            endif
            call bldgkf(gkf,ngk,numnp)
            call elapse(ktime,iout,bgk)
c
c           calculate required size of work array for IMSL inverting routine
            nwk=(numnp*(numnp+1))/2+2*numnp
            if(nwk.gt.14703) then
             write(6,*)' error in MAIN, wk is dimensioned too small'
             stop
            endif
            call syminv(gkf,ngk,numnp,nwk,wk,gkfinv)
            call elapse(ktime,iout,sinv)
c
            call symful(gkfinv,ngk,numnp,gkibig)
            call elapse(ktime,iout,symf)
c
c           last thing to do is dump the gkibig matrix to a binary file
            open(95,file='gkinv.out',status='new',form='unformatted')
              write(95)numnp,gkibig
            close(95)
            call elapse(ktime,iout,dmp)
c
            ktime=ktot
            call elapse(ktot,iout,tot)
c
            stop
            end
c
c*******************************************************************************
c
            subroutine bldgkf(gkf,ngk,numnp)
c
c           routine that builds full upper triangular gkf matrix from profile
c           compacted gk matrix which is dumped from doubt
c
C           parameter (j12pa=2276, j02g=99,j02on=615,j06pa=160,j01g=2000,
C     *           j25ps=9801,j09op=4950)
            include 'DIMENS.INC'
            dimension dtk(j12pa),kdiag(j02g),num(j02g),loct1(j02on),
     *          loct2(j02on),loctpt(j06pa),lptdtk(j06pa),gk(j01g),
     *          dhdt(j25ps),ptt(j09op)
            dimension gkf(ngk)
```

```
      open(95,file='params.out',status='old',form='unformatted',
     *     access='readonly')
      read(95)nk
      read(95)ndtk
      read(95)jj12pa,dtk
      if(jj12pa.ne.j12pa) goto 999
      read(95)jj02g,kdiag
      if(jj02g.ne.j02g) goto 999
      close(95)

      open(95,file='senpro.out',status='old',form='unformatted',
     *     access='readonly')
      read(95) jj01g,gk
      if(jj01g.ne.j01g) goto 999
      close(95)

      k=1
      gkf(k) = gk(k)
c
c     this loop used number of terms in column i (ntci) to place terms
c     sequentially from gk into proper position in gkf. the ifs variable
c     is the index "full" symmetric matrix.  actually it's full upper-
c     triangular.
      do 20 i=2,numnp
        ntci = kdiag(i) - kdiag(i-1)
        do 10 j=1,ntci
          k=k+1
          ir = i-ntci+j
          ifs = nupper(ir,i)
          gkf(ifs) = gk(k)
   10   continue
   20 continue

      return
  999 write(6,*)' error, one of dimensions didn"t match parameter'
      stop
 1000 format(i10,1p2e10.2)
      end
c
C*********************************************************************
      SUBROUTINE ELAPSE(KT1,IOUT,SECT)
C*********************************************************************
C
C     PROGRAM TO FIND ELAPSED TIME IN CPU SECONDS BETWEEN SUCCESSIVE
C      CALLS OF THIS ROUTINE
C
C     FOR UNIX Y MUST BE DIMENSIONED TO 2 SPACES
C
      CHARACTER*4 SECT
      DIMENSION Y(2)
C
```

```
      KT2=1000.0*ETIME(Y)
      T=KT2-KT1
      KT1=KT2
      IF(IOUT.EQ.0) RETURN
      TIME=T/1000.0
      WRITE(IOUT,2000) SECT,TIME
      write(6,2000)SECT,TIME
      RETURN
 2000 FORMAT(1X,A4,F10.3,' SECS')
      END
c
c*******************************************************************************
c
            integer function nupper(i,j)
c
c           function to find location of the (i,j)th element of a matrix stored in
c           upper triangular form. to use for lower triangular storage just reverse
c           the arguments in the calling statement
            nupper = (((j-1)*j)/2)+i
            return
            end
c
c*******************************************************************************
c
            subroutine symful(asym,nasym,n,aful)
c
c           routine to create a full 2d matrix from a symmetricly stored 1d matrix
c
c           n is the order of the matrices. remember upper of lower storage modes
c           are the same, hence the reversing of subscripts in the 10 loop.
c
       include 'DIMENS.INC'
            dimension asym(nasym),aful(j02g,j02g)

            k=0
            do 20 i=1,n
             do 10 j=1,i
               k=k+1
               aful(i,j)=asym(k)
               aful(j,i)=asym(k)
   10         continue
   20    continue
            if(k.gt.nasym) goto 999
            return
999         write(6,*)' error in SYMFUL, asym dimensioned too small'
            stop
            end
c
c*******************************************************************************
c
```

```
      subroutine syminv(gkf,ngk,numnp,nwk,wk,gkfinv)
c
c     gkf is allready in full symmetric storage mode, lower or upper.
c     this routine uses IMSL library routne LINV2P to calculate the
c     symmetric inverse matrix
      dimension gkf(ngk),gkfinv(ngk),wk(nwk)
c
      call linv2p(gkf,numnp,gkfinv,idgt,d1,d2,wk,ier)
c
      if(ier.ne.0) then
        write(6,*)' error in IMSL routine,  ier = ',ier
        stop
      endif
      return
      end
c
c*******************************************************************************
```

TRACE.F Listing

```
c*****************************************************************************
c
c          code to run posterior head covariance trace sensitivity algorithm.
c          requires second derivative matrix of h by t twice, i.e. Dt't'h.
c
c          written by swen magnuson   9/87
c
c          code is computer intensive and care should be taken in execution.
c          input files are generated by CERT and kinv.f.  these files are binary.
c          the output from this code is a file dtth.out which is also binary.
c
c          see subroutine INPUT for explanation of dimensioning parameters
c
c*****************************************************************************
c
           include 'DIMENS.INC'
           common/stuff/dtk(j12pa),kdiag(j02g),num(j02g),loct1(j02on),
     *                loct2(j02on),loctpt(j06pa),lptdtk(j06pa),gk(j01g),
     *                dhdt(j25ps),ptt(j09op),dtksub(j02g,j02g),hk(j02g),
     *                tt(j06pa)
           common/nodes/numnp,logt
           dimension prod(j02g*j09op),gkinv(j02g,j02g),
     *          gkirws(j02g*j02g),dtth(j02g*j09op),dttr(j02g),
     *          ptt2d(j02g,j02g),pttcols(j02g*j02g),dttrs(j02g)

           character*4 dum,inp,ipr,gki,rmc,dmp,qua,tot
           data inp/'INPT'/,ipr/'IPRD'/,gki/'KINV'/,rmc/'MULT'/,
     *          dmp/'DUMP'/,qua/'QUAD'/,tot/'TOTL'/

           open(93,file='run.time')
           iout=93
           ktime=0
           call elapse(ktime,0,dum)
           ktot=ktime

c          decide if dtth matrix was built in a previous run
           write(6,*)' build new dtth matrix ?'
           write(6,*)'      1 - yes'
           write(6,*)'      2 - no'
           read(5,*)ind

c          get CERT output
       call input(ind)
           call elapse(ktime,iout,inp)

           if(ind.eq.2) goto 10

c          build intermediate product
           call iprod(prod)
           call elapse(ktime,iout,ipr)
```

