# COSIMULATION OF HEADS AND LOG-TRANSMISSIVITIES USING A NUMERICAL SPECTRAL-PERTURBATION METHOD

## A NEW TECHNIQUE FOR OBTAINING STOCHASTIC PREDICTIONS OF GROUNDWATER FLOW THROUGH HETEROGENEOUS POROUS MEDIA

BY



D.A. (TONY) ZIMMERMAN
ALLAN GUTJAHR
JOHN L. WILSON

HYDROLOGY REPORT NO. H88-4
(FORMERLY OPEN FILE REPORT NO. 88-4)

NEW MEXICO TECH HYDROLOGY REPORT SERIES

HYDROLOGY PROGRAM
SOCORRO, NEW MEXICO 87801

AUGUST, 1988

This report was also submitted by D.A. Zimmerman as an Independent Study in partial fulfillment of the requirement of the M.S. Degree in Hydrology

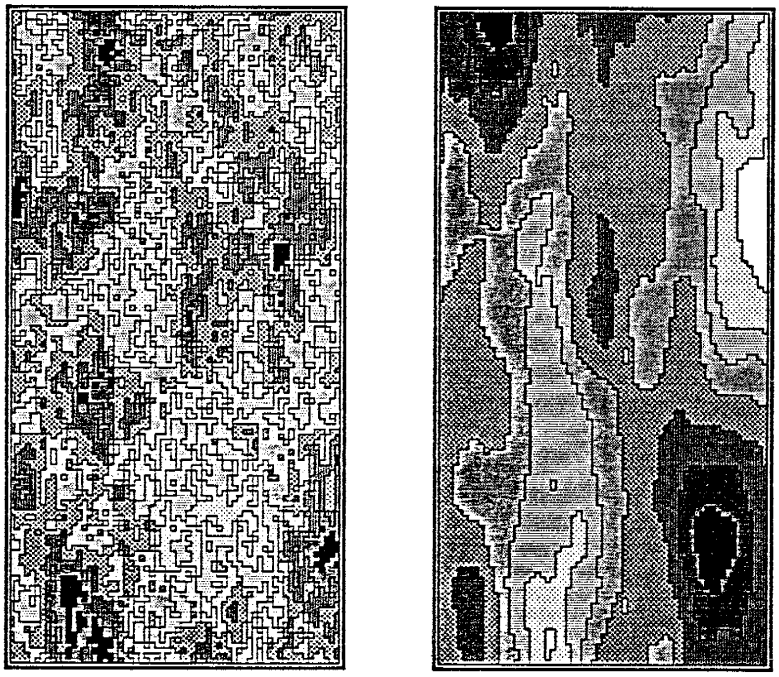
Research sponsored by:
U.S. Department of Energy
Research Grant No. DE-AC19-85BC10850

# A New Technique for Obtaining Stochastic Predictions of Groundwater Flow through Heterogeneous Porous Media:

## Cosimulation of Heads and Log-Transmissivities Using a Numerical Spectral-Perturbation Method

by

D. A. (Tony) Zimmerman



Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Hydrology

New Mexico Institute of Mining and Technology
Socorro, New Mexico
August, 1988

NEW MEXICO INSTITUTE OF MINING AND TECHNOLOGY
OFFICE OF GRADUATE STUDIES

REPORT OF M.S. ADVISORY COMMITTEE GEOSCIENCE

This form is to be used by the advisor to transmit information to the Graduate Office. Use the same form for each meeting of the advisory committee.

Student's Name D. A. (Tony) Zimmerman Date 7/27/88

Members Present Dr. John Wilson, Geoscience ; Dr. Allan Gutjahr, Mathematics

Degree Program being Followed: Master of Science ( ) with thesis
( X ) without thesis

Course Program

| | | | |
|---|---|---|---|
| HYD534 Applied Subsurface Hydrology | (3) | HYD500 Directed Research | (3) |
| HYD525 Hydrogeochemistry | (3) | HYD571 Advanced Modeling | (3) |
| HYD538 Vadose Zone Hydrology | (4) | MTH487 Geostatistics | (3) |
| HYD571P Analytic Methods | (3) | HYD545 Stochastic Methods | (3) |
| MTH430 Partial Differential Equations | (3) | MTH587 Time Series Analysis | (3) |
| | | HYD590 Independent Study | (3) |

Adviser's Acceptance [signature] Date 7/27/88

Department Chairman [signature] Date 27 July, 88

Independent Study (X)

Title A New Technique for Obtaining Stochastic Predictions of Groundwater Flow Through Heterogeneous Porous Media: Cosimulation of Heads and Log-Transmissivities Using a Numerical Spectral-Perturbation Method

Accepted: _____ Date _____

Thesis ( )

Title N/A

N/A

Defense Report: N/A

Accepted: N/A Date _____

All Requirements Completed:

Department Chairman: [signature] Date 1 August 88

# ACKNOWLEDGEMENTS

Funds for this research were provided by the U.S. Department of Energy research grant No. DE–AC19–85BC10850.

The following individuals contributed to this research: Warner Losh, Dave Peterson, Annette Schafer–Perini and Swen Magunson.

# ACKNOWLEDGEMENTS

It seems appropriate that I should first acknowledge those persons who initiated and fostered my interest in stochastic processes and influenced my decision to return to graduate school, during which time this work was undertaken: many thanks to Charlie Cole, Harlan Foote, and Britt Jacobson of Battelle Northwest Laboratories in Richland, Washington. I am extremely grateful to my advisor, Dr. John L. Wilson, for his infinite patience, insightful guidance, and excellent editorial comments. I am equally appreciative of the tremendous input I received from Dr. Allan L. Gutjahr, especially with regard to his role in helping me gain a thorough understanding of this material and in working out the details of the algorithms. I wish to thank Warner Losh, manager of the HydroVax Computer System, for continually enhancing the system for the benefit of our research, especially for putting the TEX publishing software on the system, which was used to print this document. Many thanks are due to fellow researchers Dave Peterson, Annette Shafer-Perini and Swen Magnuson for their interest and participation in discussions during the course of this work. Although not directly involved, my buddy Leon Redbone constantly made me laugh, always baying with incredible volume whenever I would step outside for a break, or sneaking in the building to get out of the heat (no dogs allowed); his companionship was always uplifting especially during the many frustrating periods of this work. Finally, I wish to thank my friend and fiancé, Cynthia Ardito, for her patience and support, especially over the last six months during which time I extended my indulgence into this esoteric research rather than getting on with the "real world."

This work was sponsored by the U.S. Department of Energy under research grant No. DE-AC19-85BC10850. Financial support is gratefully acknowledged.

## Abstract

A new technique for obtaining stochastic predictions of groundwater flow through heterogeneous aquifer materials is presented. The aquifer properties are viewed as statistically homogeneous stochastic processes which are characterized by covariance functions. The governing stochastic stochastic differential equation for flow is written for mean-removed random variables whose joint distribution is assumed to be Gaussian. This transformed equation is then linearized by applying perturbation theory. Finally, spectral theory is used to obtain a relationship between the head spectrum and the logarithm of Transmissivity (lnT) spectrum. Using this relationship, an algorithm is developed for generating discrete random fields of head and lnT simultaneously. The probability behavior of the head process can then be estimated by computing its statistics over an ensemble of fields obtained through repeated application of the cogeneration algorithm. This cosimulation procedure is computationally much cheaper and faster than the standard sequential Monte Carlo method which involves solving a system of algebraic equations for each realization of the head field obtained. However, the method is restricted to relatively small input variances.

The theory is initially developed for the infinite domain problem in order to avoid complications arising from boundary conditions. It is then extended, using geostatistical methods, to enable the cosimulations to be conditioned on measurements of T and/or head, thus addressing the bounded domain problem.

The technique is demonstrated (without conditioning) for steady, confined flow in one and two-dimensional domains with unidirectional, linear mean hydraulic gradient. Practical aspects associated with computer implementation of the procedure are discussed in detail. The cogenerated fields are examined to determine under what conditions they preserve the proper statistical behavior and satisfy the mass conservation principle.

# Table Of Contents

## List of Figures



## List of Tables

# INTRODUCTION

This study deals with numerical modeling of fluid flow through heterogeneous materials. Geohydrologists and petroleum engineers have long recognized that natural earth systems are extremely heterogeneous; ie. variations in the lithology and the hydrologic properties that characterize the flow in aquifers or petroleum resevoirs are known to be distributed in a very irregular manner in space. Furthermore, the property values are extremely variable, often spanning several orders of magnitude within a given hydrogeologic unit. These heterogenieties play a key role in determining the fate of solutes transported through the flow system and are a major cause of large-scale dispersion of contaminants in the groundwater. Groundwater contamination problems resulting from industrial waste discharges, underground storage tank leaks, landfill leachates, agricultural activities and other sources have received much attention in recent years, especially with regard to developing mitigative strategies for polluted aquifers. In addition, much of the research emphasis in the petroleum industry has focused on developing new techniques for enhancing oil recovery in heterogeneous environments [see eg *Coats et al*, 1970, 1974; *Aziz and Settari*, 1979]. Obtaining a good understanding of how the flow system behaves is an essential prerequisite to evaluating the efficacy of different remediation alternatives or recovery techniques. In recognizing the need to account for the influence of aquifer/reservoir heterogenieties on the flow behavior, groundwater hydrologists and petroleum engineers have attempted to increase their understanding of subsurface flow processes by incorporating more of the details of these complex heterogeneous systems into their analyses.

An analysis is carried out by developing a model of the flow system in which a number of simplifying assumptions are combined with a mathematical description of the flow processes; the model is then used to simulate the flow behavior under hypothetical conditions and predict the response of the system to those conditions. The methods used in the analyses have often shifted from analytical to numerical methods as the latter are more capable of handling aquifer heterogenieties, irregu-

lar geometries, and complex distributions of known state and flux. As advances in computer technology and computational methods have progressed during the past two decades, it has become possible to incorporate much more detailed descriptions of the material property variations into these numerical models [*see eg. Cole et al.*, 1985,1987]. In spite of this, a great deal of uncertainty exists in our numerical models, especially concerning the accuracy and representativeness of the input parameters. In the traditional approach to numerical modeling, the model parameters are treated as deterministic functions of space; ie. the spatial distribution of the material properties is assumed known. In reality it is inferred, usually from only a handful of measurements, some knowledge of the geology, and frequently with the aid of geostatistical methods such as kriging. Both hand interpreted and kriged values are inherently smooth interpolators and therefore do not accurately reflect the true (random) nature of the heterogenieties. Furthermore, in the process of incorporating the interpolated data into the numerical model, the continuous nature of the porous medium is approximated by discrete blocks of uniform composition, so that much of the fine scale detail of the spatial variability is lost. Modeling uncertainty thus arises from our limited ability to adequately represent the continuous and random nature of the spatial variations and from interpolation errors resulting from a paucity of field data. The reliability of predictions made with these models then naturally comes into question in light of these uncertainties.

Stochastic methods provide a means of assessing the reliability of model predictions by presenting the results of the flow analysis in terms of probabilities rather than absolute quantities. In the stochastic approach, the spatial variation of the material properties is modeled as a spatial stochastic process or random field; ie. the material properties are regarded as random variables whose magnitude variations are characterized by some joint probability distribution and whose spatial dependence is characterized by an autocovariance function. The uncertainty in the input of the stochastic model is then translated into uncertainty in the output which is evaluated statistically by computing, among other things, the variance of the model predictions.

A number of different techniques have been used to make stochastic predictions of groundwater flow in heterogeneous environments. The method of perturbations is described in *de Marsily*, [1986] and has been used for flow analysis by *Sagar*, [1978], *Gelhar and Axness*, [1983] and *Anderson and Shapiro*, [1983]; *Tang and Pinder*, [1977] used it for mass transport (see comment by *Gelhar et al*, [1979]). The first order second moment method described by *Dettinger and Wilson*, [1981] might also be classified as a perturbation technique. Spectral techniques along with perturbation analysis have been used to obtain analytical solutions [*Gelhar*, 1974; *Bakr et al.*, 1978; *Gutjahr et al.*, 1978; *Gutjahr and Gelhar*, 1981; *Mizell et al.*, 1982] but are generally limited to the analysis of stationary processes in infinite domains and whose input variance is relatively small. *Naff and Vecchia*, [1986] used Green's functions to solve for three-dimensional flows in a bounded domain. Monte Carlo simulation is the most commonly used technique and can be applied to virtually any type of problem that existing codes can model. Most importantly, it is not restricted to problems with small input variances or unbounded domains. In many of the earlier works, the spatial correlation structure of the random fields was ignored [*Warren and Price*, 1961; *Freeze*, 1975] while later studies included auto-correlated parameters [*Smith and Freeze*, 1979a,b; *Smith and Schwartz*, 1981; *Dagan*, 1982; *Clifton et al*, 1984]. The major drawback with the Monte Carlo approach is its computational expense. The probability behavior of the output process is estimated by calculating the statistics of the output over an ensemble of realizations. However, for each realization of the output, a system of equations must be solved, which is an expensive and time consuming computational step.

Emphasis on reducing the uncertainty in these stochastic models led to the formulation of the inverse problem where the input parameters (eg, hydraulic conductvity, permeability, boundary conditions) are treated as dependent variables which are estimated with the aid of head observations. Further reduction of model prediction uncertainty is accomplished by requiring each realization of the input to conform locally to measured values using geostatistical techniques in a procedure known as

conditional simulation [*Delhomme*, 1979; *Clifton and Neuman*, 1982; *Kitanidis and Vomvoris*, 1983; *Townley and Wilson*, 1983; *Hoeksema and Kitanidis*, 1984, 1985a,b].

In the present study, we propose a new technique in which the input and output processes are generated simultaneously. The cosimulation approach is illustrated schematically and contrasted with the standard sequential method in Figure 1. The proposed method combines elements from the perturbation and spectral analysis techniques with Monte Carlo simulation to obtain stochastic predictions of groundwater flow. The theory is developed for conditional cosimulations in which observations of both transmissivity *and* heads are preserved. Important practical applications of this kind of modeling include predictions of oil-water cut, sweep efficiency and other oil recovery performance measures as well as travelpath, traveltime and contaminant concentration predictions in the field of hydrology. A major advantage of the proposed method is the significant reduction in the computational effort required to amass the statistics of the output (head) process. The output head and input log-transmissivity fields are cogenerated using standard techniques for generating random fields which do not require the solution of a system of equations. As a first step toward evaluating the feasibility of such a method, we test it on a simple conceptual model of steady, confined flow in a system with unidirectional linear mean hydraulic gradient. The viability of the method is evaluated by examining the individual and joint behavior of the cosimulated random fields (head and transmissivity) to determine if, or under what conditions they preserve the proper statistical relationships and satisfy the mass conservation principle.

We begin by listing the assumptions involved and then describe the cosimulation algorithm. Next we show the theoretical analysis, in one dimension, for the statistical and mass conservation measures. Finally, the technique is demonstrated for one- and two-dimensional domains and the results are compared with the the theory.

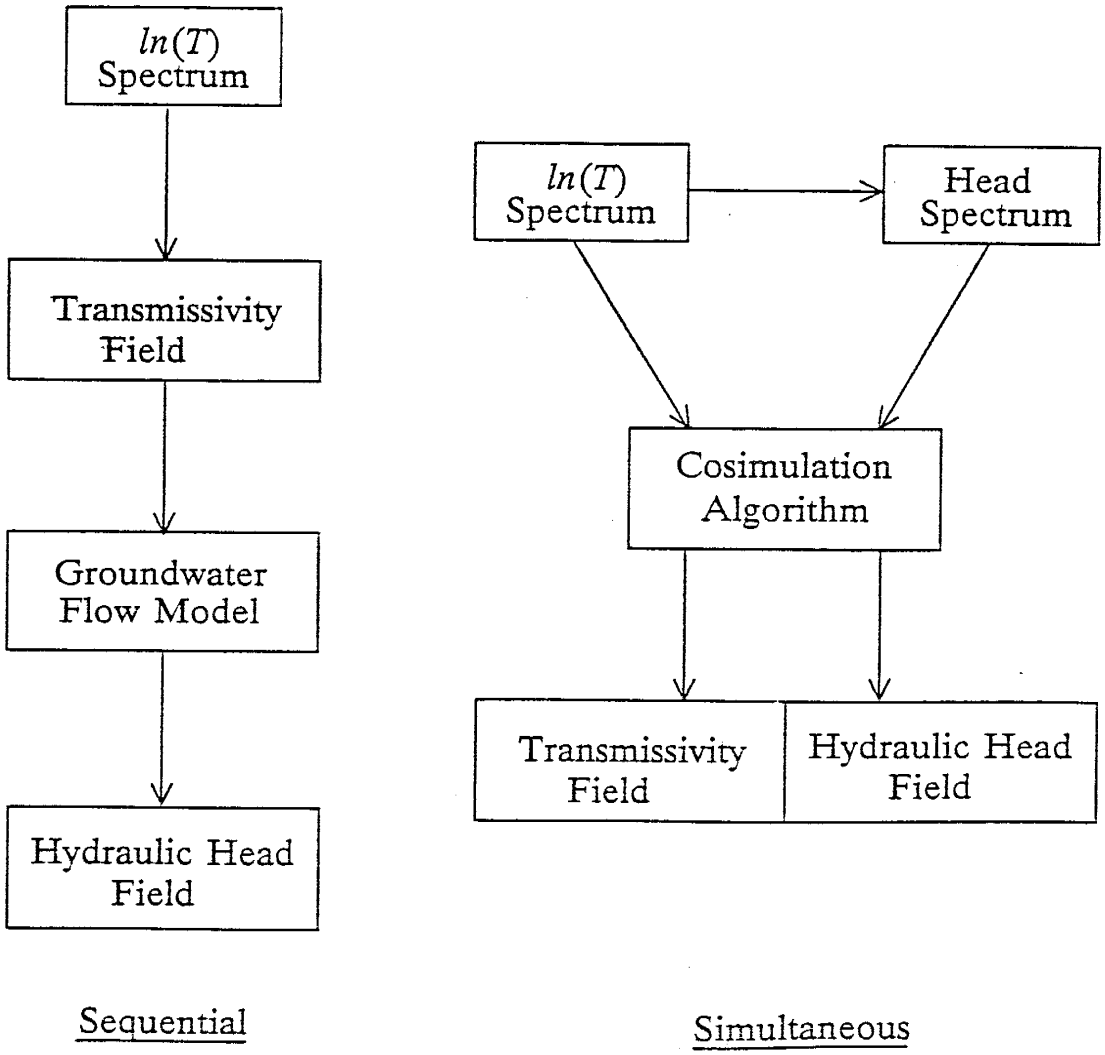



Figure 1. Schematic illustrating the difference between the classical sequential approach and the cosimulation approach.

## Assumptions Used in the Cosimulation Approach

The conceptual model and spectral analysis approach is the same as that used by *Bakr et al.*, [1978] for one and three dimensions, and *Mizell et al.*, [1982] for two dimensions in their stochastic analysis of groundwater flow. The analysis involves a transformation of the governing differential equation into an algebraic equation relating the spectrum of the head field to the spectrum of the log-hydraulic conductivity field and is described in detail in both of those papers. Here, we'll briefly review the steps in the analysis in order to emphasize the inherent assumptions involved.

We begin by treating the parameters and state variable of the governing partial differential equation for steady, confined flow in a heterogeneous medium as spatial stochastic processes or random fields. The head and conductivity fields are considered to be continuous processes whose magnitude variations are described by probability density functions at every point in space. In reality, there is only one (unique) realization that can be used to characterize this joint probability behavior, therefore measurements taken at a few locations in the real field are used to estimate the parameters which characterize the joint probability distribution of the hypothetical random field. The ergodic hypothesis, which is essentially a statement of equivalence between ensemble and spatial statistics, is therefore implicitly assumed.

The form of the distribution of log-hydraulic conductivity is assumed to be Gaussian, which can be completely described by its mean and variance statistics [*Vanmarke*, 1984]. This assumption is reasonable for the log-transformed conductivity process [*Freeze*, 1975; *Bakr*, 1976; *Neuman*, 1982] and therefore the governing stochastic differential equation is written in terms of the logarithm of hydraulic conductivity ($lnK$) or the logarithm of transmissivity ($lnT$).

For simplicity, the mathematics will be presented in a one-dimensional framework. The governing differential equation for steady, confined flow in one dimension

$$\frac{d}{dx}\left(T\frac{d\phi}{dx}\right) = 0, \tag{1}$$

is written in terms of $lnT$ as

$$\frac{d^2\phi}{dx^2} + \frac{d\,lnT}{dx}\frac{d\phi}{dx} = 0 \tag{2}$$

where the head, $\phi$, and $lnT$ variables are considered to be spatial stochastic processes or random fields.

The spatial correlation structure of each random field is characterized by an auto-covariance function; the head covariance function is derived from the $lnT$ covariance model whose functional form may be inferred from field data, or simply assumed. In Bakr's one-dimensional model, the $lnT$ covariance function, $R(\xi)$, is a hole function model of the form

$$R(\xi) \;=\; \sigma^2(1 - |\xi|/l)\, e^{-|\xi|/l} \tag{3}$$

where $\sigma^2$ is the variance of the process, $\xi$ is the lag or separation distance, and $l$ is a correlation length parameter. Mizell used two modified forms of the Whittle covariance function for his two-dimensional model of the $lnT$ covariance; here we use the Telis covariance model [*Mantoglou and Wilson*, 1981,1982]. Because field data are generally sparse, the covariance model is assumed to hold over the entire domain and the random field is thus assumed to be 2nd order stationary or statistically homogeneous. Stationarity is also a requirement of the mathematics used in the field generation method.