```
c           get gkinv matrix
            call getgki(gkinv)
            call fulrws(gkinv,numnp,gkirws)
            call elapse(ktime,iout,gki)

c           multiply gkirws time prod to get necessary columns of dtth
            call rmuc(gkirws,prod,numnp,dtth)
            call elapse(ktime,iout,rmc)

c           dump the result into a binary file for future use
   10   continue
c   10      call dumpd(dtth,numnp,ind)
            call elapse(ktime,iout,dmp)

c           call the quadruple sum routine which calculates trace sensitivity
            call quad(dtth,dhdt,ptt,dttr)
            call elapse(ktime,iout,qua)
            ione=1
            call dumptr(dttr,numnp,ione)

c           scale the trace sensitivity matrix by the kriging covariance matrix,
c           put ptt matrix into full form stored by columns
            nup = nupper(numnp,numnp)
            call symful(ptt,nup,numnp,ptt2d)
            call fulrws(ptt2d,numnp,pttcols)

c           multiply vector by matrix and put answer in dJdus

            call vecmat(dttr,pttcols,numnp,dttrs)
            itwo=2
            call dumptr(dttrs,numnp,itwo)
c

            ktime=ktot
            call elapse(ktot,iout,tot)
c
            stop
            end
c
c*******************************************************************************
c
            subroutine clearm(a,n)
c
c           routine to clear a square matrix a(nxn)
c
            include 'DIMENS.INC'
            dimension a(j02g,j02g)
            do 20 j=1,n
             do 20 i=1,n
               a(i,j) = 0.
   20       continue
            return
```

```
            end
c
c*******************************************************************************
c
      FUNCTION DOT(A,B,N)
C
C     PROGRAM TO PERFORM THE DOT PRODUCT OF TWO VECTORS
C
C ... IMPLICIT REAL*8(A-H,O-Z)
C ... REMOVE ABOVE CARD FOR SINGLE PRECISION COMPUTATIONS
C
      DIMENSION A(1),B(1)
C
      DOT=0.E0
      DO 10 I=1,N
      DOT=DOT+A(I)*B(I)
   10 CONTINUE
C
      RETURN
      END
c
c*******************************************************************************
c
            subroutine dumpd(dtth,numnp,ind)
c
c           routine to dump dtth matrix to binary file for future use
c
            dimension dtth(numnp*nupper(numnp,numnp))
            open(95,file='dtth.out',status='unknown',form='unformatted')
            if(ind.eq.1)write(95) numnp,dtth
            if(ind.eq.2)read(95) numnp,dtth
            close(95)
            return
            end
c
c*******************************************************************************
c
            subroutine dumptr(dttr,numnp,ind)
c
c           routine to dump the derivitive of the trace of the head covariance
c           matrix.
            character aname*13,aname2*13
            dimension dttr(numnp)
            data aname/'VALUES OF DTR'/
            data aname2/'VALUES OF SCA'/
            if(ind.eq.1) then
              open(94,file='dttr.out',status='unknown')
              write(94,2000)aname,dttr
              close(94)
            endif
```

```
            if(ind.eq.2) then
              open(94,file='scaled.out',status='unknown')
              write(94,2000)aname2,dttr
              close(94)
            endif
            return
 2000       format(/a13/(1x,1p10e12.3))
            end
c
C*************************************************************************
      SUBROUTINE ELAPSE(KT1,IOUT,SECT)
C*************************************************************************
C
C     PROGRAM TO FIND ELAPSED TIME IN CPU SECONDS BETWEEN SUCCESSIVE
C      CALLS OF THIS ROUTINE
C
C     FOR UNIX Y MUST BE DIMENSIONED TO 2 SPACES
C
      CHARACTER*4 SECT
      DIMENSION Y(2)
C
      KT2=1000.0*ETIME(Y)
      T=KT2-KT1
      KT1=KT2
      IF(IOUT.EQ.0) RETURN
      TIME=T/1000.0
      WRITE(IOUT,2000) SECT,TIME
      write(6,2000)SECT,TIME
      RETURN
 2000 FORMAT(1X,A4,F10.3,' SECS')
      END
c
c*************************************************************************
c
            subroutine epro(subrws,hk,npt,prod)
c
c           routine to do the additional calculation necessary when we use
c           log T instead of T.  The second derivative of the conductivity matrix
c           is no longer zero. On the first call to this routine, itime is zero and
c           icol, the column indicator, is set to 1. The increment index, inc, is
c           initialized to one more than is necessary since is is decremented
c           lower down in the routine.  This could be a little confusing since
c           the inc calculated in each call isn't added on until the next call to
c           the routine.
c
c           the following calucation is done:
c
c               -(Dy'y'K)(Iy * Iy * h)
c
c           The necessary dot products of rows and columns are done by passing
```

```
c          pointers to the function DOT. On the first call to this routine
c          the array hk is scaled by 2.303 rather than multiplying each result
c          by 2.303 after each call to DOT.
c
c          Remember that we are building only the necessary columns of prod
c          which will correspond to the necessary columns of dtth, hence the
c          icol indicator.
c
           dimension subrws(npt*npt),hk(npt),prod(npt*nupper(npt,npt))

c          calculate column of prod referenced by this call, on first time through
c          also scale the hk array by 2.303 in order to avoid duplicating
c          multiplications on subsequent calls.
           if(itime.eq.0) then
             icol = 1
             inc = npt + 1
             rln10 = alog(10.)
             call scale(hk,rln10,npt)
           endif
           if(itime.ne.0) icol = icol + inc
           inc = inc - 1
           itime = 1

c          calculate starting position in proper column of prod using icol
           ind = (icol-1)*npt + 1

c          multiply the rows of subrws by scaled hk respectively
           do 10 i=1,npt*npt,npt
             prod(ind) = prod(ind) + dot(subrws(i),hk,npt)
             ind = ind + 1
    10     continue
           return
           end
c
c*******************************************************************************
c
           subroutine filsub(l1,l2)
c
c          routine to fill submatrix dtksub from dtk using indices l1 and l2
c          which are indices to positions in loct1 and thereby refer to
c          row/column positions
c
c          the constant multiplying each entry of dtksub stems from the
c          expression listed in subroutine IPROD.
c
           include 'DIMENS.INC'
           common/stuff/dtk(j12pa),kdiag(j02g),num(j02g),loct1(j02on),
     *          loct2(j02on),loctpt(j06pa),lptdtk(j06pa),gk(j01g),
     *          dhdt(j25ps),ptt(j09op),dtksub(j02g,j02g),hk(j02g),
```