The stochastic variables $\phi$ and $lnT$ are assumed to be composed of a non-random mean part and random fluctuations about the mean. Only the random behavior of these variables is of interest as it is assumed that the non-random part is known. The governing equation is thus written to describe the behavior of the perturbations in flow by subtracting out the mean flow components. Using the same notation as in *Bakr et al*, [1978], we write the stochastic variables for one-dimensional flow as

$$\begin{aligned} \phi(x) \;&=\; H(x) \;+\; h(x) \\ lnT(x) \;&=\; F \;+\; f(x) \end{aligned} \tag{4}$$

where $H(x)$ is a non-random linear function of $x$ representing the mean hydraulic head and $h(x)$ the random perturbation about the mean. Similarly, $F$ represents the constant mean value of $lnT$ and $f(x)$ the perturbation in $lnT$ about the mean. Both $h(x)$ and $f(x)$ are mean-zero stochastic processes. Substituting (4) into (2) leads to

$$\left[\frac{d^2H}{dx^2} + \frac{d^2h}{dx^2}\right] \;+\; \left[\frac{dF}{dx} + \frac{df}{dx}\right] \cdot \left[\frac{dH}{dx} + \frac{dh}{dx}\right] \;=\; 0. \tag{5}$$

Taking the expectation of (5) to obtain the mean flow equation and subtracting that from (5) yields a non-linear equation governing the perturbations in flow,

$$\frac{d^2h}{dx^2} - J\frac{df}{dx} + \frac{df}{dx}\frac{dh}{dx} - E\left[\frac{df}{dx}\frac{dh}{dx}\right] = 0 \tag{6}$$

where $J \equiv -dH/dx$ is the mean hydraulic gradient and $E[\ ]$ denotes expectation.

In order to simplify the mathematics and obtain a linear equation, the terms involving products of perturbations are neglected; it is assumed that the deviation of the perturbation product terms from their mean value is negligibly small relative to the other terms in (6). Thus, the perturbation equation (6) is reduced to

$$\frac{d^2h}{dx^2} - J\frac{df}{dx} = 0. \tag{7}$$

The remaining mean-zero, statistically homogeneous stochastic processes ($f$ and $h$) are then represented by Fourier-Stieltjes integrals [*see eg. Vanmarke*, 1984] (also referred to as the spectral representation theorem, [*Lumley and Panofsky*, 1964])

$$f(x) = \int_{-\infty}^{+\infty} e^{i\omega x}\, dZ_f(\omega) \tag{8}$$

$$h(x) = \int_{-\infty}^{+\infty} e^{i\omega x}\, dZ_h(\omega) \tag{9}$$

where $i$ is the complex constant, $i \equiv \sqrt{-1}$, and the $dZ(\omega)$ terms are complex Fourier amplitudes of the spatial fluctuations. After substituting (8) and (9) into (7) and differentiating, the uniqueness property of the spectral representation is used [see *Bakr et al*, 1978] to obtain

$$dZ_h(\omega) = \frac{-Ji}{\omega}\, dZ_f(\omega). \tag{10}$$

Further properties of the spectral representation theorem (which are discussed in the next section) are used to transform (10) into

$$S_{hh}(\omega) = \frac{J^2}{\omega^2} S_{ff}(\omega) \tag{11}$$

which relates the head spectrum, $S_{hh}(\omega)$, to the $lnT$ spectrum, $S_{ff}(\omega)$. We'll use this one-dimensional spectral relationship as a starting point for describing the cosimulation algorithm.

## COSIMULATION ALGORITHM IN ONE-DIMENSION

The algorithm for generating the random fields of head and $lnT$ is based on a discretized representation of the head and $lnT$ spectra. The spectrum of a process is a frequency domain representation which describes how the variability of the process is distributed over different spatial frequencies. The frequencies are inversely related to spatial separation distances or lags. For example, the spectral density at high frequencies describes how much of the variance is associated with short-range fluctuations of the process. The degree to which points separated by different lags are correlated is described by the covariance function, $R(\xi)$, which is related to the spectrum or spectral density function by the Fourier transform relationship

$$R(\xi) = \int_{-\infty}^{+\infty} e^{i\omega\xi} S(\omega) d\omega \tag{12}$$

where $\xi$ is the lag or separation distance. In generating the random fields, we make use of the spectral representation theorem which states that if $f(x)$ is a real, $2^{nd}$ order stationary, mean-zero stochastic process, then there exists a unique, with probability one, complex stochastic process, $Z_f(\omega)$, such that $f(x)$ can be represented by the Fourier–Stieltjes integral

$$f(x) = \int_{-\infty}^{+\infty} e^{i\omega x} dZ_f(\omega). \tag{13}$$

The $dZ_f(\omega)$ represent complex Fourier amplitudes of the fluctuations which are related to the spectrum of the $f$ process by

$$E\left[dZ_f(\omega) dZ_f^*(\omega')\right] = \begin{cases} 0 & \omega \neq \omega' \\ S_{ff}(\omega) d\omega = dF_f(\omega) & \omega = \omega' \end{cases} \tag{14}$$

where * represents complex conjugate, $S_{ff}(\omega)$ is the spectral density function of the $f$ process and $dF_f(\omega)$ is the integrated spectrum or spectral distribution function. We can generate the random field $f(x)$ having covariance behavior $R(\xi)$ given by (12), by *constructing* the stochastic process $dZ_f(\omega)$ in (13) such that (14) holds. This is accomplished in the following manner:

The Fourier–Stieltjes integral representation (13) is expanded and written as

$$f(x) = \int_{-\infty}^{0} e^{i\omega x} dZ_f(\omega) + \int_{0}^{\infty} e^{i\omega x} dZ_f(\omega) \tag{15a}$$

$$f(x) \equiv Y_0(x) + Y_1(x). \tag{15b}$$

Substituting $-\alpha = \omega$ into $Y_0(x)$ gives

$$Y_0(x) = \int_{\infty}^{0} e^{-i\alpha x}\, dZ_f(-\alpha). \tag{16}$$

In order to insure that $f(x)$ is a real process, we construct the $dZ_f$ process such that $dZ_f(-\alpha) = -dZ_f^*(\alpha)$; then (16) becomes

$$Y_0(x) = \int_{0}^{\infty} e^{-i\alpha x}\, dZ_f^*(\alpha) = Y_1^*(x).$$

(15b) can then be written as

$$\begin{aligned} f(x) &= Y_1^*(x) + Y_1(x) \\ &= 2Re\, Y_1(x) \end{aligned}$$

and thus (13) becomes

$$f(x) = 2Re \int_{0}^{\infty} e^{i\omega x}\, dZ_f(\omega). \tag{17}$$

The representation (17) is then approximated by integrating over a finite number of frequencies up to some maximum frequency, $\Omega$. To implement this on the computer, we discretize frequency space and replace the integral in (17) with the summation

$$f(x) \cong 2Re \sum_{k=0}^{M-1} e^{i\omega_k x} dZ_f(\omega_k) \tag{18}$$

where $M$ is the number of harmonics, $\omega_k = (k + \frac{1}{2})\Delta\omega$, and $\Delta\omega = \Omega/M$. The $dZ_f(\omega_k)$ process is constructed by setting

$$\begin{aligned} dZ_f(\omega_k) &\equiv \sqrt{dF_f(\omega_k)}\ (U_k + iV_k) \\ &\cong \sqrt{S_{ff}(\omega_k)\Delta\omega}\ (U_k + iV_k) \end{aligned} \tag{19}$$

where the $U_k$ and $V_k$ are mean-zero, variance $\frac{1}{2}$ uncorrelated random variables. Substituting (19) and its complex conjugate into (14) and taking expected values gives

$$E\left[dZ_f(\omega_j) dZ_f^*(\omega_k)\right] \cong \begin{cases} 0 & j \neq k \\ S_{ff}(\omega_k)\Delta\omega & j = k. \end{cases}$$

Thus, $f(x) \equiv lnT$ can be represented approximately using (18) and (19) as

$$f(x) \;\cong\; 2Re \sum_{k=0}^{M-1} e^{i\omega_k x} \sqrt{S_{ff}(\omega_k)\Delta\omega}\; (U_k + iV_k). \tag{20}$$

Similarly, the head process, $h(x)$, can be represented by discretizing (9) in a manner analgous to (18) as

$$h(x) \;\cong\; 2Re \sum_{k=0}^{M-1} e^{i\omega_k x} dZ_h(\omega_k)$$

where $dZ_h(\omega_k)$ is given by (10) in terms of $dZ_f(\omega_k)$. Finally, we substitute (19) into (10) to obtain

$$h(x) \;\cong\; 2Re \sum_{k=0}^{M-1} e^{i\omega_k x} \frac{-Ji}{\omega_k} \sqrt{S_{ff}(\omega_k)\Delta\omega}\; (U_k + iV_k). \tag{21}$$

For each location $x$, the representations (20) and (21) involve the summation of $M$ complex numbers where the $k$th term of the sum is the product of the terms to the right of the $\sum$ in (20) and (21). The $U_k$ and $V_k$ terms are obtained by calling a random number generator routine which is usually provided as an intrinsic function on most Fortran compilers †. The cosimulation of the head and $lnT$ fields is accomplished by using the same random numbers, $U_k$ and $V_k$, in both (20) and (21). Thus, for each $k$, the product of the exponential term, the radical, and the random numbers is the same for both (20) and (21). The computational work is therefore reduced by generating $f(x)$ and $h(x)$ simultaneously by computing this product, using it in (20) and using $\frac{-Ji}{\omega_k}$ times the same product in (21). The cosimulation procedure is illustrated schematically in Figure 1.

In the next section, we use the representations (20) and (21) to theoretically test the viability of the cosimulation approach.

† Most Fortran callable random number generator routines output numbers in the range [0, 1] which yields a random process with mean 1/2 and variance 1/12. To obtain a mean-zero, variance 1/2 random process, the output on [0, 1] must be mapped into the range $[-\sqrt{6}/2, +\sqrt{6}/2]$.

## THEORETICAL ANALYSIS IN ONE-DIMENSION

The first test for success or failure of the method is to determine whether the cogenerated head and $lnT$ fields exhibit the proper statistical behavior, both individually and jointly. Secondly, if the technique is to be of any practical utility, it must be shown that the cogenerated fields satisfy the mass conservation principle. An analytical analysis of this second requirement involves formidable mathematical difficulties, therefore, the theoretical "mass balance" analysis is restricted to determining whether the discrete representations (20) and (21) satisfy the governing differential equation (the perturbation equation (7), a non-mass flow equation). The mass conservation issue is addressed, however, via numerical flux calculations. In this section, we show how the cogenerated fields, given by the representations (20) and (21), theoretically preserve the proper statistical relationships and satisfy the governing differential equation.

### Covariance Analysis

To determine whether the proper statistical structure is preserved, we examine the auto- and cross-covariance behavior of the cogenerated discrete fields as represented by (20) and (21) and compare that to the theoretical behavior for continuous processes. The theoretical auto-covariance of the head and $lnT$ fields, $R_{hh}(\xi)$ and $R_{ff}(\xi)$ respectively, is obtained by Fourier transforming the head and $lnT$ spectra respectively as in (12). Similarly, the theoretical cross–covariance function is obtained by replacing $S(\omega)$ in (12) with the cross–spectrum, $S_{fh}(\omega)$ and carrying out the integration. The analysis for the auto– and cross–covariances is redundant, so we illustrate it only for the auto-covariance. The auto-covariance, $R_{ff}(\xi)$, is defined in terms of expected values as

$$R_{ff}(\xi) \;=\; E\big[f(x)f(\tau)\big], \qquad \tau = x + \xi$$

where $x$ is a coordinate location and $\xi$ is the lag or separation distance. Using the

form (20), $R_{ff}(\xi)$ is written as

$$E\Big[f(x)f(\tau)\Big] \;\cong\; E\Bigg[\,2Re\sum_{k=0}^{M\text{-}1} e^{i\omega_k x}\sqrt{S_{ff}(\omega_k)\Delta\omega}\;(U_k+iV_k)\cdot \\ 2Re\sum_{j=0}^{M\text{-}1} e^{i\omega_j \tau}\sqrt{S_{ff}(\omega_j)\Delta\omega}\;(U_j+iV_j)\Bigg] \tag{22}$$

and we note that terms involving complex quantities can be combined as

$$e^{i\omega_k x}\;(U_k+iV_k) \;=\; \Big\{U_k cos(\omega_k x) - V_k sin(\omega_k x)\Big\} \;+\; i\Big\{U_k sin(\omega_k x) + V_k cos(\omega_k x)\Big\}$$

so that (22) becomes, after discarding the imaginary parts,

$$E\Big[f(x)f(\tau)\Big] \;\cong\; E\Bigg[\,4\sum_{k=0}^{M\text{-}1}\sum_{j=0}^{M\text{-}1}\sqrt{S_{ff}(\omega_k)}\sqrt{S_{ff}(\omega_j)}\,\Delta\omega\cdot \\ \Big\{U_k cos(\omega_k x) - V_k sin(\omega_k x)\Big\}\cdot\Big\{U_j cos(\omega_j \tau) - V_j sin(\omega_j \tau)\Big\}\Bigg]. \tag{23}$$

The only random components in (23) are the $U$'s and $V$'s inside the braces; taking the expectation on the right hand side of (23) involves the evaluation of

$$E\Big[U_k U_j\Big] \;=\; E\Big[V_k V_j\Big] \;=\; \begin{cases} 0 & j \neq k \\ 1/2 & j = k \end{cases}$$
$$E\Big[U_k V_j\Big] \;=\; 0 \text{ for all } j,k.$$

where $U_k = U_j$ and $V_k = V_j$, if $k = j$. The expectation of the terms in braces in (23) becomes

$$E\{\;\}\{\;\} \;=\; \frac{1}{2}\Big[cos(\omega_k x)cos(\omega_k \tau) \;+\; sin(\omega_k x)sin(\omega_k \tau)\Big] \;=\; \frac{1}{2}cos(\omega_k \xi)$$

where the last reduction occurs after replacing $\tau$ with $x+\xi$ and using trigonometric identities. Thus, the theoretical auto-covariance of $lnT$, derived from the discrete representation (20), is given by

$$E\Big[f(x)f(\tau)\Big] \;\equiv\; R_{ff}(\xi) \;\cong\; 2\sum_{k=0}^{M\text{-}1} cos(\omega_k \xi) S_{ff}(\omega_k)\Delta\omega \tag{24}$$

which is the discrete form of (12). Thus the covariance is preserved in its discrete form. The cross-covariance, $R_{fh}(\xi)$, can similarly be shown to be preserved by replacing $S_{ff}(\omega_j)$ with $S_{hh}(\omega_j)$ in equation (22) and carrying out the rest of the analysis in exactly the same way.

Mass Balance Analysis

The second test is to determine if the cogenerated head and $lnT$ fields satisfy the governing continuity equation. The representations (20) and (21) are developed from the perturbation equation (7) which differs from the original governing differential equation (1) due to the missing perturbation product terms. The assumption made is that the perturbation products are negligibly small, so that satisfying (7) is nearly equivalent to satisfying the governing equation (1) written in terms of $lnT$, as in (2). Note that (7) is not really a "mass" balance equation since it is written in terms of the logarithm of transmissivity. Here we check to see if the perturbation equation is satisfied by substituting the discrete representations (20) and (21) into (7); to avoid difficulties associated with differentiating those expressions (which involve the *Real* operator), we write

$$2\,Re\sum_{k=0}^{M-1} \quad = \quad \sum_{k=-M}^{M-1}.$$

Then using (20), the term $-J\frac{df}{dx}$ becomes

$$-J\frac{df}{dx} = -2J\sum_{k=-M}^{M-1} e^{i\omega_k x} i\omega_k \sqrt{S_{ff}(\omega_k)\Delta\omega}\,(U_k + iV_k) \tag{25}$$

and via (21) the $d^2h/dx^2$ term becomes

$$\begin{aligned}\frac{d^2h}{dx^2} &= 2\sum_{k=-M}^{M-1} e^{i\omega_k x}(i\omega_k)^2 \frac{-Ji}{\omega_k}\sqrt{S_{ff}(\omega_k)\Delta\omega}\,(U_k + iV_k) \\ &= 2J\sum_{k=-M}^{M-1} e^{i\omega_k x} i\omega_k \sqrt{S_{ff}(\omega_k)\Delta\omega}\,(U_k + iV_k) \end{aligned} \tag{26}$$

where the derivative operator has been carried through the summation terms. Adding the two components (25) and (26) together shows that the representations (20) and (21) satisfy the perturbation equation exactly.

In the next section, we cogenerate discrete one-dimensional random fields of head and $lnT$ and analyse those fields numerically to verify the preceding results. In addition, we perform a Darcy flux analysis to address the mass conservation issue.

The $f(x)$ and $h(x)$ representations (20) and (21) are written for continuous functions of space. In order to take advantage of Fast Fourier Transform (FFT) algorithms, we cogenerate the $f$ and $h$ fields at evenly spaced intervals and write (20) as

$$f(j\Delta x) = 2Re \sum_{k=0}^{M-1} e^{i\omega_k j\Delta x} \sqrt{S_{ff}(\omega_k)\Delta\omega}\ (U_k + iV_k) \tag{27}$$

where the discretization interval $\Delta x \equiv 2\pi/\Omega$ and $j = 0, 1 \ldots M-1$. The maximum frequency, $\Omega$, must be chosen sufficiently large to capture practically all of the information under the spectrum. Furthermore, the frequency spacing, $\Delta\omega \equiv \Omega/M$, must be chosen sufficiently small ($M$ sufficiently large) to accurately capture rapid changes in the spectrum. Deciding what values to use for $\Omega$ and $\Delta\omega$ is an important practical consideration that effects whether or not the resulting field will possess the proper statistical structure. If $\Omega$ is too small the simulated field will not achieve its prescribed variance, and if $\Delta\omega$ is too large the covariance structure at large lags will not be preserved. In our case, two fields are being generated simultaneously ($M$ and $\Omega$ are the same for both), therefore the choice for $\Omega$ and $\Delta\omega$ must satisfy the minimum requirements necessary to adequately represent both the $f$ and $h$ spectra. Furthermore, since these fields are being cogenerated, $\Omega$ and $\Delta\omega$ must also satisfy the minimum requirements of the cross-spectral density, $S_{fh}(\omega)$. An appropriate choice for $\Omega$ can be determined by examining the spectral distribution function (the integrated spectrum) and choosing the $\omega_{max} \equiv \Omega$ which contains 98 or 99 percent of the frequency content. An appropriate value for $\Delta\omega$ can be determined by examining the large lag covariance behavior of the generated field. If the discrete covariance deviates from the theoretical behavior at large lags, it may be an indication that $\Delta\omega$ is too large to adequately represent the low frequency end of the spectrum where most of the rapid changes occur (see Figure 2).

It is also important to note that the finite Fourier Transform introduces anti-periodicities in the output sequence so that only half the transformed array should

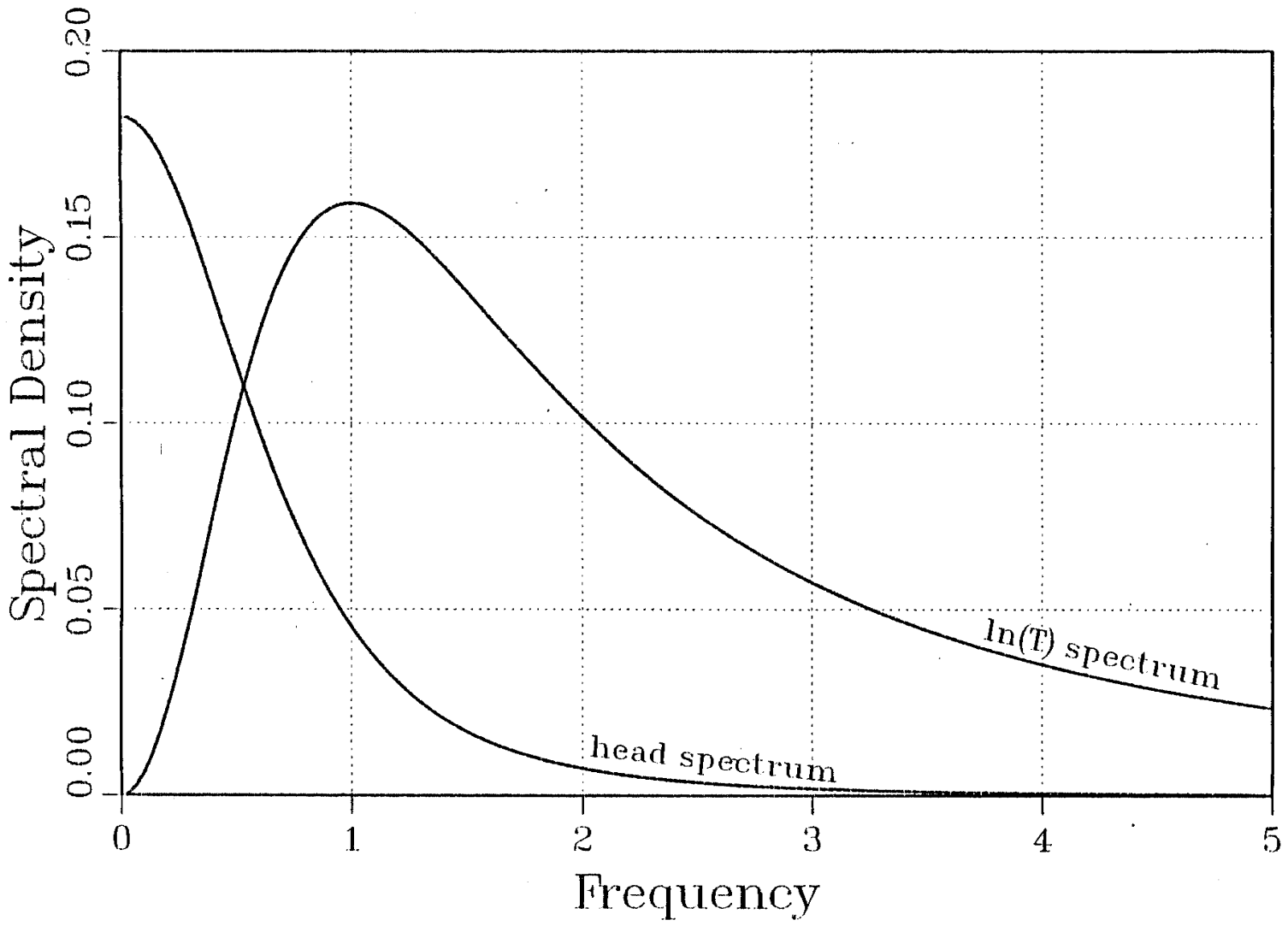



Figure 2. Spectral density functions for one-dimensional head and $lnT$ processes (corresponding to $lnT$ hole covariance function).

be used; in general this does not pose any problems, but in two-dimensions it means that 3/4ths of the transformed field must be thrown away. Thus, two-dimensional simulations may require very large input arrays which grow in size as the square of the number of generation points in the grid. A derivation showing why the anti-periodic behavior occurs is given in Appendix A (see also *McKay*, [1988]).