```
     *              tt(j06pa)
            common/nodes/numnp,logt

            if(logt.eq.0) con=-1.
            if(logt.eq.2) then
              nt = nt + 1
              con=-alog(10.) * tt(nt)
            endif

            do 20 i=l1,l2
              ic=loct1(i)
              do 10 j=l1,i
                k=k+1
                dtkk=dtk(k)*con
                ir=loct1(j)
                dtksub(ir,ic) =dtkk
                dtksub(ic,ir) =dtkk
   10         continue
   20       continue
            return
            end
C
c*************************************************************************
c
            subroutine filvec(a,istart,n,vec)
c
c           routine to fill vec with n values of a beginning with the istart
c           position in a.
            dimension a(n*n),vec(n)
            j=istart-1
            do 10 i=1,n
             j = j + 1
             vec(i) = a(j)
   10       continue
            return
            end
c
c*************************************************************************
c
            subroutine fulrws(a2d,n,a1d)
c
c           Routine to store a square 2d array by rows. Each row will be n
c           in length
c
            include 'DIMENS.INC'
            dimension a2d(j02g,j02g),a1d(n*n)
            kk = 0
            do 20 i=1,n
             do 10 j=1,n
          kk = kk + 1
```

```
            a1d(kk)=a2d(i,j)
 10        continue
 20       continue
          return
          end
c
c*************************************************************************
c
          subroutine getgki(gkinv)
c
c         routine to read gk inverse from binary file dumped from kinv.f
c
          include 'DIMENS.INC'
          common/nodes/numnp,logt
          dimension gkinv(j02g,j02g)

          open(95,file='gkinv.out',status='old',form='unformatted',
     *       access='readonly')
          read(95)nnodes,gkinv
          close(95)
          if(numnp.eq.nnodes) return

c         otherwise gkinv was built using a different number of nodes
          write(6,*)' error -  gkinv was built using ',nnodes,' nodes'
          stop
          end
c
c*************************************************************************
c
          integer function ind1(i,j)
c
c         funtion to determine position in dhdt matrix stored by columns
c
          common/nodes/numnp,logt
          ind1=(j-1)*numnp+i
          return
          end
c
c*************************************************************************
c
          integer function ind2(i,jj,kk)
c
c         function to determine position in dtth matrix stored by columns. Only
c         the last npt - (np-1) columns are stored for each submatrix. j has to
c         be greater than k in order for subroutine to work properly.
c
          common/nodes/numnp,logt
c
          j=jj
          k=kk
```

```
c           check k against j
            if(k.gt.j) then
             ks = k
             k=j
             j=ks
            endif
c
            ind2=0
            nc=0
            if(k.eq.1) goto 20
            nl=numnp-k+2
            do 10 kkk=numnp,nl,-1
             nc = nc + kkk
   10       continue
            ind2 = nc*numnp
   20       j=j-k
            ind2 = ind2 + j*numnp + i
            return
            end
c
c*******************************************************************************
c
            subroutine input(ind)
c
c           routine to read in binary data files dumped from DOUBT. uses same
c           dimensioning file as DOUBT. Arrays not used in this code are commented.
c       variable ind tells which files we need to read in
c
c           explanation of dimensioning parameters from 'DIMENS.INC'
c           j01g        =>maximum number of terms in skyline coefficent matrix
c           j02g        =>maximum number of node points
c           j02on       =>nloch, number of terms in full sparse repre of coef matrix
c           j09op       =>nupper(npt,npt)
c           j06pa   =>npt, number of t measurements (numnp for nodal represent)
c           j25ps       =>npt*numnp
c
c*******************************************************************************
c
            include 'DIMENS.INC'
            common/stuff/dtk(j12pa),kdiag(j02g),num(j02g),loct1(j02on),
     *              loct2(j02on),loctpt(j06pa),lptdtk(j06pa),gk(j01g),
     *              dhdt(j25ps),ptt(j09op),dtksub(j02g,j02g),hk(j02g),
     *              tt(j06pa)
            common/nodes/numnp,logt
c
       if(ind.eq.1)then
            open(95,file='params.out',status='old',form='unformatted',
     *      access='readonly')
            read(95)nk
            read(95)ndtk
```

```
            read(95)jj12pa,dtk
            if(jj12pa.ne.j12pa) goto 999
            read(95)jj02g,kdiag
            if(jj02g.ne.j02g) goto 999
            read(95)jj02g,num
            if(jj02g.ne.j02g) goto 999
            read(95)jj02on,loct1
            if(jj02on.ne.j02on) goto 999
            read(95)jj02on,loct2
            if(jj02on.ne.j02on) goto 999
            read(95)jj06pa,loctpt
            if(jj06pa.ne.j06pa) goto 999
            close(95)
      endif

      open(95,file='senpro.out',form='unformatted')
            read(95) jj01g,gk
            if(jj01g.ne.j01g) goto 999
            read(95) jj25ps,dhdt
            if(jj25ps.ne.j25ps) goto 999
            read(95)jj02g,hk
            if(jj02g.ne.j02g) goto 999
            read(95) jj06pa,tt
            if(jj06pa.ne.j06pa) goto 999
            read(95) numnp
            read(95) logt
            close(95)

      open(95,file='ptt.out',form='unformatted')
            read(95) jj09op,ptt
            if(jj09op.ne.j09op) goto 999
      close(95)

            return
  999       write(6,*)' error in INPUT, dimensions don"t match'
            stop
            end
c
c*******************************************************************************
c
            subroutine iprod(prod)
c
c           routine to build the intermediate product
c                                       hxt
c            (-Dt'K)(It * Dt'h) + (-Dt'k)(E    )(Dt'h * It)
c                                       txh
c           where * represents a matrix outer product.  The result is a rectangular
c           matrix of dimension (numnp x npt**2). (BIG)

c           The result is stored by columns in prod. The logic for this
c           dimension has to do with the structure of the DttH matrix. Each
```