The one-dimensional fields are analysed as follows. Covariance analysis is used to estimate the auto and cross-covariance behavior of the discrete fields and compare it with the theoretical covariance functions. The governing perturbation equation is discretized and the cosimulated fields are analysed to determine if the perturbation equation is satisfied. Darcy flux calculations are made to determine whether the algorithm preserves mass, and finally, the cosimulated head field is compared to a deterministic head solution corresponding to the same parameter ($lnT$ ) field.

Covariance Analysis

Two different $lnT$ covariance functions are used in the discrete field analyses. The first one used in these simulations is referred to here as the "hole" function and is given by (3). The corresponding spectral density function,

$$S_{ff}(\omega) = \frac{2\sigma_f^2}{\pi} \frac{\omega^2 l^3}{(1+\omega^2 l^2)^2}, \tag{28}$$

is used together with (11) to obtain the head spectrum

$$S_{hh}(\omega) = \frac{2J^2\sigma_f^2}{\pi} \frac{l^3}{(1+\omega^2 l^2)^2}. \tag{29}$$

The theoretical head covariance is obtained by subsubstituting (29) into (12) and integrating to give

$$R_{hh}(\xi) = J^2\sigma_f^2 l^2 (1+|\xi|/l)\, e^{-|\xi|/l}. \tag{30}$$

The theoretical cross-spectrum, $S_{fh}(\omega)$, is derived by multiplying both sides of (10) by $dZ_f^*(\omega)$, taking expected values and using (14) to obtain

$$\begin{aligned} E\big[dZ_h(\omega)dZ_f^*(\omega)\big] &= E\Big[\frac{-Ji}{\omega} dZ_f(\omega)dZ_f^*(\omega)\Big] \\ S_{fh}(\omega) &= \frac{-Ji}{\omega} S_{ff}(\omega). \end{aligned} \tag{31}$$

The theoretical cross-covariance function is derived by substituting (31) into (12) and integrating to give [*Gutjahr et al*, 1978]

$$R_{fh}(\xi) \;=\; \sigma_f^2 J \xi \, e^{-|\xi|/l}. \tag{32}$$

Multiple sets of discrete random fields are cogenerated and the auto- and cross-covariance statistics are calculated over the ensemble of fields. These discrete covariances are compared with the theoretical covariance functions given by (3), (30) and (32) in Figure 3. The results show excellent agreement verifying that the method preserves the proper statistical relationships. The discrete covariances were calculated (see eg *Journel and Huijbregts*, [1978]) over an ensemble of cogenerated random fields spanning 10,000 correlation lengths of the $lnT$ process (60 simulations, each 166 correlation lengths long). The correlation length is a rough measure of the average distance over which the values of the process are correlated and is defined here as the parameter "$l$" in the covariance model for $lnT$ , equation (3). Spectral estimates of these discrete fields were not made.

Perturbation Analysis

A finite difference approximation is applied to the perturbation equation (7) as

$$\frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2} \;-\; J \frac{f_{i+1} - f_{i-1}}{2\Delta x} \;=\; 0 \tag{33}$$

where the subscript $i$ indexes the nodal location in the one-dimensional $f$ and $h$ arrays. Figures 4 and 5 show the results of perturbation analyses conducted on four sets of cosimulated head and $lnT$ random fields. The fields are generated at discretization levels of 10, 20, 40, and 80 output points per correlation length of the $lnT$ process. This was done to determine what level of discretization is needed to obtain convergence. The thick dark lines represent the residuals of equation (33), whose magnitude should be small relative to the individual terms in that equation. The dashed and thin-solid lines represent the second derivative of head and gradient times first derivative of $lnT$ terms respectively. The values are normalized by an arbitrary scale factor

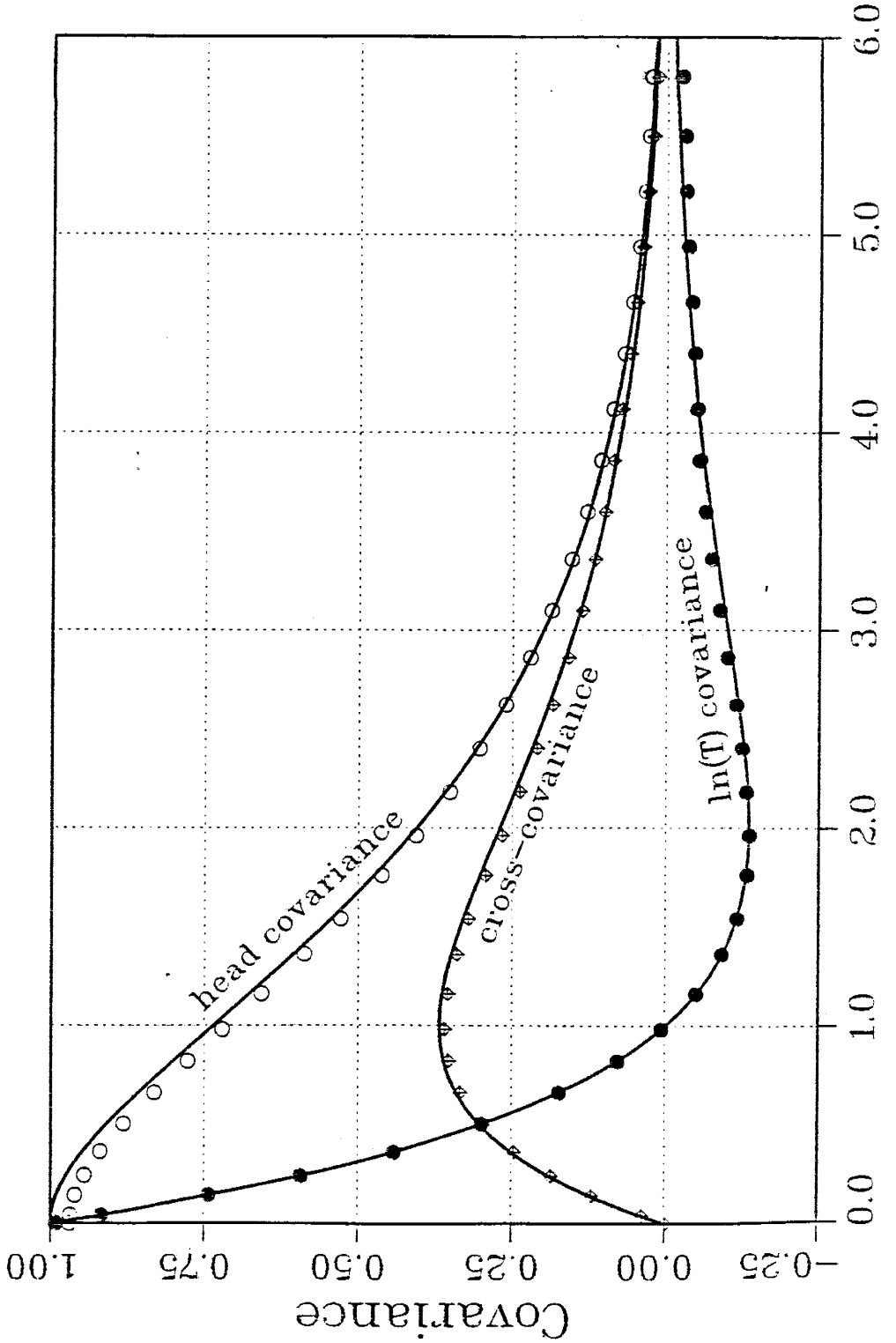



Figure 3. Theoretical (solid lines) and experimental (symbols) auto- and cross-covariance functions corresponding to hole function fields with $\sigma_f^2$=1.

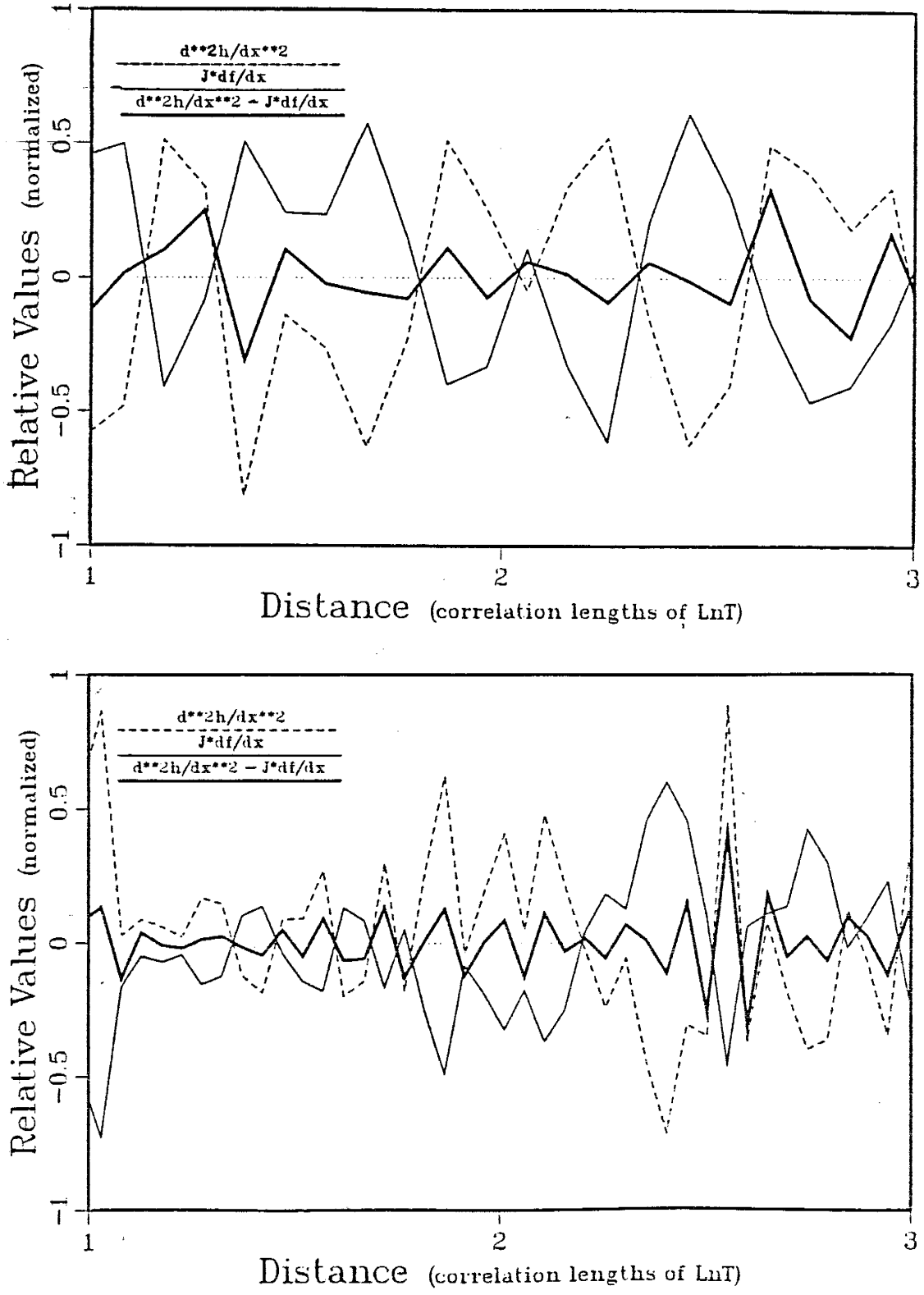



Figure 4. Perturbation results for hole fields generated at 10 points per correlation length (top) and 20 points per correlation length (bottom). Thick dark line is the residual. $\sigma_f^2 = 1$.

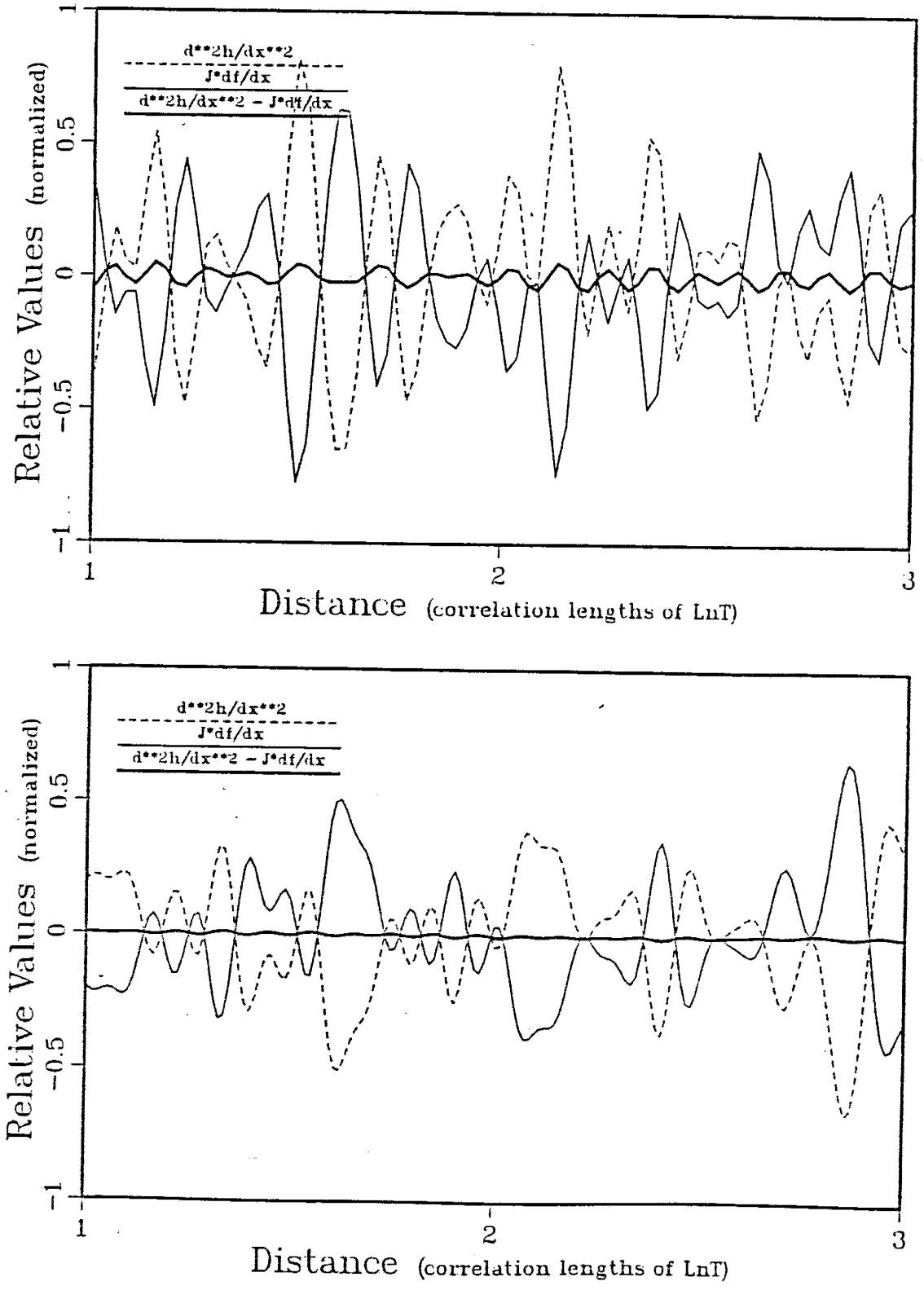



Figure 5. Perturbation results for hole fields generated at 40 points per correlation length (top) and 80 points per correlation length (bottom). Thick dark line is the residual. $\sigma_f^2 = 1$.

in order to show their relative magnitudes in the range of [−1, +1]. These plots indicate that the perturbation equation is satisfied only at extreme levels of discretization whereupon the cosimulation procedure loses its practical utility. It should be noted, however, that $\sigma_f^2$ is set equal to 1.0 in these cosimulations; perturbation analysis generally assumes $\sigma_f^2$ is small ($\sigma_f^2 \ll 1$) [*Gutjahr et al*, 1978; *Smith and Freeze*, 1979; *de Marsily*, 1986]. Figure 6 is a plot of cosimulated head perturbations and $lnT$ fields at a discretization level of 40 points per correlation length. The fields are mean-zero and plotted as they are output from the code (no normalization used). This figure shows the noisy behavior of the $lnT$ field and the smoothness of the head field. The $lnT$ field must be output at an extremely fine spacing in order to accurately compute numerical derivatives of this process.

Differentiability

These somewhat discouraging results leads to a closer examination of the theoretical $lnT$ covariance function to obtain a better understanding of, and hopefully resolve, this problem. The spectral representation of the derivative of the $lnT$ process is given by

$$f'(x) = \int_{-\infty}^{+\infty} e^{i\omega x}\, dZ_{f'}(\omega) = \frac{d}{dx} \int_{-\infty}^{+\infty} e^{i\omega x}\, dZ_f(\omega) = \int_{-\infty}^{+\infty} e^{i\omega x}\, i\omega\, dZ_f(\omega), \quad (34)$$

thus, $dZ_{f'}(\omega) = i\omega\, dZ_f(\omega)$. The variance of the derivative process can be computed by integrating its spectrum, which is given by

$$\begin{aligned} E\left[ dZ_{f'}(\omega)\, dZ^*_{f'}(\omega) \right] &= E\left[ (i\omega)\, dZ_f(\omega)\ (-i\omega)\, dZ^*_f(\omega) \right] \\ &= \omega^2\, E\left[ |dZ_f(\omega)|^2 \right] \\ &= \omega^2 S_f(\omega)\, d\omega \qquad (35) \\ &= S_{f'}(\omega)\, d\omega. \end{aligned}$$

The variance of $f'(x)$ is then given by substituting (28) into (35) and integrating :

$$\sigma^2_{f'} = \int_{-\infty}^{+\infty} S_{f'}(\omega)\, d\omega = \frac{2\sigma_f^2}{\pi} \int_{-\infty}^{+\infty} \frac{\omega^4 l^3}{(1+\omega^2 l^2)^2}\, d\omega = \infty. \quad (36)$$

This result shows that the variance of the derivative process of (3) is infinite, therefore this $lnT$ process is not differentiable. Because the discrete field is generated from a representation of the spectrum containing a finite number of frequencies, the corresponding discrete process is actually differentiable, but only at an impractical level of resolution as illustrated in Figures 4 and 5.

The Bell Covariance Function

As a consequence of the foregoing analysis, consider another covariance model for the $lnT$ process which yields stationary heads and is differentiable. A modified form of the hole function defined by

$$R_{ff}(\xi) = \sigma_f^2\left(1 - \frac{\xi^2}{l^2}\right)e^{-\frac{\xi^2}{2l^2}} \tag{37}$$

which we refer to here as the "bell" covariance function, satisfies both of these conditions. This function gives rise to the spectrum–covariance pairs shown in Table 1. The variance of the derivative process, given by $R_{f'}(0) = \frac{3\sigma_f^2}{l^2}$, varies inversely with the square of the correlation parameter, $l^2$, while the variance of the head process, $R_{hh}(0) = \sigma_f^2\, l^2\, J^2$, varies directly with $l^2$. All of the bell field simulations described here are carried out with the $l = 1$ and $\sigma_f^2 = 1$.

Table 1. Spectrum/covariance pairs for the "bell" function.

| Random Process | Covariance Function | Spectral Density |
|---|---|---|
| $f(x)$ | $\sigma_f^2\left(1 - \frac{x^2}{l^2}\right)e^{-\frac{x^2}{2l^2}}$ | $\frac{\sigma_f^2 l^3 \omega^2}{\sqrt{2\pi}} e^{-\frac{l^2\omega^2}{2}}$ |
| $h(x)$ | $\sigma_f^2 J^2 l^2 e^{-\frac{x^2}{2l^2}}$ | $\frac{\sigma_f^2 J^2 l^3}{\sqrt{2\pi}} e^{-\frac{l^3\omega^2}{2}}$ |
| $fh(x)$ | $J\sigma_f^2 x e^{-\frac{x^2}{2l^2}}$ | $\frac{J\sigma_f^2 l^3 i\omega}{\sqrt{2\pi}} e^{-\frac{l^2\omega^2}{2}}$ |
| $f'(x)$ | $\sigma_f^2\left(\frac{x^4}{l^6} - \frac{6x^2}{l^4} + \frac{3}{l^2}\right)e^{-\frac{x^2}{2l^2}}$ | $\frac{\sigma_f^2 l^3 \omega^4}{\sqrt{2\pi}} e^{-\frac{l^2\omega^2}{2}}$ |

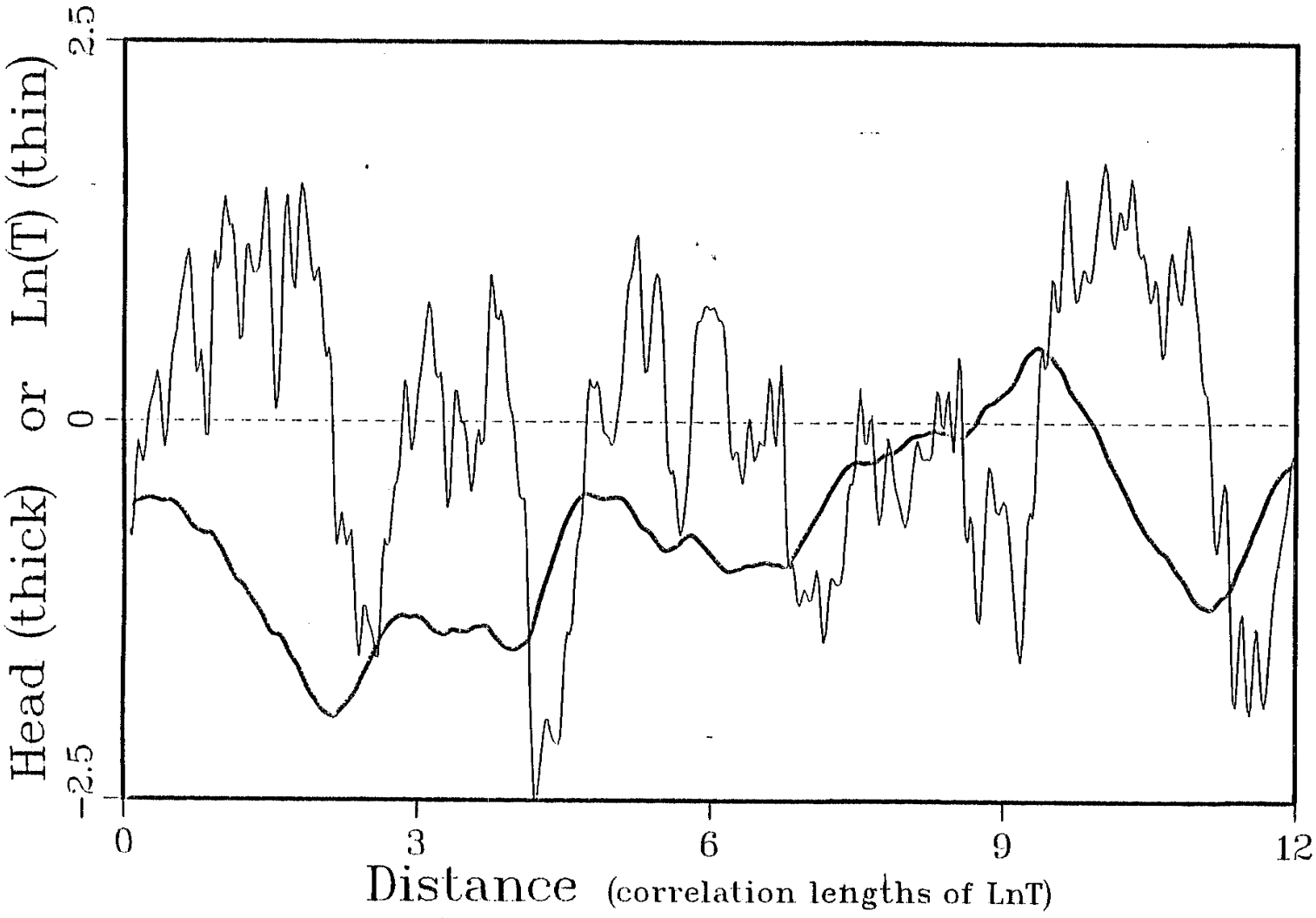



Figure 6. A hole function $lnT$ and corresponding head realization generated at 40 points per correlation length. $\sigma_f^2 = 1$.

The auto- and cross-covariance statistics are calculated on the cosimulated output fields as before and compared to their theoretical functions (Table 1). The results are plotted in Figure 7 which again shows excellent correspondence between the theoretical and experimental statistics. Results of a bell field perturbation analysis is shown in Figure 8 where, at a discretization level of 10 output points per correlation length, the residuals for this realization are negligibly small. Thus, the perturbation equation (33) is satisfied at a resonable level of discretization when the bell covariance function is used for the $lnT$ process. To contrast the bell fields with the hole fields, the cogenerated head and bell $lnT$ fields are plotted in Figure 9; the bell $lnT$ process is much smoother and consequently better suited for numerical differentiation than the $lnT$ field shown in Figure 6. Now that the perturbation equation has been adequately satisfied, the mass conservation issue can be addressed.

Mass Balance Analysis

An assessment of how well continuity is being maintained can be made by casting these results in the framework of a true mass balance analysis, ie. by calculating the Darcy flux imbalances in each grid block of a finite difference version of (1). The volumetric Darcy flux, $Q$, is calculated using

$$Q(x) \;=\; -T(x)\,\frac{d\phi(x)}{dx} \tag{38}$$

where $Q(x) = Q$ = constant for one-dimensional steady flow. The transmissivity, $T(x)$, and total hydraulic head, $\phi(x)$, are generated from

$$\begin{aligned} T(x) \;&=\; exp\Big\{F + f(x)\Big\} \\ \phi(x) \;&=\; H(x) + h(x), \end{aligned} \tag{39}$$

where $F$ is the mean of the $lnT$ process and $H(x)$ the mean hydraulic head. The mass balance errors are calculated as a percentage deviation from the mean flux, $q_m$, which is given by [*Gutjahr et al*, 1978]

$$q_m \;=\; e^F \left(1 - \frac{\sigma_f^2}{2}\right) J. \tag{40}$$

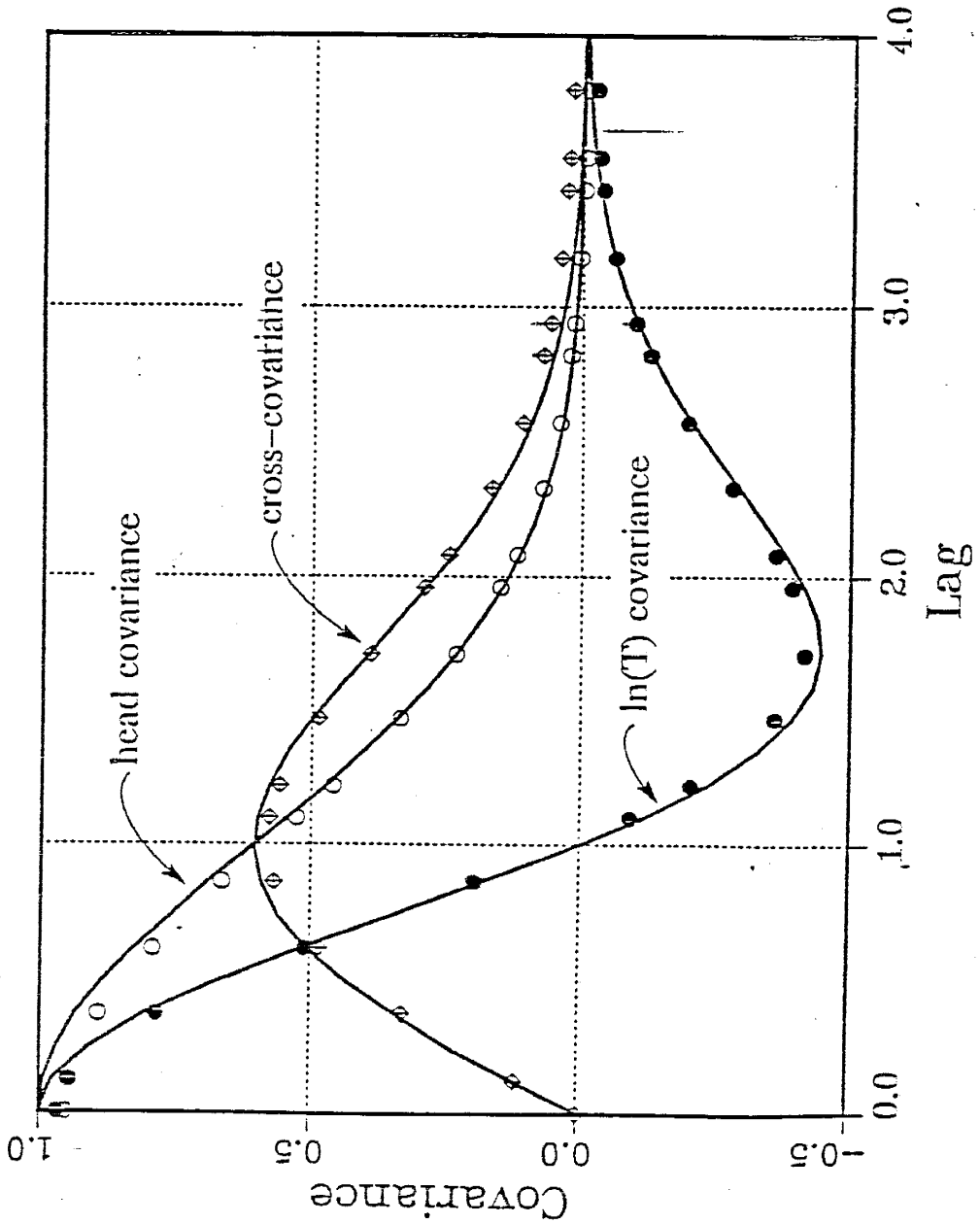



Figure 7. Theoretical (solid lines) and experimental (symbols) auto- and cross-covariance functions corresponding to bell function fields with $\sigma^2_f=1$.

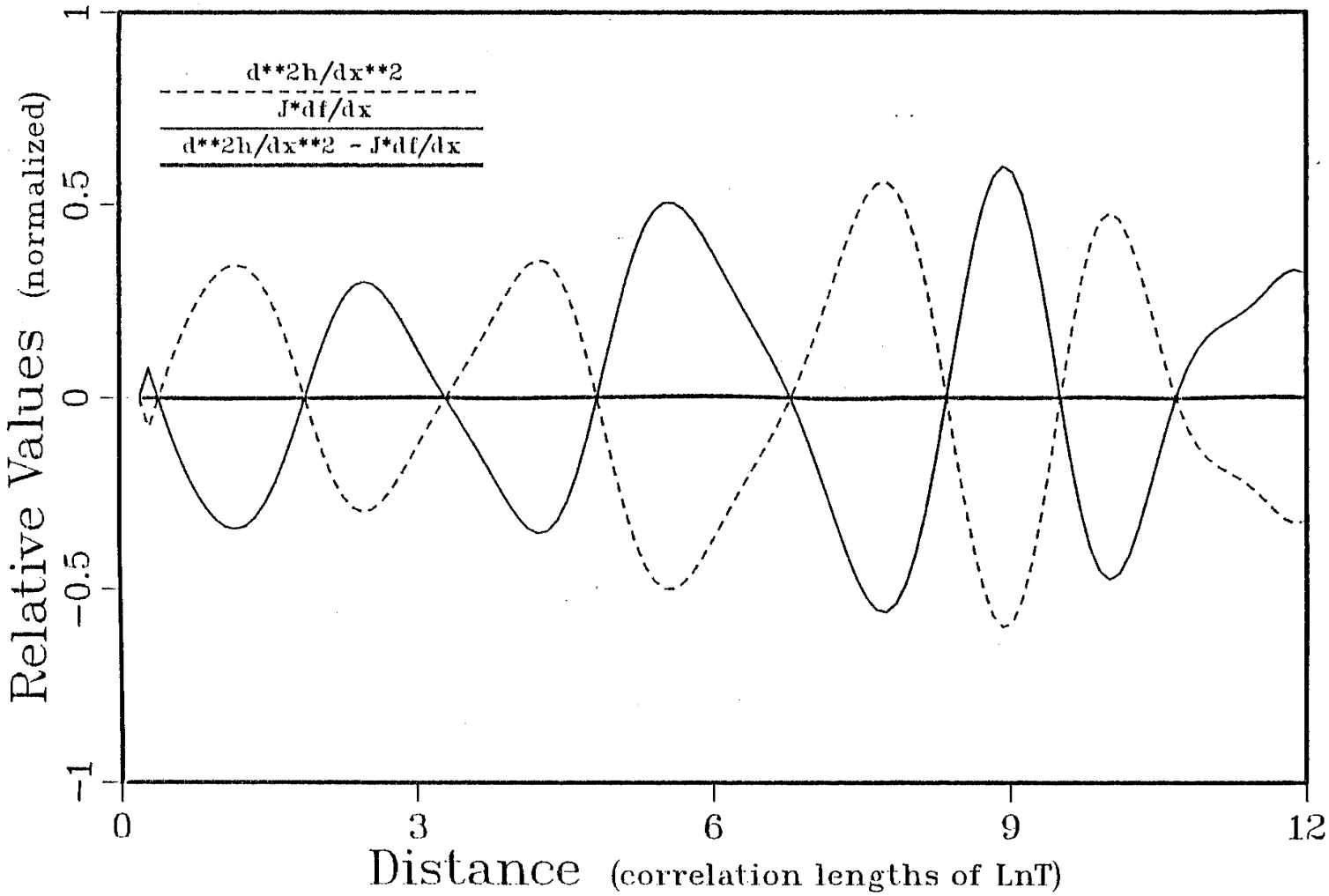



Figure 8. Perturbation results for bell fields generated at 10 points per correlation length. Thick dark line is the residual. $\sigma_f^2 = 1$.

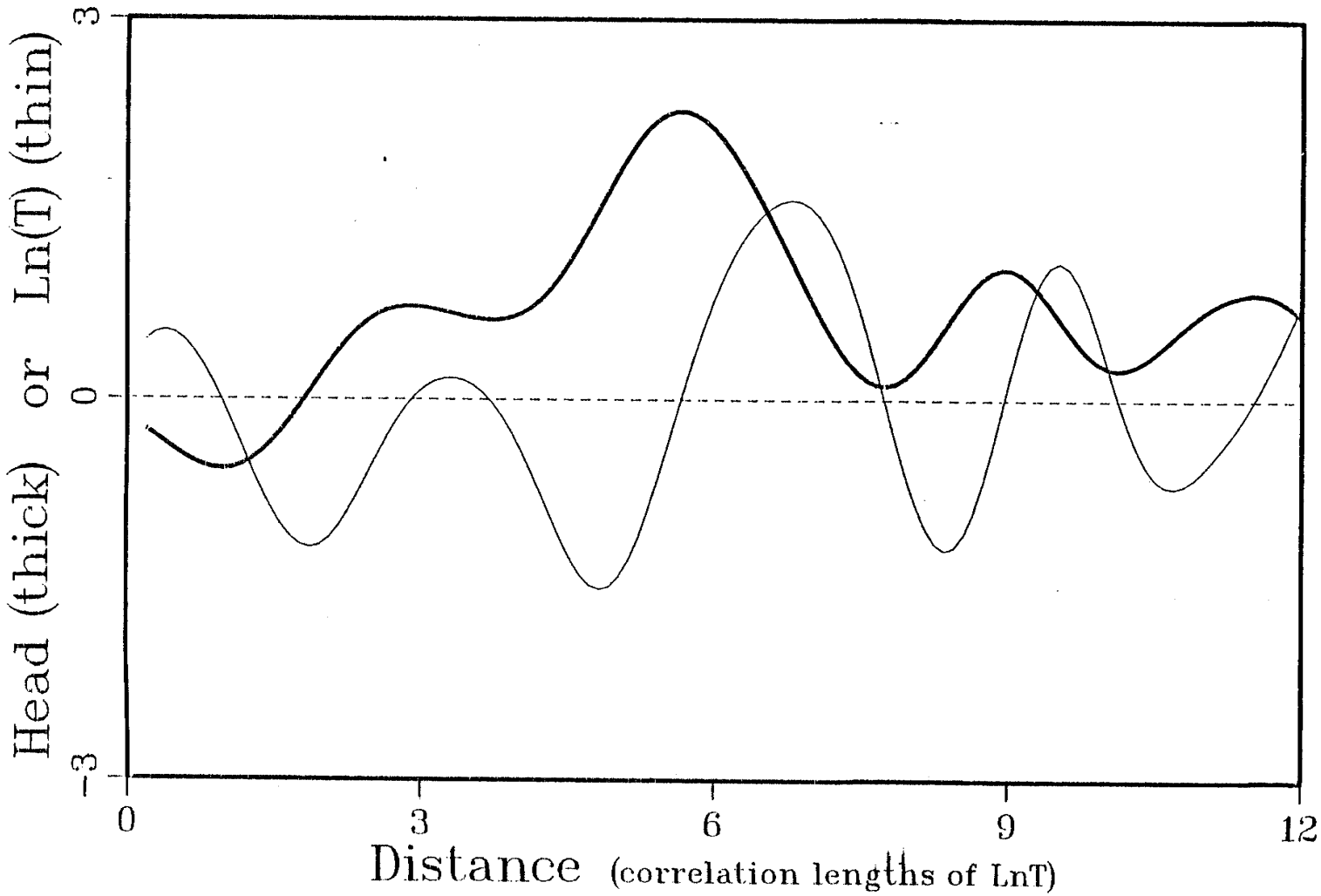



Figure 9. A bell function $lnT$ and corresponding head realization generated at 10 points per correlation length. $\sigma_f^2 = 1$. (Compare with Figure 6)

The magnitude of the flux errors varied greatly throughout a cogenerated field, remaining relatively small over most of the domain, but occasionally being interrupted by large mass balance deviations as illustrated in Figure 10. It is evident from this plot that mass is conserved globally (ie. the positive and negative mass imbalances sum to zero). The root mean-square error (RMSE), calculated over an entire realization, is used as a scalar measure of the average local mass imbalance. The magnitude of the RMSE is partially a function of the input variance, $\sigma_f^2$, and partially a function of the resolution at which the fields are represented and numerical derivatives estimated. Table 2 shows the results of several runs for different values of $\sigma_f^2$ and for different levels of resolution for a bell $lnT$ field. The statistics for each run are based on 40 simulations, each 6 correlation lengths long, or a total of about 250 correlation lengths of the $lnT$ process (the statistical estimates stabilized with this quantity of data). The results clearly show that a relatively high resolution grid and a low input variance are necessary for the algorithm to yield acceptable mass balance results. It should also be noted that these results are indicative of how well the linearized version of (1), equation (7), approximates the true equation (1). The local flux imbalances are plotted along with the head and $lnT$ fields as described in the following section.