```
c          sucessive submatrix has one more column duplicated in the previous
c          matrix.  By avoiding calculating these columns now I am saving
c          computations and storage.
c
           include 'DIMENS.INC'
           common/stuff/dtk(j12pa),kdiag(j02g),num(j02g),loct1(j02on),
     *          loct2(j02on),loctpt(j06pa),lptdtk(j06pa),gk(j01g),
     *          dhdt(j25ps),ptt(j09op),dtksub(j02g,j02g),hk(j02g),
     *          tt(j06pa)
           common/nodes/numnp,logt
           dimension subrws(j02g*j02g),prod(numnp*nupper(numnp,numnp))

           npt=numnp
           k=0
           l2=0
           do 10 np=1,npt
             l1=l2 +1
             l2=loctpt(np)
             call clearm(dtksub,npt)
             call filsub(l1,l2)
             call fulrws(dtksub,npt,subrws)
             call rwsmc(subrws,dhdt,npt,np,prod)
             call hpro(subrws,dhdt,np,npt,prod)
             if(logt.eq.2) call epro(subrws,hk,npt,prod)
  10       continue
           return
           end
c
c*******************************************************************************
c
c
           subroutine hpro(subrws,dhdt,np,npt,prod)
c
c          routine to calculate contribution of product of np'th DTK submatrix
c          with permutation matrix times Dt'h * It.  Answers are put into
c          proper place in PROD (hard product) matrix.
c
c          symmetry of the eventual DTTH matrix is taken advantage of in this
c          routine. the np'th submatrix of prod only needs npt-np+1 columns built.
c
c          in addition to indicating which DTK submatrix is being sent in, np
c          also indicates which submatrix column will be built by this
c          call to the subroutine.
c
c          operate on rows of DTKsub in outer loop. Increment columns of DHDT
c          in inner loop to calculate same position in successive prod
c          submatrices.
c
c          ic = index for PROD column being filled
c          ir = index for PROD row being filled
```

```
c           nt2 = number of times to go through the 10 loop for this call
c           ind = position indicator for prod matrix
c           inc = incrementer to tell how many less columns to move over in 10 loop
c
            include 'DIMENS.INC'
            dimension subrws(npt*npt),dhdt(npt*npt)
          dimension prod(npt*nupper(npt,npt))
c
            nptsq = npt*npt
            nt2=np*npt
            ir=0
            do 20 i=1,nptsq,npt
              ir = ir + 1
              ic = np
              inc=0
              do 10 j=1,nt2,npt
                inc=inc+1
                ind=(ic-1)*npt + ir
                prod(ind) = prod(ind) + dot(subrws(i),dhdt(j),npt)
                ic = ic + npt - inc
     10       continue
     20     continue
            return
            end
c
c*******************************************************************************
c
            integer function nupper(i,j)
c
c           function to find location of the (i,j)th element of a matrix stored
c           in upper triangular form. for lower triangular storage just reverse
c           the arguments in the calling statement
            if(j.ge.i) nupper = (((j-1)*j)/2)+i
            if(j.lt.i) nupper = (((i-1)*i)/2)+j
            return
            end
c
c*******************************************************************************
c
            subroutine rmuc(a,b,n,c)
c
c           this subroutine is similar to rwsmc but it is designed for one call
c           only.  every column of b is multiplied by each row of a to produce
c           the columns of c.
c
            include 'DIMENS.INC'
            dimension a(n*n),b(n*nupper(n,n)),c(n*nupper(n,n))

            k=0
            n2=n*n
```

```
            nc=n*nupper(n,n)
            do 20 j=1,nc,n
              do 20 i=1,n2,n
                k=k+1
                c(k)=dot(a(i),b(j),n)
   20       continue
            return
            end
c
c*******************************************************************************
c
            subroutine rwsmc(a,b,npt,np,c)
c
c           routine to do matrix multiplication of matrices stored by columns
c           and rows (see below). the routine is designed for successive calls
c           in that one less column of the product matrix is calculated each
c           for each call.
c
c           does a X b = c
c
c           a(npt,npt)   =>   a(npt**2)   stored by rows linearly
c           b(npt,npt)   =>   b(npt**2)   stored by columns likewise
c           c(npt,npt)   =>   see below   stored by columns, answer
c
c           k increments continually through successive calls to rwsmc.
c           np indexs which submatrix is being built. each call to rwsmc means
c           one less column needs to be built. i indexes rows of a and j indexes
c           columns of b.
c
            include 'DIMENS.INC'
            dimension a(npt*npt),b(npt*npt),c(npt*nupper(npt,npt))
c
            nptsq=npt*npt
            icb=1+(np-1)*npt
            do 20 j=icb,nptsq,npt
              do 20 i=1,nptsq,npt
                k =k + 1
                c(k) = c(k) + dot(a(i),b(j),npt)
   20       continue
            return
            end
c
c*******************************************************************************
c
            subroutine quad(dtth,dhdt,ptt,dttr)
c
c           routine to do the quadruple sum in order to evaluate the senstitivities
c           of the head covariance matrix trace to transmissivity measurements at
c           each node of the finite element grid. Indices for the dtth and dhdt
c           matrix are the whole trick.  There is an integer function for both.
```

```
c
c          id2c = index for dtth that stays constant inside 10 loop
c          id2  = index for dtth that varies inside 10 loop
c          id1  = index for dhdt that varies inside 10 loop
c          id1c = index for dhdt that stays constant inside 10 loop
c          nt   = index for number of time through the 10 loop
c
           common/nodes/numnp,logt
           dimension dtth(numnp*nupper(numnp,numnp)),dhdt(numnp*numnp),
     *                ptt(nupper(numnp,numnp)),dttr(numnp)

           n=numnp
         nt=0
           do 40 nl=1,n
             sum=0.
             do 30 i=1,n
               do 20 j=1,n
            id2c=ind2(i,nl,j)
                dtthc=dtth(id2c)
                id1c=ind1(i,j)
                dhdtc=dhdt(id1c)
                do 10 k=1,n
                      v=0.
                      id1 = ind1(i,k)
                      dhdtid = dhdt(id1)
                      v = dtthc*dhdtid
                      id2=ind2(i,nl,k)
                      dtthid=dtth(id2)
                      v=v+dtthid*dhdtc
                  pttn = ptt(nupper(j,k))
                      sum = sum + v*pttn
                      nt = nt + 1
    10             continue
    20           continue
    30           continue
                 dttr(nl) = sum
    40         continue
           return
           end
c
c*******************************************************************************
c
           subroutine scale(a,con,n)
c
c          routine to scale the array a by a constant
c
           dimension a(n)
           do 10 i=1,n
             a(i) = con*a(i)
    10     continue
```