Table 2. Mass balance results for bell fields generated at 10 and 20 points per correlation length; numbers are RMSE in %.

| Pts/Cl | $\sigma_f^2 = 0.1$ | $\sigma_f^2 = 0.25$ | $\sigma_f^2 = 0.5$ | $\sigma_f^2 = 1.0$ |
|---|---|---|---|---|
| 10 | **1.9** | 6.6 | 16 | **70** |
| 20 | 0.8 | 2.9 | 8.8 | **25** |

Comparison with a Deterministic Solution

To further aid the analysis, we compare the cogenerated results with a deterministic solution for the flow. The deterministic solution represents the solution to the "forward" problem where the head field is solved for, given the $lnT$ parameter

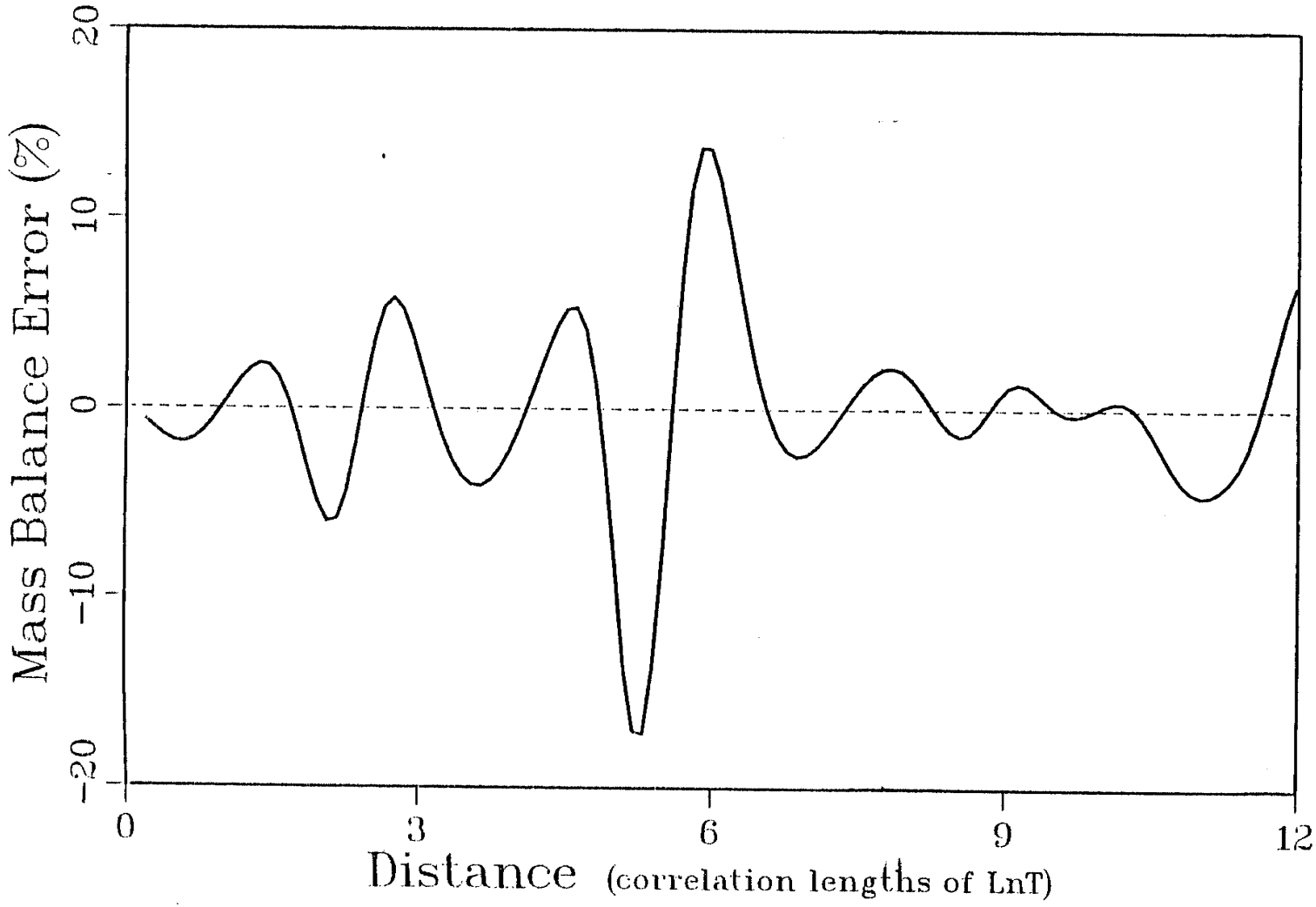



Figure 10. Local Darcy flux imbalances for one realization of cogenerated bell $lnT$ and head fields. This curve corresponds to the dotted curve in Figure 12. ($\sigma_f^2$= 0.25, 10 pts/cl)

distribution and boundary conditions as knowns. Standard numerical flow simulation models may be used for this purpose. Only the boundary conditions present a problem since the cogenerated solution is developed in connection with a mean hydraulic head gradient arising from boundary conditions essentially placed at infinity. To overcome this, we chose $\phi_c(0)$ and $\phi_c(L)$, the cogenerated total hydraulic head values at $x$=0 and $x$=$L$, as Dirichlet boundary conditions for the deterministic case. Although this is not entirely consistent with the theory, which is based on an infinite domain conceptual model, it was felt that some insight might be gained from this approach.

Three different cases are examined, each one representing a sample run corresponding to one of the bold-faced entries in Table 2. These cases were chosen to illustrate the effects of a reduction in the input variance and to show what happens with increased resolution. The results for the deterministic solution and the cogenerated solution are plotted together with the parameter field for the perturbations in $lnT$ and the mass balance errors in Figures 11, 12 and 13. Figure 11 represents the case of high variance, low resolution ($\sigma_f^2$=1.0, 10 pts/cl); Figure 12 represents the case of low variance, low resolution ($\sigma_f^2$=0.25, 10 pts/cl); Figure 13 represents the case of high variance, high resolution ($\sigma_f^2$=1.0, 20 pts/cl). In all of these cases, the cogenerated solution generally mimics the behavior of the deterministic solution and displays the proper character in that heads are correlated over greater distances than $lnT$ , the hydraulic gradient steepens when $f(x)$ dips below zero (low transmissivity areas), and the gradient flattens in areas of high transmissivity. The slanted dashed lines in these Figures represent the mean hydraulic head and the shaded zones above and below that line represent the local mass balance errors. The magnitude of the mass balance error is scaled arbitrarily and is therefore not associated with the ordinate values on the plot; only the relative errors between locations are important. However, the same scale factor was used on each of the three plots in Figures 11–13.

As demonstrated by the deterministic solutions (thick dark curves), the one-dimensional steady flow head solution must be a monotonically decreasing function in the direction of flow, otherwise local flow reversals will occur and continuity can not

be maintained. The cogenerated solutions shown in Figures 11 and 13 have gradients going in both directions at several places throughout the field; this is indicated by the darker shades of the mass balance curves and corresponds to the areas of greatest mass imbalance. In Figure 12, where $\sigma_f^2 = 0.25$, no flow reversals occur (for this particular realization) and the magnitudes of these local mass balance errors is generally much lower than the high variance cases throughout the field. The standard deviation of the local Darcy flux errors for this realization is 4.8%. (these are the local flux errors plotted in Figure 10). Thus, even for smaller input variances, there is some difficulty in achieving an acceptable local mass balance.

Note that the flux errors occur where the gradients of the deterministic and cosimulated head fields differ. There is no constraint on the cogeneration algorithm to prevent reverse head gradients from occuring, since the algorithm involves only perturbed quantities. To impose such a constraint would be so costly computationally that no benefit would be derived from implementing the cosimulation procedure.

In the next section, we develop the theory for conditioning the cosimulations to preserve specified values of the head and $lnT$ processes.

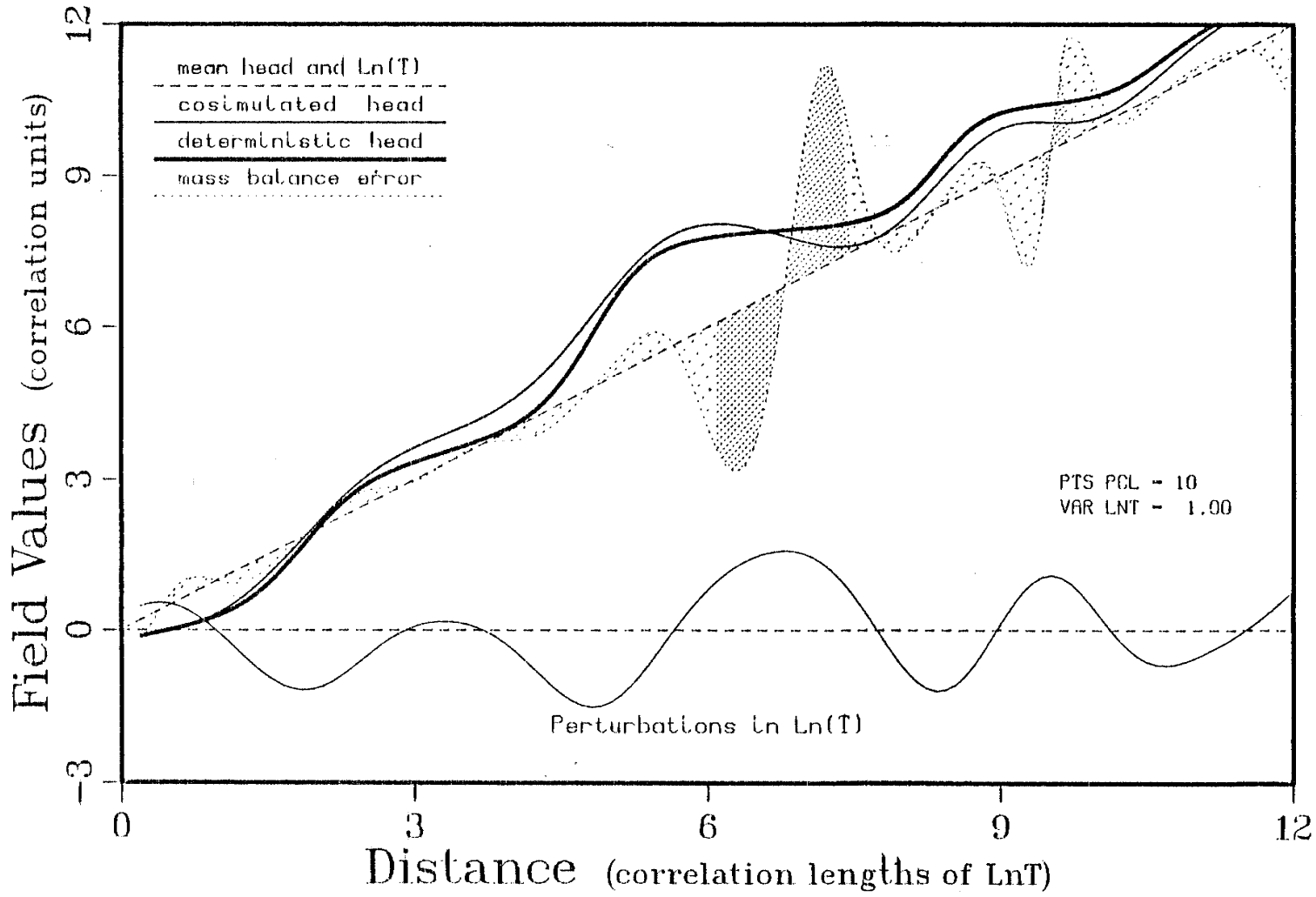



Figure 11. Cosimulated and deterministic heads and $lnT$ perturbations for a bell function realization. Shaded curves represent relative mass balance errors (arbitrary scale). High variance, low discretization case ($\sigma_f^2$= 1.0, 10 pts/cl). Darker shades indicate zones where head gradient reversals occur.

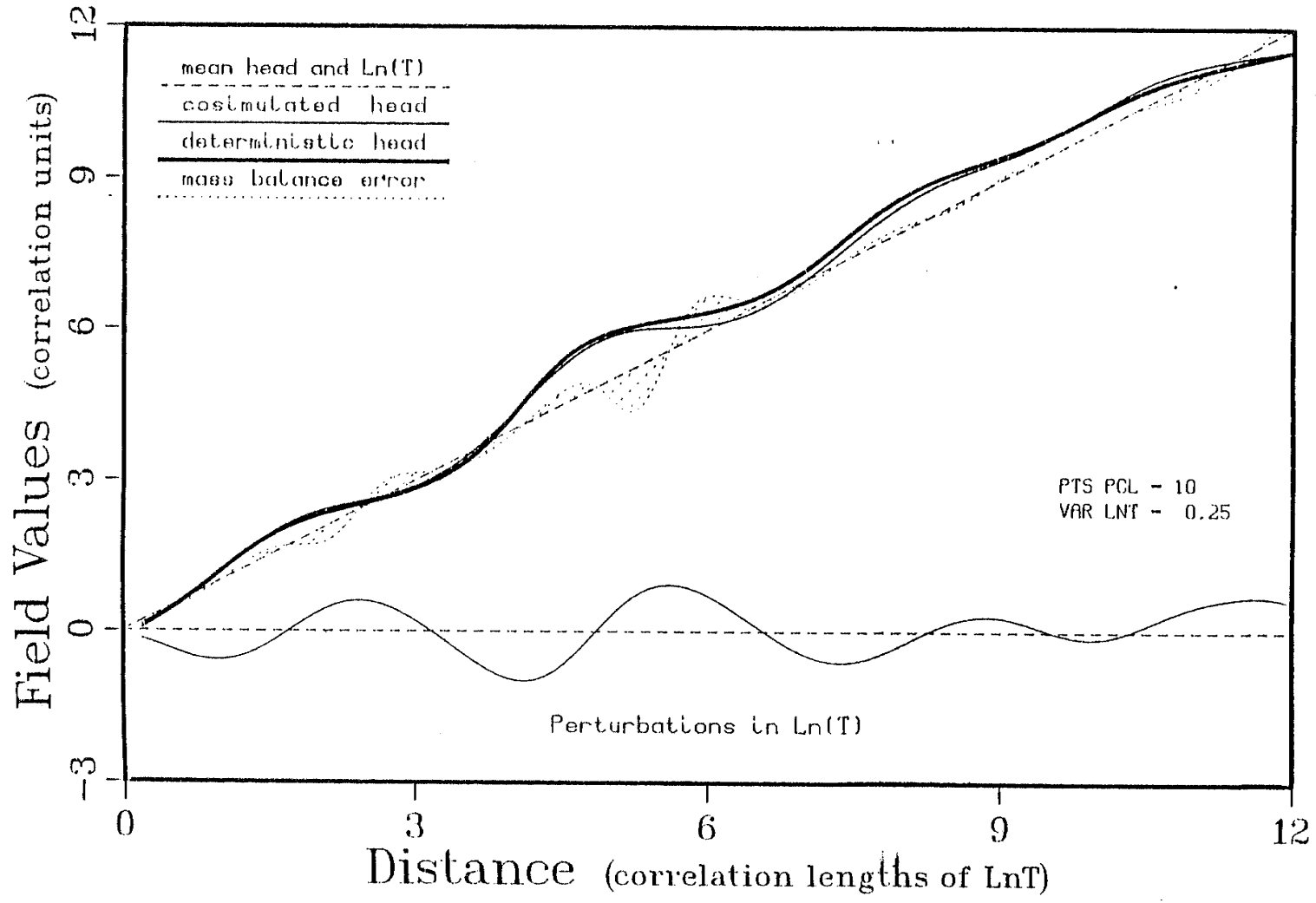



Figure 12. Cosimulated and deterministic heads and $lnT$ perturbations for a bell function realization. Shaded curves represent relative mass balance errors (arbitrary scale). Low variance, low discretization case ($\sigma_f^2$= 0.25, 10 pts/cl). The mass balance errors (dotted line) are plotted in Figure 10.

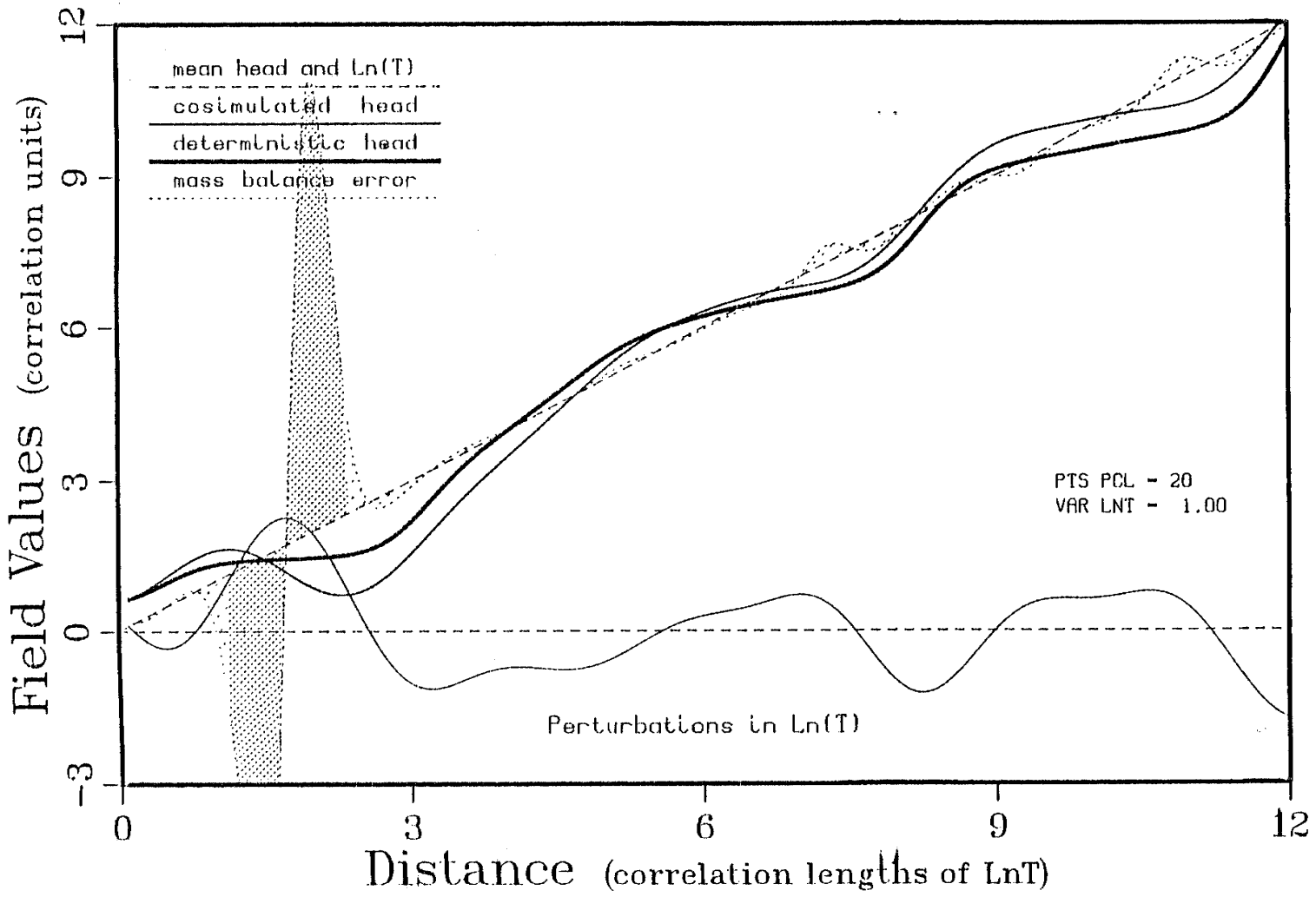



Figure 13. Cosimulated and deterministic heads and $lnT$ perturbations for a bell function realization. Shaded curves represent relative mass balance errors (arbitrary scale). High variance, high discretization case ($\sigma_f^2$= 1.0, 20 pts/cl). Darker shades indicate zones where head gradient reversals occur.

## CONDITIONING

Over the past 10 years there have been an increasing number of papers dealing with the treatment of flow systems in terms of random fields and the development of variance/uncertainty reduction techniques; this has led a variety of approaches for solving the inverse problem [*Wilson et al*, 1978; *Delhomme*, 1979; *Neuman and Yakowitz*, 1979; *Neuman et al*, 1980; *Dagan*, 1982; *Clifton and Neuman*, 1982; *Kitanidis and Vomvoris*, 1983; *Townley and Wilson*, 1983; *Hoeksema and Kitanidis*, 1984,1985; *Dagan*, 1985; *Jacobson*, 1985]. Here, we incorporate knowledge about the head and $lnT$ processes and develop the theory for conditionally cosimulating the head and $lnT$ fields.

The linearized equation for the one-dimensional perturbations in flow is given by (7) where the spectral components of $h$ and $f$ are related by (11). The $f(x)$ and $h(x)$ processes are unconditionally simulated using the spectral representations (20) and (21). To see how conditioning can be incorporated into the algorithm, we write the representations in real form and change to matrix notation†. The discrete representations (20) and (21) can be written in real form as

$$f_j = f(j\Delta x) = 2\sum_{k=0}^{M-1} [u_k cos(\omega_k j\Delta x) - v_k sin(\omega_k j\Delta x)]\sqrt{S_{ff}(\omega_k)\Delta\omega} \quad (38)$$

$$h_j = h(j\Delta x) = 2\sum_{k=0}^{M-1} \frac{J}{\omega_k}[v_k cos(\omega_k j\Delta x) + u_k sin(\omega_k j\Delta x)]\sqrt{S_{ff}(\omega_k)\Delta\omega} \quad (39)$$

where *j=0,1... M-1*. In vector/matrix notation (38) and (39) become

$$\vec{f} = \vec{A}\begin{pmatrix}\vec{u}\\ \vec{v}\end{pmatrix} \quad (40)$$

$$\vec{h} = \vec{B}\begin{pmatrix}\vec{u}\\ \vec{v}\end{pmatrix} \quad (41)$$

where $\vec{f}$ and $\vec{h}$ are the random field column vectors

$$\vec{f} = (f_0, f_1 \ldots f_{M-1})^t$$

$$\vec{h} = (h_0, h_1 \ldots h_{M-1})^t,$$

† The convention used here is lower case for vectors and upper case for matrices; both vectors and matrices are bold-faced with an arrow accent. Hopefully, this will avoid confusion with the convention of using capital letters for random processes and lower case letters for the values of the processes.

$\vec{u}$ and $\vec{v}$ are column vectors of uncorrelated normally distributed random numbers

$$\vec{u} = (u_0, u_1 \ldots u_{M-1})^t$$

$$\vec{v} = (v_0, v_1 \ldots v_{M-1})^t,$$

$\vec{A}$ is an $M \times 2M$ matrix with elements

$$\begin{aligned} a_{j,k} &= 2cos(\omega_k j\Delta x)\sqrt{S_{ff}(\omega_k)\Delta\omega} \qquad k=0,1\ldots M\text{-}1 \\ a_{j,k+M} &= -2sin(\omega_k j\Delta x)\sqrt{S_{ff}(\omega_k)\Delta\omega} \qquad k=0,1\ldots M\text{-}1 \end{aligned} \tag{42}$$

and $\vec{B}$ is an $M \times 2M$ matrix with elements

$$\begin{aligned} b_{j,k} &= 2\frac{J}{\omega_k}sin(\omega_k j\Delta x)\sqrt{S_{ff}(\omega_k)\Delta\omega} \qquad k=0,1\ldots M\text{-}1 \\ b_{j,k+M} &= 2\frac{J}{\omega_k}cos(\omega_k j\Delta x)\sqrt{S_{ff}(\omega_k)\Delta\omega} \qquad k=0,1\ldots M\text{-}1. \end{aligned} \tag{43}$$

For conditioning, the values of certain elements of the $\vec{f}$ and $\vec{h}$ vectors must be preserved; it is convenient to rearrange the rows of the $\vec{f}$ and $\vec{h}$ matrix equations (40) and (41) to put the observed values first as

$$\vec{f} \equiv \begin{pmatrix} \vec{f}_0 \\ \vec{f}_1 \end{pmatrix} = \begin{pmatrix} \vec{A}_0 \\ \vec{A}_1 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} \tag{44}$$

$$\vec{h} \equiv \begin{pmatrix} \vec{h}_0 \\ \vec{h}_1 \end{pmatrix} = \begin{pmatrix} \vec{B}_0 \\ \vec{B}_1 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} \tag{45}$$

where $\vec{f}_0$ and $\vec{h}_0$ are vectors of the observations (conditioning points) containing $n_f$ and $n_h$ elements respectively, and $\vec{f}_1$ and $\vec{h}_1$ are vectors of the unconditioned values of $\vec{f}$ and $\vec{h}$ having $M - n_f$ and $M - n_h$ elements respectively. Our objective is to find the $\begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix}$ that satisfy (44) and (45) and have the proper probabilistic structure. This is accomplished using standard conditional simulation techniques to obtain conditioned vectors $\begin{pmatrix} \vec{u}_c \\ \vec{v}_c \end{pmatrix}$ such that (44) and (45) hold and then to transform these as in (20) and (21) to obtain $\vec{f}_1$ and $\vec{h}_1$. The procedure is as follows:

(1) Generate unconditionally simulated vectors $\vec{u}_s$ and $\vec{v}_s$.

(2) Form kriged estimates of $\vec{u}$ and $\vec{v}$, denoted as $\vec{u}_K$ and $\vec{v}_K$, based on the objective of preserving observations $\vec{f}_0$ and $\vec{h}_0$.

(3) Form kriged estimates of $\vec{u}_s$ and $\vec{v}_s$, denoted as $\vec{u}_{sK}$ and $\vec{v}_{sK}$, for the same observation locations as in step 2.

(4) generate the conditionally simulated vectors $\vec{u}_c$ and $\vec{v}_c$ via

$$\begin{pmatrix} \vec{u}_c \\ \vec{v}_c \end{pmatrix} = \begin{pmatrix} \vec{u}_K \\ \vec{v}_K \end{pmatrix} + \left[ \begin{pmatrix} \vec{u}_s \\ \vec{v}_s \end{pmatrix} - \begin{pmatrix} \vec{u}_{sK} \\ \vec{v}_{sK} \end{pmatrix} \right]. \tag{46}$$

Then the conditionally cosimulated random fields $\vec{f}_1$ and $\vec{h}_1$ are obtained by substituting the $\begin{pmatrix} \vec{u}_c \\ \vec{v}_c \end{pmatrix}$ vectors from (46) into (44) and (45). Each of these steps will be discussed in more detail later. To better understand how this technique is carried out, particularly steps (2) and (3) above, it is helpful to briefly review some of the theory behind the geostatistical technique of kriging.

Review of Kriging Concepts (known mean case)

Kriging is a local estimation or interpolation technique that attempts to find the best estimator of the mean value of a regionalized variable over some domain. Here we'll consider only one-dimensional stationary processes, $Y(x)$, with mean equal to zero. The kriging estimator, $\hat{Y}(x)$, is defined as a linear combination of $n$ known or observed values $Y_o(x_i)$ as

$$\hat{Y}(x) \;=\; \sum_{i=1}^{n} \lambda_i Y_0(x_i) \tag{47}$$

where the $n$ $\lambda_i$'s are unknown kriging weights which are unique to each point $x_k$ for which $Y(x_k)$ is to be estimated. This estimator is unbiased ($E[\hat{Y}(x)] = E[Y(x)]$) and is called the Best Linear Unbiased Estimator (BLUE). It is "best" in the mean-square sense in that it finds the $\lambda_i$'s which minimize the mean of the squared differences between the predicted and known values, ie.

$$\text{BLUE minimizes } E\left[(\hat{Y}(x) - Y(x))^2\right]. \tag{48}$$

By substituting (47) into (48) and reducing [*see eg, Journel and Huijbregts*, 1978], a system of $n$ linear equations in $n$ unknowns (the $\lambda_i$'s) is obtained for each $x_k$,

$$\sum_{i=1}^{n} \lambda_i C(x_i - x_j) \;=\; C(x_i - x_k) \qquad j = 1, 2 \ldots n \tag{49}$$

where $C(x_i - x_j)$ is the covariance between $Y_o(x_i)$ and $Y_o(x_j)$, and $C(x_i - x_k)$ is the covariance between $Y_o(x_i)$, and $\hat{Y}(x_k)$. Knowledge of the covariance behavior of the

process is assumed known. Using vector/matrix notation, (49) is written as

$$\vec{\Sigma}_0 \vec{\lambda}_k \;=\; \vec{c}_k$$

where the vector of kriging weights, $\vec{\lambda}_k = (\lambda_1, \lambda_2 \ldots \lambda_n)_k^t$, is subscripted to indicate its association with the point $x_k$. $\vec{\Sigma}_0$ is an $n \times n$ covariance matrix of the observation values, and $\vec{c}_k$ is an $n \times 1$ vector of covariances between the observation points and the kriged point $x_k$. If the same set of observations are used for all prediction points, then the $\vec{\Sigma}_0$ matrix need be inverted only once and the $\vec{\lambda}_k$ are found for any $x_k$ by

$$\vec{\lambda}_k \;=\; \vec{c}_k{}^t [\vec{\Sigma}_0]^{-1}. \tag{50}$$

The kriged estimate for point $x_k$ is found by writing (47) in matrix form, $\hat{y}_k = \vec{\lambda}_k^t \vec{y}_0$ (the superscript $^t$ represents a transpose), where $\hat{Y}(x_k) \;\equiv\; \hat{y}_k$, and using (50) to obtain

$$\hat{y}_k \;=\; \vec{c}_k{}^t [\vec{\Sigma}_0]^{-1}\, \vec{y}_0 \tag{51}$$

where $\vec{y}_0$ is the vector of observation values, $\vec{y}_0 = (y_1, y_2 \ldots y_n)_o^t$ with $y_i$ corresponding to $Y_o(x_i)$ in (47). The $\vec{\lambda}$ for all points $x_k$, $k = 1, 2 \ldots m$ can be written in matrix form as

$$\vec{\Lambda} \;=\; [\vec{\Sigma}_0]^{-1} \vec{\Sigma}_\times \tag{52}$$

where $\vec{\Lambda}$ is an $n \times m$ matrix of kriging weights and $\vec{\Sigma}_\times$ is an $n \times m$ matrix of covariances between the prediction points and the observation points. The kriged estimates for all $x_k$, which we denote here as $\vec{y}_1$ for analogy later on, are given by

$$\vec{y}_1 \;=\; \vec{\Sigma}_\times^t [\vec{\Sigma}_0]^{-1}\, \vec{y}_0. \tag{53}$$

To summarize the preceding results, let $\vec{y}$ represent a discrete, one-dimensional, mean-zero random field composed of observed values $\vec{y}_0$ and kriged values $\vec{y}_1$,

$$\vec{y} \;\equiv\; \begin{pmatrix} \vec{y}_0 \\ \vec{y}_1 \end{pmatrix}. \tag{54}$$

We can write the covariance matrix for $\vec{y}$ as

$$Cov(\vec{y}) \;=\; E\left[ \begin{pmatrix} \vec{y}_0 \\ \vec{y}_1 \end{pmatrix} \cdot \begin{pmatrix} \vec{y}_0 \\ \vec{y}_1 \end{pmatrix}^t \right] \;\equiv\; \begin{pmatrix} \vec{\Sigma}_{11} & \vec{\Sigma}_{12} \\ \vec{\Sigma}_{21} & \vec{\Sigma}_{22} \end{pmatrix} \tag{55}$$

where the submatrices in (55) are defined as

$$\vec{\Sigma}_{11} \equiv \vec{\Sigma}_0 = Cov(\vec{y}_0, \vec{y}_0^t) \tag{56}$$

$$\vec{\Sigma}_{12} = \vec{\Sigma}_{21}^{\,t} \equiv \vec{\Sigma}_\times = Cov(\vec{y}_0, \vec{y}_1^t) \tag{57}$$

$$\vec{\Sigma}_{22} \equiv \vec{\Sigma}_k = Cov(\vec{y}_1, \vec{y}_1^t)$$

where $\vec{\Sigma}_{22}$ is an $m$ by $m$ unconditional covariance matrix of the prediction points, $x_k$, $k = 1, 2 \ldots m$. Then from (53), (56) and (57), we can write the kriged estimates of $\vec{y}_1$ based on the observations $\vec{y}_0$ in terms of the submatrices in (55) as

$$\vec{y}_1 = \vec{\Sigma}_{21}[\vec{\Sigma}_{11}]^{-1}\,\vec{y}_0. \tag{58}$$

The kriging covariance matrix, $\vec{\Sigma}_K$, (the covariance matrix of the prediction points *conditioned* on the observations $\vec{y}_0$) can be shown to be

$$\vec{\Sigma}_K = \vec{\Sigma}_{22} - \vec{\Sigma}_\times^t[\vec{\Sigma}_0]^{-1}\vec{\Sigma}_\times. \tag{59}$$

Implementation Procedure

We now return to the four steps necessary for conditionally cosimulating the $\vec{f}$ and $\vec{h}$ fields in (44) and (45).

Step 1: We begin by generating unconditioned vectors $\vec{u}_s$ and $\vec{v}_s$. The $u_k$'s and $v_k$'s (the subscript $k$ now refers to the running index in (38) and (39), not to a prediction point) are mean-zero, variance $\frac{1}{2}$ uncorrelated normal random variables which are obtained via a random number generator. In this work, we used the algorithm of Marsaglia and Bray [*Dudewicz and Rally*, 1981].

Step 2: To form the kriged estimates $\vec{u}_K$ and $\vec{v}_K$ that will preserve the observations $\vec{f}_0$ and $\vec{h}_0$, we first consider only the observations of $\vec{f}$ and $\vec{h}$ (the conditioning points) and write

$$\begin{pmatrix} \vec{f}_0 \\ \vec{h}_0 \\ \vec{u} \\ \vec{v} \end{pmatrix} = \begin{pmatrix} \vec{A}_0 \\ \vec{B}_0 \\ \vec{I}_1 \\ \vec{I}_2 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} \tag{60}$$

where the left hand side is a vector with $n_f + n_h + 2M$ components, $\vec{A}_0$ is an $n_f \times 2M$ matrix, $\vec{B}_0$ is an $n_h \times 2M$ matrix, and $\vec{I}_1$ and $\vec{I}_2$ are each $M \times 2M$ matrices defined as

$$\vec{I}_1 = (\vec{I}, \vec{0})$$
$$\vec{I}_2 = (\vec{0}, \vec{I})$$

where $\vec{I}$ and $\vec{0}$ are both $M \times M$ matrices with $\vec{I} \equiv$ the idendity matrix and $\vec{0} \equiv$ null matrix. Here we include the $\vec{u}$ and $\vec{v}$ vectors in the left hand side of (60) and consider that vector to be a "field" composed of known observations, $\vec{f}_0$ and $\vec{h}_0$, and a collection of random variables $\vec{u}$ and $\vec{v}$. The idea is to look at these as a *joint* set of random variables in which we wish to predict some (the $\vec{u}$ and $\vec{v}$) given others (the $\vec{f}_0$ and $\vec{h}_0$). There is nothing in the theory that says the field to be kriged must actually represent some physical reality; the "field" on the left hand side of (60) is a mixture of random variables from the space domain and the frequency domain.

The $f_j$'s and $h_j$'s are linear combinations of the $u_k$'s and $v_k$'s (equations (38) and (39)) which are multivariate normal random variables, therefore the vector

$$\begin{pmatrix} \vec{f}_0 \\ \vec{h}_0 \\ \vec{u} \\ \vec{v} \end{pmatrix} \quad \text{is also multivariate normal (MVN).} \tag{61}$$

From MVN theory [*Barr and Zehna*, 1971], if a random variable $Y \equiv \vec{y}$ is MVN with mean $\vec{\mu}$ and covariance matrix $\vec{\Sigma}$, denoted as

$$\vec{y} \equiv \begin{pmatrix} \vec{y}_0 \\ \vec{y}_1 \end{pmatrix}, \qquad \vec{\mu} \equiv \begin{pmatrix} \vec{\mu}_0 \\ \vec{\mu}_1 \end{pmatrix}, \qquad \vec{\Sigma} \equiv \begin{pmatrix} \vec{\Sigma}_{11} & \vec{\Sigma}_{12} \\ \vec{\Sigma}_{21} & \vec{\Sigma}_{22} \end{pmatrix}, \tag{62}$$

then the conditional distribution of $\vec{y}_1$ given $\vec{y}_0$ is MVN with

$$\text{conditional mean} = \vec{\mu}_1 + \vec{\Sigma}_{21}[\vec{\Sigma}_{11}]^{-1}[\vec{y}_0 - \vec{\mu}_0] \tag{63}$$
$$\text{and covariance matrix} = \vec{\Sigma}_{22} - \vec{\Sigma}_{21}[\vec{\Sigma}_{11}]^{-1}\vec{\Sigma}_{12}. \tag{64}$$

The MVN results above correspond to the kriging results (53)–(57) with respect to the means and covariances. Thus, the conditional mean $\begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix}$ in (61) correspond to the kriged estimates of $\begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix}$ based on the observations $(\vec{f}_0, \vec{h}_0)^t$.

Substituting $(\vec{f}_0,\ \vec{h}_0)^t$ for $\vec{y}_0$ and $\begin{pmatrix}\vec{u}\\ \vec{v}\end{pmatrix}$ for $\vec{y}_1$ above and using the results from (54)–(58) and (61)–(64) enables us to write the kriged estimates $\vec{u}_K$ and $\vec{v}_K$ that preserve the observations $(\vec{f}_0,\ \vec{h}_0)^t$ (the conditioning points) as

$$\begin{pmatrix}\vec{u}_K\\ \vec{v}_K\end{pmatrix} = \vec{\Sigma}_x^t[\vec{\Sigma}_0]^{-1}\begin{pmatrix}\vec{f}_0\\ \vec{h}_0\end{pmatrix}. \tag{65}$$

Step 3: To find the kriged estimates of $\vec{u}_s$ and $\vec{v}_s$ at the locations corresponding to the observation points, we use the kriging weights $\vec{\Sigma}_x^t[\vec{\Sigma}_0]^{-1}$ from (65) to obtain he kriged estimates $\vec{u}_{sK}$ and $\vec{v}_{sK}$ for a field $\begin{pmatrix}\vec{u}'\\ \vec{v}'\end{pmatrix}$ defined by

$$\begin{pmatrix}\vec{u}'\\ \vec{v}'\end{pmatrix} = \begin{pmatrix}\vec{u}_{s0}\\ \vec{v}_{s0}\\ \vec{u}_{sK}\\ \vec{v}_{sK}\end{pmatrix}. \tag{66}$$

Here we predict (krige) the $\begin{pmatrix}\vec{u}_{sK}\\ \vec{v}_{sK}\end{pmatrix}$ based on observations $\begin{pmatrix}\vec{u}_{s0}\\ \vec{v}_{s0}\end{pmatrix}$ where the $\begin{pmatrix}\vec{u}_{s0}\\ \vec{v}_{s0}\end{pmatrix}$ are the unconditioned $\begin{pmatrix}\vec{u}_s\\ \vec{v}_s\end{pmatrix}$ values at the points corresponding to $(\vec{f}_0,\ \vec{h}_0)^t$.

Step 4: Finally, we generate $\begin{pmatrix}\vec{u}_c\\ \vec{v}_c\end{pmatrix}$ from the simulated and kriged $\begin{pmatrix}\vec{u}\\ \vec{v}\end{pmatrix}$ fields using (46) and substitute that result into (44) and (45) to obtain conditionally cosimulated $\vec{f}$ and $\vec{h}$ fields.

Although this conditioning algorithm has not been implemented in this work, it appears to be a relatively straight forward procedure. The development of the theory is the same for two- and three-dimensions, the only difference being the form of (38) and (39) and the dimensions of (40) and (41).

## Cogeneration and Analysis of Discrete Two–Dimensional Random Fields

In this section we develop the theory and algorithms for the cosimulation of two-dimensional random fields of head and log-transmissivity. Implementing and analysing the two-dimensional problem is considerably more complicated than the one-dimensional case and many problems have been and continue to be encountered as this work is being developed. Consequently, as of this writing, the work is still in progress, therefore the results presented here must be viewed as preliminary only.

We begin with the governing partial differential equation for steady, confined, two-dimensional flow,

$$\frac{\partial}{\partial x_1}\Big(T\frac{\partial \phi}{\partial x_1}\Big) + \frac{\partial}{\partial x_2}\Big(T\frac{\partial \phi}{\partial x_2}\Big) = 0, \tag{67}$$

which is transformed into the equation governing the perturbations in flow,

$$\frac{\partial^2 h}{\partial x_1^2} + \frac{\partial^2 h}{\partial x_2^2} - J\frac{\partial f}{\partial x_1} = 0, \tag{68}$$

in exactly the same fashion as in the one-dimensional case. The details of this development are presented in *Mizell et al*, [1982]. Substituting the Fourier-Stieltjes representations (8) and (9) into (68) and differentiating leads to

$$dZ_h(\vec{\omega}) = \frac{-J_1 i\omega_1}{\omega_1^2+\omega_2^2}\, dZ_f(\vec{\omega}) \tag{69}$$

which relates the complex Fourier amplitudes of the fluctuations between the $f(\vec{x})$ and $h(\vec{x})$ processes. The vectors $\vec{x}$ and $\vec{\omega}$ are defined as $\vec{x} = (x_1, x_2)$ and $\vec{\omega} = (\omega_1, \omega_2)$. Here the mean hydraulic gradient $J$ is subscripted to indicate that the mean gradient is in the $x_1$-direction; $\omega_1$ and $\omega_2$ refer to spatial frequencies in the $x$ and $y$ directions. Applying the result (14) to both sides of (69) leads to the spectral relationship between the head and $lnT$ perturbations in two-dimensions,

$$S_{hh}(\vec{\omega}) = \frac{J_1^2\omega_1^2}{(\omega_1^2+\omega_2^2)^2} S_{ff}(\vec{\omega}). \tag{70}$$

We chose to use the Telis spectrum for the $lnT$ process, an isotropic spectral density given by [*Mantoglou and Wilson*, 1981,1982]

$$S_{ff}(\omega) = \frac{2\sigma_f^2 b}{\pi^2}\frac{|\omega|}{(b^2+\omega^2)^2} \tag{71}$$

where $\omega^2 = \omega_1^2 + \omega_2^2$ and $b$ is a correlation parameter. In Appendix B we show that the Telis spectrum leads to a stationary head process. To cogenerate two-dimensional fields we make repeated use of (20) and (21) with

$$\frac{-Ji}{\omega_k} \quad \text{in (21) replaced by} \quad \frac{-J_1 i\omega_1}{\omega_1^2+\omega_2^2}$$

where now, the integration is carried out over two-dimensional frequency space. The one-dimensional cosimulation algorithm described previously is relatively straightforward and easy to implement. The two-dimensional algorithm, on the other hand, is significantly more complicated and care must be taken to insure that the arrays are filled in the proper manner.