```
            return
            end
c
c*******************************************************************************
c
            subroutine vecmat(vec,rmat,numnp,ans)
c
c           routine to multiply a vector by a matrix
c
            dimension vec(numnp),rmat(numnp*numnp),ans(numnp)
c           calculate indirect effect directly by multiplying dJdh bye dhdt
       n=0
            do 10 i=1,numnp*numnp,numnp
              n = n + 1
              ans(n) = dot(vec,rmat(i),numnp)
   10       continue

            return
            end
c
c*******************************************************************************
c
            subroutine symful(asym,nasym,n,aful)
c
c           routine to create a full 2d matrix from a symmetricly stored 1d matrix
c
c           n is the order of the matrices. remember upper of lower storage modes
c           are the same, hence the reversing of subscripts in the 10 loop.
c
            dimension asym(nasym),aful(169,169)

            k=0
            do 20 i=1,n
             do 10 j=1,i
               k=k+1
               aful(i,j)=asym(k)
               aful(j,i)=asym(k)
   10        continue
   20    continue
            if(k.gt.nasym) goto 999
            return
999         write(6,*)' error in SYMFUL, asym dimensioned too small'
            stop
            end
c
c*******************************************************************************
```

DIMENS.INC Listing

```
C*************************************************************************
c presently set for nps=1,npt=99,npa=22,npb=22,max(nloc(i))=80
c                  ndata=35,nxy=99,neqn=ndata+lk
C*************************************************************************
C
C          DATA FILE FOR DIMENSIONING DOUBT.FOR.  THE PARAMETERS DEFINED IN
C          THIS FILE ARE USED IN DOUBT TO DIMENSION ARRAYS OCCURING IN NAMED
C          COMMON BLOCKS.  THIS FILE IS CALLED IN DOUBT USING AN "INCLUDE"
C          STATEMENT.  EACH SUBROUTINE IN DOUBT THAT REQUIRES USE OF COMMON
C          BLOCK DATA HAS AN "INCLUDE" STATEMENT.
C
C          THE IDENTIFICATION SCHEME FOR THE PARAMETERS IS AS FOLLOWS:
C
C             J -  NUMBER  - LETTER IDENTIFYING COMMON BLOCK IN WHICH
C                          ARRAY FIRST OCCURS.
C
C       THE LOGIC BEHIND THIS APPROACH IS BASED ON THE NEED FOR A UNIQUE
C       PARAMETER THAT WOULD NOT BE USED AS A VARIABLE ANYWHERE IN DOUBT.
C          SWEN AND TONY FELT IT WAS POOR PROGRAMMING PRACTICE TO USE
C        ALPHABETIC AND NUMERIC CHARACTERS IN THE SAME VARIABLE; THEREFORE,
C          WE DID EXACTLY THAT TO BE ENSURE UNIQUE PARAMETERS SINCE THIS
C          PRACTICE DOESN'T OCCUR IN DOUBT     FOR THE MOST PART.
C
C*************************************************************************
C
C       RE-DIMENSIONING
C       ===============
C
C          IN ORDER TO REDIMENSION DOUBT, THE FIRST STEP IS TO GO THROUGH
C          TABLE 8-2 IN THE CERT USER'S MANUAL AND FOLLOW THE INSTRUCTIONS
C     TO DETERMINE THE MAXIMUM SIZE OF EACH VARIABLE.  THEN THESE MAXIMUM
C    VALUES MUST BE MANUALLY ENTERED IN THE BLOCK DATA SUBROUTINE IN DOUBT.
C          THEN FOR ACTUAL DIMENSIONING, THE VALUES OF THE PARAMETERS IN THIS

C       FILE MUST BE CHANGED ACCORDING TO THE BRIEF INSTRUCTIONS BEFORE EACH
C          PARAMETER STATEMENT.  THESE INSTRUCTIONS ARE TAKEN FROM TABLE 8-1.
C          FOR FURTHER CLARIFICATION REFER TO THIS TABLE.
C
C       IN SUMMARY, THERE ARE TWO PLACES THAT REQUIRE CHANGES FOR
C          REDIMENSIONING.  ONE IS IN THE PARAMETER STATEMENTS IN THIS FILE
C       AND THE OTHER IS IN THE BLOCK DATA SUBROUTINE IN DOUBT.
C
C    BEGINNING OF PARAMETER INSTRUCTIONS
C
C          J01G=NK      => NUMNP*semi-bandwidth
C          J02G=NUMNP   => # of node points
C          J03G=MJCOL   => max # non-zero terms in a row/column of GK (see p.155)
```

```
C         J04G=NUMEL  => # of elements
C         J05G=NLOCA  => total # of nodes on all boundaries
C         J06G=NLOCH  => (degree + 1)*NUMNP
C         J07G=NZLNKS => # of piecewise linear boundaries
C         J08G=NZPTS  => # of zones of node points which define well clusters
C         J09G=       => # of well in all NZPTS
C         J10G=NZELTS => # of zones of elements for recharge or leakage.
C         J11G=NEL(I) => # of elements in zones NZELTS
C
          PARAMETER  (J01G=2000, J02G=99,  J03G=10, J04G=160, J05G=44,
     $           J06G=800,  J07G=22,  J08G=1,  J09G=1,   J10G=1,
     $           J11G=1)
C
C
C         J01H=NTMH   => # of times at which head measurements are made.
C         J02H=max(NLOCHH) => 3*max # of head measurements at a given time.
C         J03H=MNHML  => total possible head measurement locations
C         J04H=NMLH   => total # of head measurements at non-nodal locations
C         J05H=NMNH   => total # of head measurements at nodal locations
C      J06H=NCVL   => total # of locs to calculate non-nodal head variances
C      J07H=3*NCVL
C
          PARAMETER  (J01H=1 , J02H=240 , J03H=100  , J04H=100 ,J05H=100,
     $           J06H=1 , J07H=3)
C
C
C         J01K=max(NUMNP,NUMEL)
C         J02K=NUPPER(NEQN,NEQN)  =>NEQN = NDATA+6
C         J03K=NEQN*NXY => NXY =># of locations at which kriged values are needed
C         J04K=N0*NDATA => N0 =># of nearest neighbors for identifying GC
C                          =>  for kriging N0 is set to NDATA, after GC
C                                    identification has taken place.
C         J05K=NUPPER(N0+1,N0+1)*NDATA
C         J06K=NDATA  =># of data pts passed to SUB AKRIP, (NSTM for kriging)
C         J07K=NMODEL =>max # of covariograms to test
C
          PARAMETER  (J01K=160 ,  J02K=1500, J03K=4059 , J04K=1225,
     $           J05K=23310,  J06K=35 ,  J07K=30)
C
C
C         J01ON=NLOCS
C         J02ON=NLOCT
C         J03ON=NLOCA   =>total number of nodes on boundaries (again)
C      J04ON=NLOCB
C         J05OP=max(II) =>?limit of II iterations per line search, >30
C         J06OP=NPUTOT  => NPS + NPT + NPA + NPB
C         J07OP=max(K)  =>(?MAXK) MAXK is calculated in SUB SEARCH
C         J08OP=NUPPER(NPS,NPS) =>PSS
C         J09OP=NUPPER(NPT,NPT) =>PTT
C         J10OP=NUPPER(NPA,NPA) =>PAA
```