The Two-Dimensional Algorithm

To understand how this is implemented on the computer, we write the two-dimensional form of the spectral representation theorem (13) as

$$f(\vec{\boldsymbol{x}}) \;=\; \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{i\vec{\omega}\cdot\vec{\boldsymbol{x}}}\, dZ_f(\vec{\omega}) \tag{72}$$

where $\vec{\boldsymbol{x}} = (x_1, x_2)^t$ and $\vec{\omega} = (\omega_1, \omega_2)^t$, and the superscript $^t$ represents a vector-matrix transpose. We begin by discretizing the two-dimensional frequency domain into $2M$ intervals along each frequency axis and writing (72) as

$$\begin{aligned} f(x_1,x_2) \;=\; \sum_{m_1=-M}^{M-1}\sum_{m_2=-M}^{M-1} exp\Big\{ i\Big[(m_1+\tfrac{1}{2})\,\Delta\omega x_1 + (m_2+\tfrac{1}{2})\Delta\omega x_2\Big]\Big\} \\ \cdot\; dZ_f\Big((m_1+\tfrac{1}{2})\Delta\omega, (m_2+\tfrac{1}{2})\Delta\omega\Big) \end{aligned} \tag{73}$$

where each frequency, $\omega_{m_j} \equiv (m_j+\frac{1}{2})\Delta\omega$ with $\Delta\omega = \Omega/2M$ and $\Omega$ is the maximum frequency at which the spectrum is truncated. Next, we discretize space by setting $x_i = j_i\Delta x$, $i=1,2$ $j=0,1\ldots M-1$. Then we define the relationship between frequency and space as $\Delta x = \frac{2\pi}{\Omega} = \frac{2\pi}{N\Delta\omega}$, $N = 2M$, whence (73) becomes

$$\begin{aligned} f(j_1\Delta x, j_2\Delta x) \;=\; \\ \sum_{m_1=-M}^{M-1}\sum_{m_2=-M}^{M-1} exp\Big\{ i\,2\pi\,\Big[(m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2\Big]/N\Big\}\, dZ_f\Big(m_1, m_2\Big) \end{aligned} \tag{74}$$

where the discrete frequencies inside the $dZ_f(\ )$ term are referenced by the indices $m_1$ and $m_2$ for convenience. Also, we let $\Delta x = 1$ so that the discrete points $f(j_1\Delta x, j_2\Delta x)$ can be referenced simply as $f(j_1, j_2)$. We will see that the conditions which allow us to use "twice the real part" of a Fourier transformed sequence as in the one-dimensional case, arise from the symmetry between quadrants over which the integration is carried out in two-dimensional frequency space. To show this, we break (74) into four sums as

$$
\begin{aligned}
f(j_1, j_2) \ = \\
&\sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} exp\Big\{ i2\pi \Big[ (m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \Big]/N \Big\}\, dZ_f(m_1, m_2) \ + \\
&\sum_{m_1=-M}^{-1} \sum_{m_2=0}^{M-1} exp\Big\{ i2\pi \Big[ (m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \Big]/N \Big\}\, dZ_f(m_1, m_2) \ + \\
&\sum_{m_1=-M}^{-1} \sum_{m_2=-M}^{-1} exp\Big\{ i2\pi \Big[ (m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \Big]/N \Big\}\, dZ_f(m_1, m_2) \ + \\
&\sum_{m_1=0}^{M-1} \sum_{m_2=-M}^{-1} exp\Big\{ i2\pi \Big[ (m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \Big]/N \Big\}\, dZ_f(m_1, m_2) \ +
\end{aligned}
\tag{75}
$$

Here, we will refer to each of the double sums in (75) as $\Sigma 1$, $\Sigma 2$, $\Sigma 3$, and $\Sigma 4$ corresponding to the quadrants over which the summations apply and the order in which the terms appear in the equation. Thus, we write (75) as

$$
f(j_1, j_2) \ = \ \Sigma 1 \ + \ \Sigma 2 \ + \ \Sigma 3 \ + \ \Sigma 4. \tag{76}
$$

We now change the indices in $\Sigma 3$ by substituting $m_1' = -(m_1+1)$ and $m_2' = -(m_2+1)$ so that $\Sigma 3$ becomes

$$
\begin{aligned}
\Sigma 3 \ = \ \sum_{m_1'=M-1}^{0} \sum_{m_2'=M-1}^{0} &exp\Big\{ i2\pi \Big[ -(m_1'+\tfrac{1}{2})j_1 - (m_2'+\tfrac{1}{2})j_2 \Big]/N \Big\} \\
&\cdot\, dZ_f\big(-m_1', -m_2'\big).
\end{aligned}
\tag{77}
$$

The $dZ_f$ process is constructed such that

$$
dZ_f(-m_1, -m_2) \ = \ dZ_f^*(m_1, m_2), \tag{78}
$$

hence, (77) becomes

$$\Sigma 3 = \sum_{m_1'=0}^{M-1}\sum_{m_2'=0}^{M-1} exp\Big\{-i2\pi\Big[(m_1'+\tfrac{1}{2})j_1+(m_2'+\tfrac{1}{2})j_2\Big]/N\Big\}dZ_f^*(m_1',m_2') \quad (79a)$$

$$\Sigma 3 = \Sigma 1^* \quad (79b)$$

The symmetry between the first and third quadrants is such that one is the complex conjugate of the other. In a similar manner as that for quadrants 2 and 4, we let $m_1' = -(m_1+1)$ in $\Sigma 2$ to give

$$\Sigma 2 = \sum_{m_1'=0}^{M-1}\sum_{m_2=0}^{M-1} exp\Big\{i2\pi\Big[-(m_1'+\tfrac{1}{2})j_1+(m_2+\tfrac{1}{2})j_2\Big]/N\Big\}\, dZ_f(-m_1',m_2)$$

and $m_2' = -(m_2+1)$ in $\Sigma 4$ giving

$$\Sigma 4 = \sum_{m_1=0}^{M-1}\sum_{m_2'=0}^{M-1} exp\Big\{i2\pi\Big[(m_1+\tfrac{1}{2})j_1-(m_2'+\tfrac{1}{2})j_2\Big]/N\Big\}\, dZ_f(m_1,-m_2').$$

Writing (78) as $dZ_f(-m_1,m_2) = dZ_f^*(m_1,-m_2)$ shows that

$$\Sigma 4 = \sum_{m_1=0}^{M-1}\sum_{m_2'=0}^{M-1} exp\Big\{-i2\pi\Big[-(m_1+\tfrac{1}{2})j_1+(m_2'+\tfrac{1}{2})j_2\Big]/N\Big\}dZ_f^*(-m_1,m_2') \quad (80a)$$

$$\Sigma 4 = \Sigma 2^* \quad (80b)$$

Thus, using the results (79b) and (80b) we can write (76) as

$$\begin{aligned} f(j_1,j_2) &= \Sigma 1+\Sigma 2 \;+\; \Sigma 1^*+\Sigma 2^* \\ &= (\Sigma 1+\Sigma 2) \;+\; (\Sigma 1+\Sigma 2)^* \\ &= 2Re\Big\{\,\Sigma 1+\Sigma 2\,\Big\}. \end{aligned} \quad (81)$$

In order to combine the results into one expression (and consequently one FFT) as indicated in (81), we write $\Sigma 1 + \Sigma 2$ as the sum of the first two terms on the right-hand side of (75); we then change the $m_1$ index in $\Sigma 2$ to run from $M$ to $2M-1$ by multiplying the second sum in (75) by $e^{i2\pi 2Mj_1/N} = 1$ giving

$$\Sigma 2 = \sum_{m_1=-M}^{-1}\sum_{m_2=0}^{M-1} exp\Big\{i2\pi\Big[(m_1+\tfrac{1}{2}+2\mathrm{M})j_1+(m_2+\tfrac{1}{2})j_2\Big]/N\Big\}dZ_f(m_1,m_2), \quad (82)$$

and letting $m_1' = m_1 + 2M$ so that (82) becomes

$$\Sigma 2 = \sum_{m_1'=M}^{2M-1} \sum_{m_2=0}^{M-1} exp\Big\{ i2\pi \Big[ (m_1'+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \Big] / N \Big\} dZ_f(m_1'-2M, m_2). \quad (83)$$

We can now combine (83) with the first term in (75) to write $(\Sigma 1 + \Sigma 2)$ as

$$\Sigma 1 + \Sigma 2 = \sum_{m_1=0}^{2M-1} \sum_{m_2=0}^{M-1} exp\Big\{ i2\pi \Big[ (m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \Big] / N \Big\} dZ_f^{\circ}(m_1, m_2) \quad (84a)$$

where

$$dZ_f^{\circ}(m_1, m_2) = \begin{cases} dZ_f(m_1, m_2) & 0 \le m_1 \le M-1 \\ dZ_f(m_1-2M, m_2) & M \le m_1 \le 2M-1. \end{cases} \quad (84b)$$

The discrete $dZ_f(\ )$ process is constructed in a manner analogous to (19) as

$$\begin{aligned} dZ_f(m_1, m_2) \equiv dZ_f(\omega_{m_1}, \omega_{m_2}) &= dZ_f\Big( (m_1+\tfrac{1}{2})\Delta\omega, (m_2+\tfrac{1}{2})\Delta\omega \Big) \\ &\cong \sqrt{S_{ff}\Big( (m_1+\tfrac{1}{2})\Delta\omega, (m_2+\tfrac{1}{2})\Delta\omega \Big)}\, \Delta\omega \left( U_{m_1,m_2} + iV_{m_1,m_2} \right). \end{aligned} \quad (85)$$

Finally, recall that the anti-periodic behavior of the one-dimensional finite Fourier Transform (see appendix A) allows us to use only half the output sequence when transforming each row or column of the two-dimensional matrix. In general this does not pose any problems, but in two-dimensions it means that 3/4ths of the transformed field must be thrown away. Thus, two-dimensional simulations may require very large input arrays of size equal to quadruple the number of generation points in the output field. For example, the representation (84) is comprised of an $2M \times M$ matrix of $dZ_f(\ )$ terms which are to be transformed; due to the anti-periodicity, the output field should be no larger than $M \times M/2$. In order to make full use of information contained in (84) and obtain a field of size $M \times M$, we must augment the $d\vec{Z}_f$ matrix with zeros as

$$d\vec{Z}_f \equiv \begin{pmatrix} \vec{\Sigma}1 & \vec{\Sigma}2 \\ \vec{0} & \vec{0} \end{pmatrix} \quad (86)$$

where

$$\vec{0} \equiv \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

and

$$\vec{\Sigma}1 \equiv \begin{pmatrix} dZ_f(0,0) & \dots & dZ_f(M\text{-}1,0) \\ \vdots & \ddots & \vdots \\ dZ_f(0,M\text{-}1) & \dots & dZ_f(M\text{-}1,M\text{-}1) \end{pmatrix}$$

$$\vec{\Sigma}2 \equiv \begin{pmatrix} dZ_f(-M,0) & \dots & dZ_f(-1,0) \\ \vdots & \ddots & \vdots \\ dZ_f(-M,M\text{-}1) & \dots & dZ_f(-1,M\text{-}1) \end{pmatrix}.$$

Each of these sub-matrices is of size $M \times M$. We then transform the $d\vec{\boldsymbol{Z}}_f$ matrix, take twice the real part of the transformed matrix, and use one-quarter of the resulting matrix as the $M \times M$ random field. The discrete $M \times M$ random field, $f(j_1, j_2)$, can thus be represented in summation notation as

$$f(j_1,j_2) = \sum_{m_1=0}^{2M-1} \sum_{m_2=0}^{2M-1} exp\Big\{ i2\pi \left[ (m_1+\tfrac{1}{2})j_1 + (m_2+\tfrac{1}{2})j_2 \right] / N \Big\} dZ_f^{\circ}(m_1,m_2) \quad (87a)$$

where

$$dZ_f^{\circ}(m_1,m_2) = \begin{cases} dZ_f(m_1,m_2) & 0 \le m_1 \le M\text{-}1, \quad m_2 \le M\text{-}1 \\ dZ_f(m_1\text{-}2M,m_2) & M \le m_1 \le 2M\text{-}1, \quad m_2 \le M\text{-}1 \\ 0 & m_2 \ge M. \end{cases} \quad (87b)$$

To cogenerate the head field along with the $lnT$ field, we generate the corresponding $dZ_h(\ )$ process by multiplying the $dZ_f^{\circ}(\ )$ in (87) by the transfer function shown in (69) as the coefficient to $dZ_f(\vec{\omega})$, and use the same random numbers $U_{m_1,m_2}$ and $V_{m_1,m_2}$ as that used for the $lnT$ process in (85).

2D Implementation

For our problem, the cosimulated head field has an anisotropic correlation structure, therefore, we cogenerate the fields onto a rectangular shaped region with the long dimension corresponding with the direction of greater correlation persistence. Since a constant frequency spacing is used, the $m_1$ and $m_2$ indices in (87) run from 0 to $2M_1$–1 and 0 to $2M_2$–1 respectively, where $M_1 \neq M_2$. In the Fortran code, the $m_1$ and $m_2$ indices run from 1 to $2M_1$ and 1 to $2M_2$ by factoring out an $e^{i\pi(j_1+j_2)/N}$

from the argument of the $exp\{\ \}$ term in (87a). Also, the $d\vec{Z}_f$ matrix is Fourier transformed via a series of one-dimensional FFT's along the rows and columns of the matrix ($M_2$ row transforms each of length $2M_1$, and $M_1$ column transforms each of length $2M_2$). Both of these coding strategies result in some computational savings.

The discrete two-dimensional random fields are generated on grids of varying size: A rather large grid of size nodes-x by nodes-y equal to $512 \times 1024$ is used for the statistical analyses; this large size is needed for the head field in order to insure it contains a sufficient number of relatively independent samples upon which to compute its statistics. This is because the heads are correlated over much greater distances than the $lnT$ process. A smaller grid of size $128 \times 256$ is used for the mass balance analysis and for the plots that illustrate the spatial variability of the head and $lnT$ processes.

The spectral and space parameters used for the two-dimensional cosimulations were chosen such that $\sigma_f^2 = 1$ and resolution equals 7 points per correlation length for the large grid, while $\sigma_f^2 = 0.25$ and resolution equals 10 points per correlation length for the smaller grid. Once generated, the fields are plotted using a pseudo-gray shade contouring algorithm where each shade represents an interval between contours of the field. Eight shading patterns are used to represent values ranging from $-3$ to $+3$ standard deviations about the mean with dark shades representing values below the mean and light shades representing values above the mean (If six shading levels were used, then each level would represent one standard deviation in the data, however eight shades are used to capture and illustrate more of the variablity of the process.). This method of presenting the data is particularly well suited for large data sets and has the advantage of displaying the detailed patterns of spatial variability in a way that is often easier to interpret than standard contouring methods. Additionally, some head fields are represented as surfaces in three dimensions for comparitive purposes.

The two-dimensional discrete random fields are analysed in the following way: Variogram/covariance analysis is used to analyse the statistical structure of the individual fields and cross-covariance analysis is used to estimate their joint statistical

behavior. The spatial "character" of the fields is examined through visual inspection to determine if they exhibit the proper spatial patterns relating to correlation scale, isotropy, etc. A "quasi-true" head solution is then obtained through deterministic modeling and the head patterns are compared with the cogenerated solution. Finally, the mass conservation issue is addressed by examining the mass balance residuals and by calculating local percent mass balance errors. Details of each of these analysis steps are discussed in the following sections.

Variogram/Covariance Analyses

Variogram analysis is a commonly used method of estimating the statistical parameters that characterize the probability behavior of random fields. The variogram, $\gamma(\xi)$, (technically the semi-variogram) is defined [*Journel and Huijbregts*, 1978] by

$$\gamma(\xi) = \frac{1}{2}\left[(f(\vec{x}+\vec{\xi}) - f(\vec{x}))^2\right]$$

where $f(\vec{x})$ is the random field and $\vec{\xi}$ is a separation vector. The variogram can be expressed in terms of covariances for a stationary field as

$$\gamma(\xi) = R(0) - R(\xi); \tag{88}$$

since $\gamma(\xi)$ is the covariance function subtracted from the variance, $R(0)$, it resembles an upside-down covariance function. The discrete isotropic variogram analysis is carried out for the cogenerated $lnT$ field using a computationally expensive method involving a series of one-dimensional variogram calculations oriented at various angles in space. More efficient variogram estimates can be made on full matrix data sets using Fast Fourier Transform techniques in a manner similar to that for covariances [*McKay*, 1988]. The head field exhibits an anisotropic covariance structure where the principle axes of anisotropy are alligned with the grid; hence we examine the head covariance structure in directions parallel and normal to the mean flow via a series of one-dimensional covariance calculations along the rows and columns of the head matrix. These variogram estimates are then converted to covariances via (88) and plotted against the theoretical covariance functions.

The theoretical $lnT$ covariance function corresponding to the Telis spectrum (71) was derived by *Mantoglou and Wilson*, [1981] and is given by (B1) in Appendix B. The deviation of the discrete $lnT$ field mean from the theoretical mean was small relative to its sample standard deviation. The variance of the cogenerated $lnT$ field matched the theoretical value to within 5%; this is considered to be a reasonably good match given that the sample statistics are based on a single realization and that the computer generated field represents only a small portion of the underlying theoretical random field. Figure 14 shows the results of the covariance analysis of the cogenerated $lnT$ field; the solid line represents the theoretical covariance while the dots represent the covariance estimates for the cogenerated field. The discrete field does a poor job of preserving the theoretical covariance behavior, an indication that there may be a coding error or some other problem (The theoretical covariance function is evaluated for plotting purposes using equation 12.2.3 of *Abramowitz and Stegun*, [1964] where the Longman method [*Longman*, 1956] was used to evaluate the infinite integral in that equation.).

The head covariance function is derived by first substituting (71) into (70) to obtain the two-dimensional head spectrum

$$S_{hh}(\vec{\omega}) = \frac{2\sigma_f^2 J_1^2 b}{\pi^2} \frac{\omega_1^2}{\omega^3(b^2+\omega^2)^2} \tag{89}$$

where $\omega^2 = \omega_1^2 + \omega_2^2$. The theoretical head covariance corresponding to the head spectrum (89) is derived in Appendix C and is given by

$$R_{hh}(\xi,\chi) = \frac{\sigma_f^2 J_1^2}{b^2}\left\{\frac{2b\xi}{\pi}cos^2\chi + (1-3cos^2\chi)\left[L_0(b\xi)-I_0(b\xi)\right] + \right. \\ \left. - \left(cos^2\chi\left[b\xi+\frac{6}{b\xi}\right] - \frac{3}{b\xi}\right)\left[L_1(b\xi)-I_1(b\xi)\right]\right\} \tag{90}$$

where $\xi$ and $\chi$ are radial coordinates with $\chi$ representing the angle between the mean flow direction and $\xi$. $I_j(\ )$ and $L_j(\ )$ are modified Bessel functions and modified Struve functions respectively, of order $j$. As for the cogenerated $lnT$ process, the deviation of the discrete head field mean from its theoretical mean was small relative to its sample standard deviation. The variance of the head field, however, usually

turned out to be off by approximately a factor of 2, being about one-half the theoretical variance. Initially, it was felt that the 128 × 256 grid was too small to contain a sufficient number of independent samples upon which to calculate the head variance statistics, thus a larger 512 × 1024 grid was used. Even on this larger grid, however, the discrete head variance falls far short of its theoretical value. Covariance estimates are made in directions parallel ($\chi = 0$) and normal ($\chi = \pi/2$) to the mean flow and compared with the theoretical behavior; those results (see Figure 15) show good correspondence between theoretical and experimental behavior at large lags, but very poor correspondence at short lags. This size grid is plenty large enough to "capture" the covariance structure of the process [see eg, *Zimmerman and Wilson*, 1988] given the correlation parameters used here (see section on 2D Implementation), therefore, these rather disappointing results are attributed to an error or errors of unknown origin.

In spite of this, some of these two-dimensional results are encouraging in the sense that the general character of the cosimulated fields appears to behave in the right way. For example, the anisotropic nature of the head field is illustrated in Figure 16 where the head perturbations are plotted (together with the $lnT$ field) using the pseudo-gray shade technique to illustrate the spatial variability patterns. The head field exhibits the proper character in that the head variations are much smoother (correlated over greater distances) than the $lnT$ variations and the anisotropic behavior of the head field is clearly observable and consistent with the expected correlation behavior in the directions normal and parallel to the mean flow (The mean flow direction is normal to the long dimension of the plot). Also the head field shows steeper gradients where the $lnT$ plot is dark (low transmissivity areas) and flat gradients in the lighter areas (high transmissivity zones).

Cross-Covariance Analysis

The joint statistical behavior of the cosimulated fields is examined in directions parallel and normal to the mean flow direction by comparing discrete cross-covariance estimates with the theoretical cross-covariance functions in each of those directions.

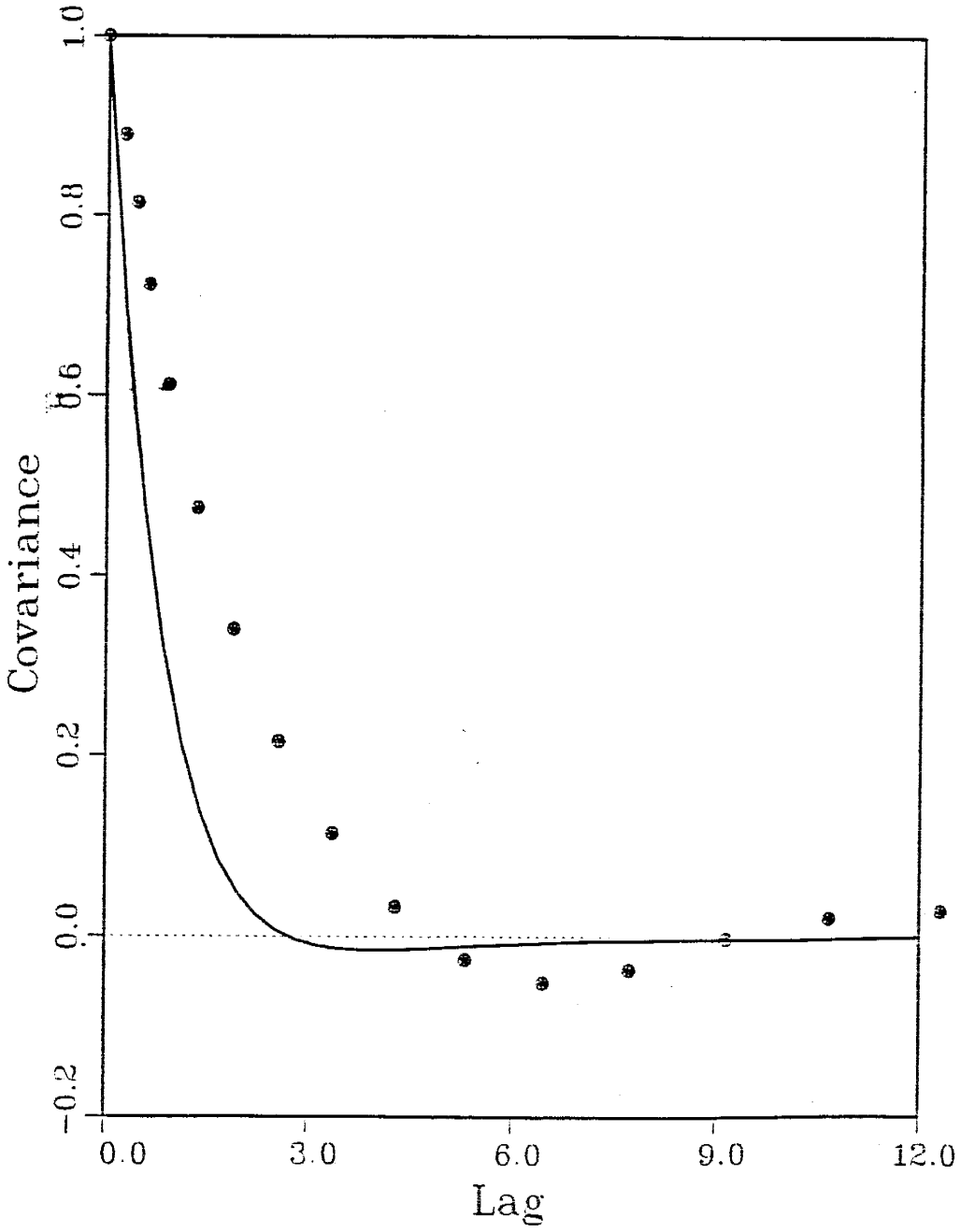



Figure 14. Theoretical (solid curve) and experimental (dots) covariances for the two-dimensional $lnT$ process (Telis covariance, $\sigma_f^2 = 1$).