```
C         J11OP=NUPPER(NPB,NPB) =>PBB              npt=99 (for most part)
C         J12OP=NPS*NPT         =>PST              nps=1
C         J13OP=NPS*NPA         =>PSA              npa=22
C         J14OP=NPS*NPB         =>PSB              npb=22
C         J15OP=NPT*NPA         =>PTA              max(nloc(i))=80
C         J16OP=NPT*NPB           =>PTB
C         J17OP=NPA*NPB         =>PAB
C      J18OU=NSOL    =># of times at which solutions are to be calculated
C
          PARAMETER (J01ON=99 ,   J02ON=615 , J03ON=44 , J04ON=22,
     $          J05OP=50 ,   J06OP=144 , J07OP=200, J08OP=1,
     $          J09OP=4950,  J10OP=253,  J11OP=253,
     $          J12OP=99,    J13OP=22,   J14OP=22,
     $          J15OP=2175,  J16OP=2175, J17OP=484, J18OU=1)
C
C         J01PA=NLTGPS =># of elements in groups for parameterizing S
C         J02PA=NLTGPT =>ditto for T
C         J03PA=NODGPS =># of nodes in groups for parameterizing S
C         J04PA=NODGPT =>ditto for T
C         J05PA=NPS =>max(NUMNP,NUMEL) or smaller depending on IPS, see p.155
C         J06PA=NPT =>max(NUMNP,NUMEL) or smaller depending on IPT, see p.155
C      J07PA=NPA => same as NBOUND at present; also MSIDE = MNPA in BLOCK DATA
C         J08PA=NPB => total # of boundary parameters in BT, spatial and temp.
C         J09PA=NLOCBB => # of terms in ADB for all spatial parameters only
C         J10PA=NBOUND => NZLNKS + NZPTS + NZELTS
C         J11PA=NDSS => depends on IPS (see p.157 of manual)
C         J12PA=NDTK => similar to NDSS
C         J13PA=NDAK => see p.157, but description doesn't always work
C         J14PA=sum I=1,NBOUND of NTBT(I) => total # of b measurements
C         J15PA=NFRAC => see p.157
C         J16PM=NSTM => total number of S and T measurements
C         J17PM=NABM => total number of "a" and b measurements
C         J18PO=max(NLOC(I)*NPS
C         J19PO=max(NLOC(I)*NPT
C         J20PO=max(NLOC(I)*NPA
C         J21PO=max(NLOC(I)*NPB
C         J22PR=max(NPS,NPT)
C         J23PS=NUPPER(NUMNP,NUMNP)  =>PHH
C         J24PS=NUMNP*NPS
C         J25PS=NUMNP*NPT
C         J26PS=NUMNP*NPA
C         J27PS=NUMNP*NPB
C
          PARAMETER (J01PA=1,  J02PA=1 ,  J03PA=1 ,  J04PA=1 ,
     $          J05PA=1 ,  J06PA=160, J07PA=22 , J08PA=22,
     $          J09PA=22,  J10PA=22,  J11PA=917, J12PA=2276,
     $          J13PA=300, J14PA=22,  J15PA=22,  J16PM=160,
     $          J17PM=22,  J18PO=80,  J19PO=7920,
     $          J20PO=1760, J21PO=1760, J22PR=99,  J23PS=4950,
     $          J24PS=99,  J25PS=9801, J26PS=2178,
```

```
     $            J27PS=2178)
C
C          J01U=max(NLOC(I))*NUMNP
C       J02U=max(NUPPER(NLOC(I),NLOC(I)))
C          J03W=  =>at present sufficient for MWORKI=max(MNSTM,MMAXI)
C          JO4W=  =>if NUMEL>NUMNP, MWORK should be >= NUMNP*NUMEL
C
           PARAMETER  (J01U=7920 , J02U=3240 , J03W=160 , J04W=15840 )
C
C    END OF FILE
```

Appendix E

Data Files

Several data files are included in Appendix D. The original CERT input and output files for the sixty-one node grid from the application in *Magnuson and Wilson,* 1988 are the most important. These contain in order the finite element grid, the thirty log transmissivity values from the synthetic field, the maximum likelihood estimated variogram parameters, the log transmissivity kriging results, the first order, second moment covariance propagation results, the trace sensitivities, and the perturbation trace reduction results. Data files from all the other runs are similar and not included.

CERT Input File for Thirty Measurements

```
61 NODES, KRIGING & UNCERTAINTY PROPAGATION, GRID 3
  1   0  99   1   GEOM
 94  61      1.      1.   F   F
      1    0.0     0.0
      2    0.0    10.0
      3    0.0    20.0
      4    0.0    30.0
      5    0.0    40.0
      6    0.0    50.0
      7    0.0    60.0
      8   10.0     0.0
      9   10.0    10.0
     10   10.0    20.0
     11   10.0    30.0
     12   10.0    40.0
     13   10.0    50.0
     14   10.0    60.0
     15   20.0     0.0
     16   20.0    10.0
     17   20.0    20.0
     18   20.0    30.0
     19   20.0    40.0
     20   20.0    50.0
     21   20.0    60.0
     22   30.0     0.0
     23   30.0    10.0
     24   30.0    20.0
     25   30.0    30.0
     26   30.0    40.0
     27   30.0    50.0
     28   30.0    60.0
     29   40.0     0.0
     30   40.0    10.0
     31   40.0    20.0
     32   40.0    30.0
     33   40.0    40.0
     34   40.0    50.0
     35   40.0    60.0
     36   50.0     0.0
```

37	50.0	10.0	
38	50.0	20.0	
39	50.0	30.0	
40	50.0	40.0	
41	50.0	50.0	
42	50.0	60.0	
43	60.0	0.0	
44	60.0	10.0	
45	60.0	20.0	
46	60.0	30.0	
47	60.0	40.0	
48	60.0	50.0	
49	60.0	60.0	
50	70.0	0.0	
51	70.0	10.0	
52	70.0	20.0	
53	70.0	30.0	
54	70.0	40.0	
55	70.0	50.0	
56	80.0	0.0	
57	80.0	10.0	
58	80.0	20.0	
59	80.0	30.0	
60	80.0	40.0	
61	90.0	30.0	
1	9	1	8
2	9	2	1
3	9	3	2
4	9	10	3
5	9	17	10
6	9	16	17
7	9	15	16
8	9	8	15
9	11	3	10
10	11	4	3
11	11	5	4
12	11	12	5
13	11	19	12
14	11	18	19
15	11	17	18
16	11	10	17
17	13	5	12
18	13	6	5
19	13	7	6
20	13	14	7
21	13	21	14
22	13	20	21
23	13	19	20
24	13	12	19
25	23	15	22
26	23	16	15
27	23	17	16
28	23	24	17
29	23	31	24
30	23	30	31
31	23	29	30
32	23	22	29
33	25	17	24
34	25	18	17
35	25	19	18
36	25	26	19

37	25	33	26
38	25	32	33
39	25	31	32
40	25	24	31
41	27	19	26
42	27	20	19
43	27	21	20
44	27	28	21
45	27	35	28
46	27	34	35
47	27	33	34
48	27	26	33
49	37	29	36
50	37	30	29
51	37	31	30
52	37	38	31
53	37	45	38
54	37	44	45
55	37	43	44
56	37	36	43
57	39	31	38
58	39	32	31
59	39	33	32
60	39	40	33
61	39	47	40
62	39	46	47
63	39	45	46
64	39	38	45
65	41	33	40
66	41	34	33
67	41	35	34
68	41	42	35
69	41	49	42
70	41	48	49
71	41	47	48
72	41	40	47
73	51	43	50
74	51	44	43
75	51	45	44
76	51	52	45
77	51	58	52
78	51	57	58
79	51	56	57
80	51	50	56
81	53	45	52
82	53	46	45
83	53	47	46
84	53	54	47
85	53	60	54
86	53	59	60
87	53	58	59
88	53	52	58
89	55	47	54
90	55	48	47
91	55	49	48
92	55	54	60
93	59	58	61
94	59	61	60

2 0 0
7
1 2 3 4 5 6 7

```
4
56  57  58  61
 1   0  99   1   PARAMS
 1   2
 1   1   0   1   1
    0.
 1   2   0   1   1
    0.
 1   0  99   1   PMEAS
    0      30
   39.      8.      0.   0.3161      0.      0.
    7.     41.      0.   2.4581      0.      0.
   34.      8.      0.   0.1446      0.      0.
   14.     45.      0.   1.7996      0.      0.
   24.     38.      0.   1.9174      0.      0.
   75.     20.      0.   0.8836      0.      0.
   47.     15.      0.   1.0988      0.      0.
    6.     43.      0.   2.7870      0.      0.
   50.      9.      0.   1.0553      0.      0.
   69.      8.      0.   0.9947      0.      0.
   30.     49.      0.   2.2237      0.      0.
   18.     40.      0.   1.3284      0.      0.
   46.      1.      0.   2.9150      0.      0.
   48.     15.      0.   1.0405      0.      0.
   40.     28.      0.   1.5900      0.      0.
   32.      7.      0.   0.7119      0.      0.
    3.     42.      0.   2.1949      0.      0.
   38.     47.      0.   1.6157      0.      0.
   30.     29.      0.   2.2693      0.      0.
   37.     37.      0.   2.4595      0.      0.
   11.      3.      0.   1.1776      0.      0.
   24.     23.      0.   1.7112      0.      0.
   76.     36.      0.   2.3931      0.      0.
   65.     14.      0.   0.3540      0.      0.
   34.     20.      0.   1.4583      0.      0.
   32.     57.      0.   3.4212      0.      0.
   75.     10.      0.   0.8283      0.      0.
   69.     30.      0.   1.1574      0.      0.
   38.     11.      0.   0.4093      0.      0.
   46.     52.      0.   2.9837      0.      0.
    2      0
    1      1   1.0E+15     10.      0.      0.
    2     56   1.0E+15      0.      0.      0.
 1   0  99   1    HMEAS
    0      0      0
 1   0  99   1    SOLN
T   T   F
   0.      0.      0.   0.   0.   0   0
4
F   T   F   F
0   1   1   1   1   1
 1  67  99   1   PREPRO
 1   3   0   2   0   0   0   0
 1   0 0.0      0.   0.   0.   15   0   0   0   0 0.9   1
  20.8    0.992
2   2   0
F
0   0   0.   F   F
1
1   0  99   1    RUNIT
```

CERT Results for Thirty Measurements

VALUES OF kriged Log transmissivity

1.535e+00	1.573e+00	1.723e+00	1.913e+00	2.122e+00	2.311e+00	2.276e+00	1.284e+00	1.364e+00	1.608e+00
1.806e+00	2.116e+00	2.284e+00	2.375e+00	1.265e+00	1.220e+00	1.552e+00	1.798e+00	1.523e+00	2.132e+00
2.599e+00	1.233e+00	7.656e-01	1.498e+00	2.255e+00	2.102e+00	2.335e+00	3.185e+00	1.832e+00	5.064e-01
1.221e+00	1.781e+00	2.148e+00	2.254e+00	3.015e+00	2.428e+00	1.026e+00	1.103e+00	1.538e+00	2.062e+00
2.672e+00	2.812e+00	1.664e+00	8.976e-01	7.917e-01	1.319e+00	1.966e+00	2.400e+00	2.530e+00	1.396e+00
8.255e-01	7.320e-01	1.243e+00	2.120e+00	2.320e+00	1.404e+00	1.087e+00	1.180e+00	1.798e+00	2.314e+00
1.867e+00									

VALUES OF kriging standard deviations for Log transmissivity

8.321e-01	8.412e-01	8.726e-01	8.024e-01	5.392e-01	7.346e-01	9.109e-01	5.128e-01	6.727e-01	7.829e-01
7.241e-01	4.492e-01	6.228e-01	8.555e-01	7.087e-01	6.827e-01	5.879e-01	6.023e-01	3.628e-01	6.238e-01
7.831e-01	6.761e-01	4.904e-01	4.787e-01	2.908e-01	5.327e-01	2.887e-01	5.384e-01	5.857e-01	3.471e-01
5.303e-01	3.868e-01	5.077e-01	4.607e-01	6.686e-01	5.433e-01	2.834e-01	5.829e-01	7.051e-01	7.236e-01
5.770e-01	7.556e-01	7.348e-01	5.807e-01	6.369e-01	6.964e-01	7.692e-01	8.165e-01	8.878e-01	7.191e-01
3.801e-01	5.206e-01	2.910e-01	6.624e-01	8.550e-01	8.138e-01	6.062e-01	5.954e-01	6.408e-01	6.520e-01
8.624e-01									

VALUES OF first order, second moment estimated heads (same as deterministic in this case)

1.000e+01	1.000e+01	1.000e+01	1.000e+01	1.000e+01	1.000e+01	1.000e+01	9.411e+00	9.436e+00	9.390e+00
9.406e+00	9.369e+00	9.273e+00	9.187e+00	8.612e+00	8.656e+00	8.632e+00	8.605e+00	8.622e+00	8.568e+00
8.429e+00	7.333e+00	7.678e+00	8.092e+00	8.106e+00	8.079e+00	8.038e+00	8.042e+00	6.248e+00	6.668e+00
7.642e+00	7.714e+00	7.677e+00	7.702e+00	7.752e+00	5.771e+00	5.851e+00	6.689e+00	7.138e+00	7.253e+00
7.385e+00	7.450e+00	5.097e+00	4.775e+00	5.561e+00	6.141e+00	6.650e+00	6.941e+00	7.165e+00	2.801e+00
2.662e+00	3.046e+00	4.493e+00	5.447e+00	6.313e+00	5.318e-15	3.475e-15	1.127e-14	2.500e+00	4.237e+00
2.913e-14									

VALUES OF first order, second moment estimated head standard deviations

1.376e-15	1.765e-15	2.946e-15	4.504e-15	5.719e-15	1.008e-14	1.720e-14	5.467e-01	5.574e-01	5.386e-01
5.269e-01	5.241e-01	6.834e-01	7.664e-01	9.365e-01	8.943e-01	1.008e+00	1.034e+00	1.086e+00	1.184e+00
1.344e+00	1.359e+00	1.302e+00	1.174e+00	1.248e+00	1.340e+00	1.484e+00	1.494e+00	1.750e+00	1.484e+00
1.311e+00	1.399e+00	1.528e+00	1.589e+00	1.598e+00	1.801e+00	1.782e+00	1.421e+00	1.578e+00	1.664e+00
1.705e+00	1.693e+00	1.890e+00	1.600e+00	1.630e+00	1.758e+00	1.879e+00	1.831e+00	1.786e+00	1.768e+00
1.717e+00	1.526e+00	2.450e+00	2.286e+00	2.076e+00	5.218e-15	1.984e-15	6.228e-15	1.700e+00	2.476e+00
2.424e-14									

Trace Sensitivity Results for Thirty Measurements

VALUES OF Derivative of TRace with respect to changes in transmissvity

-1.586e+00	-1.266e+00	-3.139e+00	-2.679e+00	-1.190e+01	-6.435e+00	-4.031e+00	-1.377e+00	-3.398e+00	-2.790e+00
-9.773e+00	-1.459e+01	-1.989e+01	-3.843e+00	-1.146e+00	-8.518e-01	-4.230e+00	-4.444e+00	-5.407e+00	-6.872e+00
-6.240e+00	-3.640e-01	-1.269e+00	-1.503e+00	-5.183e+00	-1.914e+00	-5.607e+00	-9.873e-02	1.616e+00	-5.686e-01
-2.588e+00	-1.193e-01	2.606e-01	8.988e-02	2.534e+00	4.615e+00	-2.872e+00	-1.604e+00	1.023e+00	1.198e+00
5.472e+00	7.439e-01	1.389e+01	-1.453e+00	-1.304e+00	9.242e-01	1.209e+01	3.941e+00	2.081e+00	1.202e+00
-1.118e+01	-3.468e+00	-8.805e+00	1.404e+01	1.674e+01	1.022e+01	-1.599e+00	-5.350e+00	1.214e+01	4.624e+01
2.166e+01									

VALUES OF SCAled Derivatives of TRace w.r.t log transmissivity, scaled by full covariance matrix

-1.454e+00	-4.456e+00	-7.434e+00	-7.309e+00	-5.031e+00	-8.027e+00	-8.942e+00	-1.618e-01	-4.496e+00	-7.636e+00
-8.763e+00	-3.626e+00	-1.036e+01	-1.027e+01	-2.412e-01	-2.444e+00	-3.836e+00	-3.859e+00	-8.743e-01	-5.706e+00
-7.967e+00	5.066e-01	-7.277e-01	-4.379e-01	-5.193e-01	-6.426e-01	-7.455e-01	-1.203e+00	1.078e+00	-1.492e-01
-6.692e-01	3.776e-01	1.243e+00	1.830e-01	2.736e+00	3.784e+00	-2.330e-01	1.459e-01	3.881e+00	7.822e+00
7.556e+00	7.755e+00	9.800e+00	1.458e+00	-1.100e-01	5.018e+00	1.705e+01	1.955e+01	1.631e+01	7.045e+00
-2.076e+00	-1.934e+00	-5.214e-01	1.743e+01	2.837e+01	1.055e+01	3.652e+00	3.916e+00	1.225e+01	2.807e+01
2.592e+01									

Perturbation Trace Reductions for Thirty Measurements

VALUES OF REDuction of traces from adding kriging measurements from the original 30 logT measurements

4.584e-01	1.224e+00	2.059e+00	1.603e+00	1.224e+00	5.129e+00	6.676e+00	7.649e-02	1.652e+00	2.332e+00
2.384e+00	3.891e-01	1.142e+01	1.064e+01	5.190e-01	8.719e-01	1.409e+00	6.003e-01	6.109e-02	3.573e+00
1.020e+01	4.739e-01	1.976e-01	7.116e-02	1.686e-01	4.742e-02	2.500e-01	2.554e+00	1.462e+00	9.018e-03
4.354e-01	5.889e-02	3.831e-02	3.171e-02	7.982e-01	1.487e+00	3.831e-02	4.661e-01	1.477e+00	1.724e+00
2.241e+00	7.925e-01	7.155e+00	8.241e-01	3.888e-01	1.407e+00	5.369e+00	4.858e+00	2.353e+00	6.680e+00
1.142e-01	4.068e-02	4.891e-01	6.357e+00	7.989e+00	9.255e+00	4.541e+00	8.956e+00	2.289e+01	2.963e+01
2.371e+01									