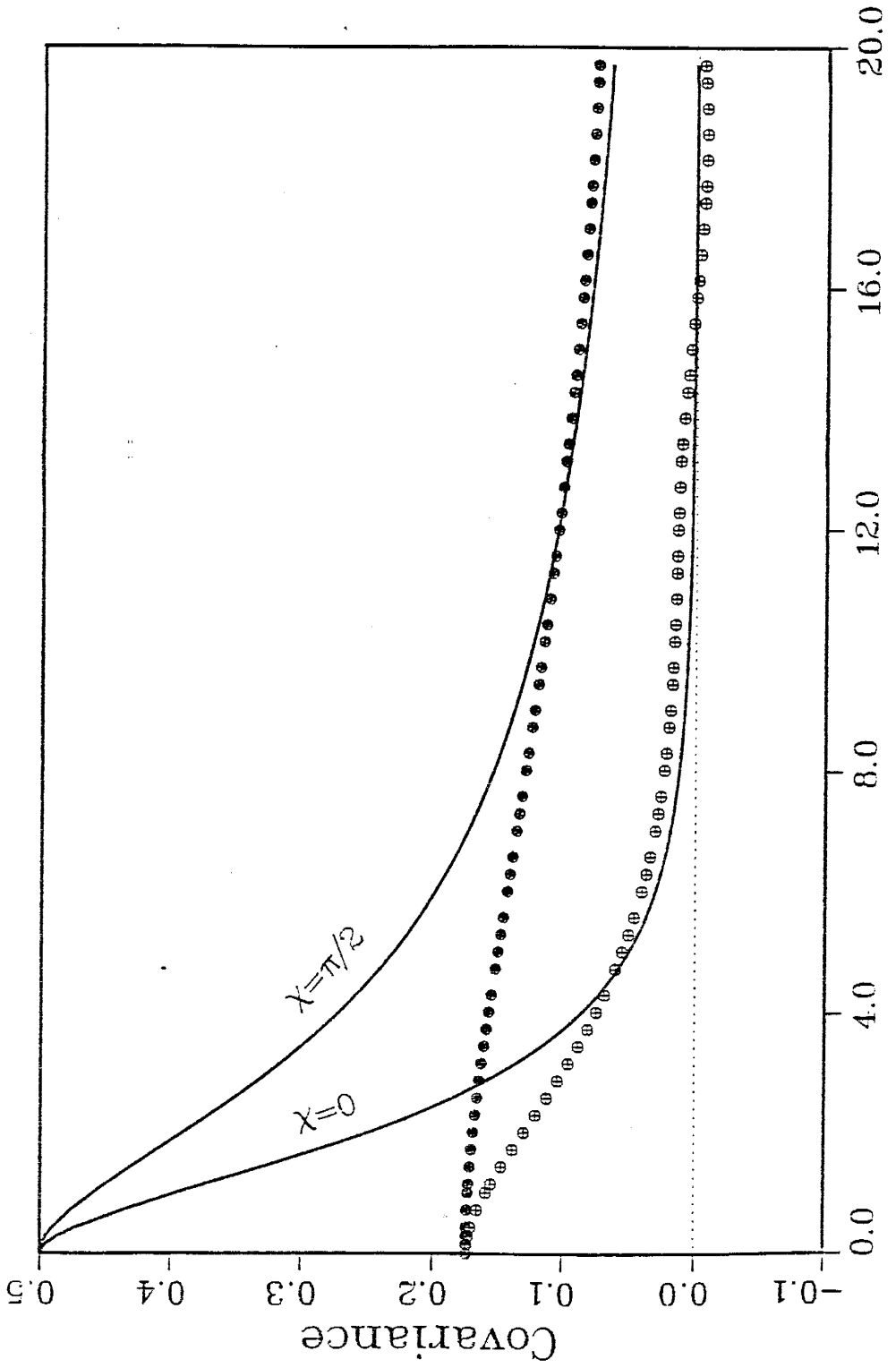



Figure 15. Theoretical (solid curve) and experimental (dots) covariances for the two-dimensional head field (theoretical $\sigma_h^2 = 0.5$).

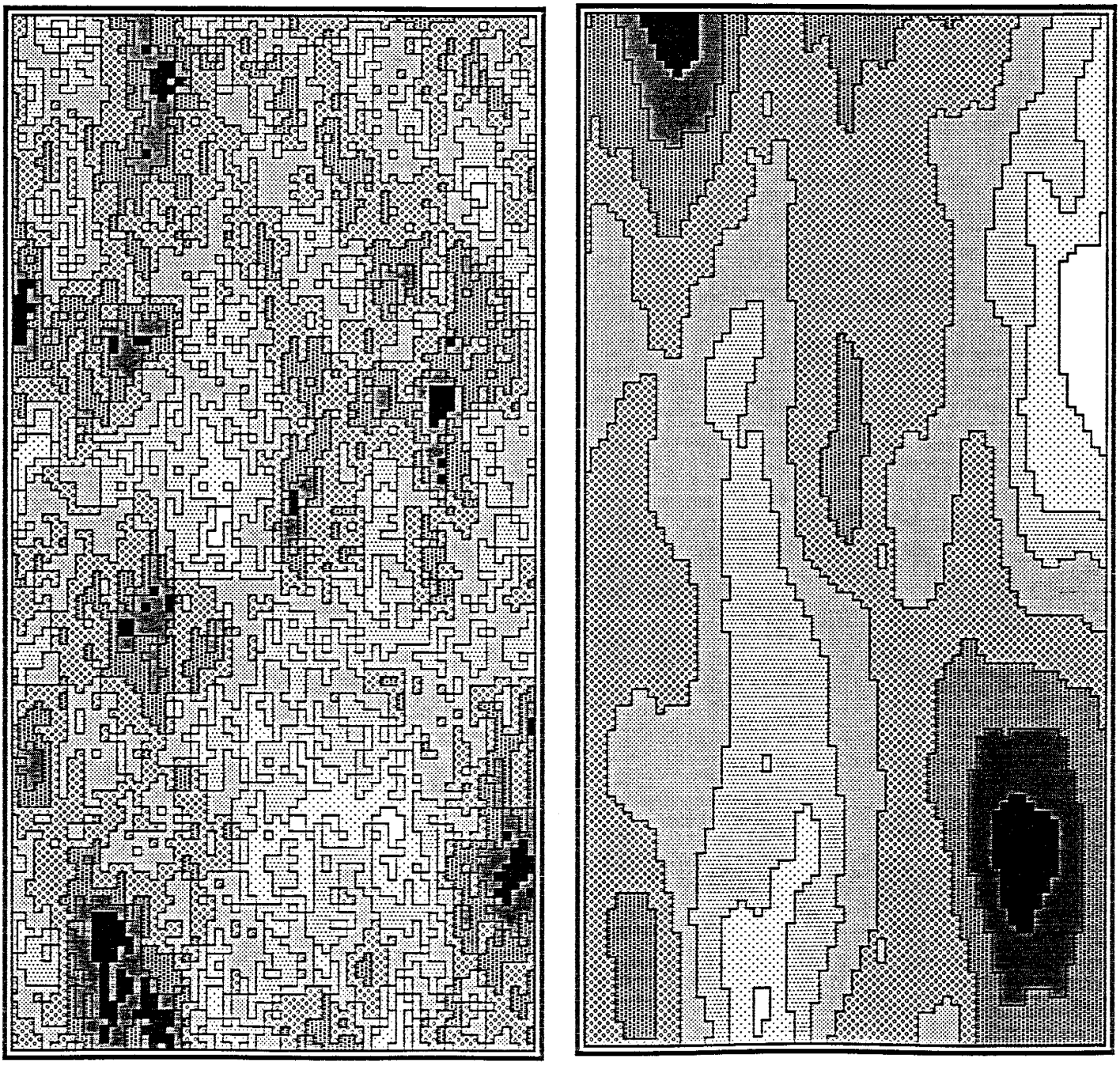


Figure 16. Shaded maps of cosimulated $lnT$ (left) and perturbed head (right) fields. Note differences in correlation persistence and head field anisotropy ($\sigma_f^2$=1). Dark is low value, white is high value.

We begin our analysis by deriving the cross-spectral density function of the head and $lnT$ fields by multiplying both sides of (69) by $dZ_f^*(\vec{\omega})$ and using (14) to obtain

$$S_{fh}(\vec{\omega}) = \frac{-J_1 i\omega_1}{\omega_1^2+\omega_2^2} S_{ff}(\vec{\omega}). \tag{91}$$

Substituting the $lnT$ spectral density (71) into (91) gives

$$\begin{aligned} S_{fh}(\vec{\omega}) &= \frac{-J_1 i\omega_1}{\omega_1^2+\omega_2^2} \frac{2\sigma_f^2 b}{\pi^2} \frac{\omega}{(b^2+\omega^2)^2} \\ &= \frac{-2\sigma_f^2 J_1 b}{\pi^2} \frac{i\omega_1}{\omega(b^2+\omega^2)^2} \end{aligned} \tag{92}$$

$$S_{fh}(\omega,\alpha) = \frac{-2\sigma_f^2 J_1 b}{\pi^2} \frac{i\,cos\alpha}{(b^2+\omega^2)^2} \tag{93}$$

where $\omega^2 = \omega_1^2+\omega_2^2$ and $\omega_1 = \omega\,cos\alpha$, with $\alpha$ representing the angle between $\omega$ and the mean flow direction. It is immediately obvious that, at an angle of $\alpha = \frac{\pi}{2}$, the direction normal to the mean flow, the cross-spectrum is zero for all frequencies, therefore the heads and $lnT$ are uncorrelated in that direction. The theoretical cross-covariance function is derived from the cross-spectral density function (92) in appendix D and is given by

$$R_{fh}(\xi,\chi) = J\sigma_f^2\,cos\chi \left\{ I_0(b\xi) - L_0(b\xi) - \frac{1}{b\xi}\left[ I_1(b\xi) - L_1(b\xi)\right]\right\}. \tag{94}$$

Cross-covariance estimates were made on the output fields in directions parallel and normal to the mean flow directions which correponds with the rows and columns of the output matrices. The discrete cross-covariance estimates are plotted against the theoretical curves in Figure 17. The theoretical curves corresponding to $R_{fh}(\xi,\chi)$ in (94) and $R_{hh}(\xi,\chi)$ in (90) were calculated as described earlier for the theoretical $lnT$ covariance function. One would hope to achieve a better match between theoretical and experimental results than that shown in Figures 14, 15 and 17, however, the discrete auto and cross-covariances do exhibit the correct type of behavior. Also, it should be noted that these covariance calculations are based on single realizations and may therefore lack a sufficient number of independent samples for accurate statistical estimates.

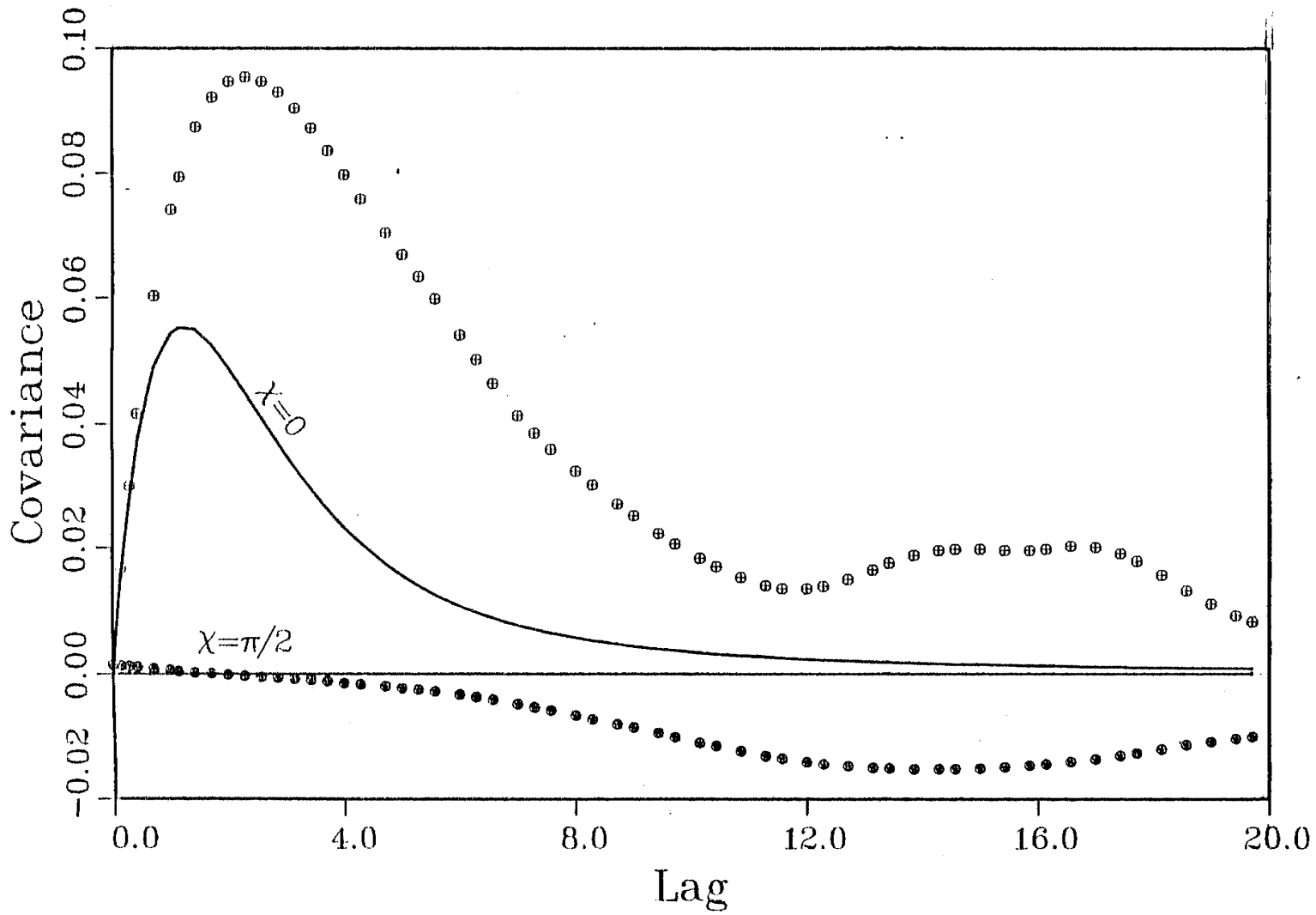



Figure 17. Theoretical (solid curves) and experimental (dots) cross-covariances for two-dimensional Telis fields. $\chi = 0$ is parallel and $\chi = \pi/2$ is normal to the mean flow direction.

Global Mass Balance Analysis

The mass conservation principle must be satisfied both locally and globally in order to use this technique to obtain stochastic predictions of flow, travel paths, travel times, etc. One way of assessing the magnitude of the mass balance errors is to calculate the residuals from the numerical representation

$$\vec{A}\vec{h} - \vec{b} \;=\; \vec{r},$$

where $\vec{A}$ is the cofficient matrix of numerical transmissibilities, $\vec{h}$ is the vector of cosimulated heads, $\vec{b}$ is a vector of source/sink terms, and $\vec{r}$ is the residual vector. If the media were homogeneous, then the local mass balance errors (as measured by the difference between the influx and outflux in a particular node) would be small relative to the mean flux. However, the system is heterogeneous, and the actual flow passing through any grid block of a finite difference grid may be large or small relative to the mean flux; therefore it does not make sense to compare the magnitude of the local mass imbalances with the mean flux. However, the total (or global) mass imbalance, calculated by summing up the local mass imbalances, should be small relative to the mean system flux; ie. the total amount of mass gained or lost within the system, however distributed, should be small relative to the average flow passing through the system. This is a scalar measure of how well the cosimulated fields satisfy the continuity principle.

The flux imbalance at each node is calculated by summing the volumetric Darcy flux across the four edges of each grid block or node. The flux calculations are made using two-dimensional versions of (38) and (39). In two-dimensions, the mean flux, $q_m$, is given by [*Gelhar*, 1986]

$$q_m \;=\; K_g\, J_1 \tag{95}$$

where $K_g$ is an effective hydraulic conductivity equal to the geometric mean or logarithmically averaged hydraulic conductivity. The total or global volumetric system flux, $Q_s$, is given by $Q_s = N_2 q_m \Delta x_2\, \Delta x_3$ where $N_2$ is the number nodes in the $x_2$-direction (normal to the mean flow) and $\Delta x_2$ and $\Delta x_3$ are the horizontal and vertical grid block dimensions in directions normal to the mean flow.

Unfortunately, this residual analysis does not yield very encouraging results: The global mass imbalance for the cogenerated fields is about 10% of the mean system flux whereas the total residual for the deterministic solution amounts to 0.0001% of the system flux. For comparison, we perform the same residual analysis for fields in which *i*) the heads and transmissivities are unrelated (ie. no connection either through cogeneration or deterministic modeling) and *ii*) a case in which there is no randomness in the head field (ie. no perturbations, simply a uniform sloping water table). Results from the residual analyses for these two cases are virtually identical to that from the cogenerated case. Furthermore, we examined the flux just around the boundary of the gridded domain and found that the sum of the boundary fluxes alone equaled 10% of the mean system flux for the cogenerated case and for cases *i*) and *ii*) described above (the boundary fluxes should sum to zero; the deterministic boundary fluxes summed to 0.0001% of the system flux). These global mass balance results are not too surprising, given the results from the statistical analyses.

Local Mass Balance Analysis

A more rigorous analysis involves an examination of the local mass balance errors; the magnitude of these errors can only be determined if the true solution to the system is known and the fluxes through individual grid blocks can be calculated. In order to obtain a solution which conserves mass, boundary conditions must be specified and the system must be solved deterministically. Although the introduction of boundary conditions is clearly not consistent with the manner in which the cosimulated heads are generated (ie. without boundary conditions), the deterministic solution provides a reasonable basis for comparison of the head patterns.

We experimented with a number of different types of boundary conditions but found that only the Dirichlet boundary conditions proved interesting. At first, Neumann boundary conditions were specified around the exterior of the grid (using the outer-two layers of cogenerated heads and transmissivities to calculate the fluxes) while holding one head in the center of the grid to "tie down" the solution. But since

the boundary fluxes do not sum to zero, the deterministic solution does not resemble the cogenerated solution. This is because the deterministic model is required to preserve mass and therefore will find the solution that balances mass by passing all the boundary flux imbalances through the one held node in the center of the grid. This results in a "groundwater mound or trough" around the held node and therefore the solution does not resemble the cogenerated head field.

Specifying Dirichlet boundary conditions around the perimeter of the grid naturally ties down the shape of the solution better and yields more interesting results. The cogenerated perturbed heads are added to the mean head to yield a total hydraulic head field which is used to set the boundary head values. A preconditioned conjugate-gradient matrix solver [*Golub and Van Loan*, 1983; *Gill, Murray and Wright*, 1981] is used to obtain the solution. After obtaining the deterministic solution, the mean head field is subtracted off to yield a "deterministic perturbed head field." This perturbed field is plotted together with the cosimulated heads in Figure 18. The spatial variability patterns of the two fields are very similar indicating that the cogenerated solution has, in general, "the right shape." Thus, flow through the cosimulated field would resemble the flow through the deterministic solution; one way of checking this would be to compare pathlines of particles traveling through each system (it is our intention to look at these kinds of comparative measures in future research).

Although the patterns of the cogenerated and deterministic perturbed head fields are similar, the variance of the deterministic heads is approximately an order of magnitude greater than the cogenerated heads; this is probably due to the relationship of the deterministic heads with transmissivities whereas the cogenerated heads are generated in conjunction with the logarithm of transmissivity. A three-dimensional perspective plot of the "pseudo-true" and cogenerated head fields is shown in Figure 19 illustrating the difference in the total amount of variability in these fields.

Having a "pseudo-true" solution available (the deterministic solution) allows us to calculate local mass balance errors by comparing the residuals in each grid block with

the actual local flux passing through the block as determined by the deterministic solution. For the cogenerated fields plotted in Figure 16, these errors range from $-25\%$ to $+25\%$; the errors are normally distributed with a mean of less than 1% and standard deviation of 6%. The meaning of the magnitude of these percentage figures is rather dubious, given the disparity in the variance of the two head fields and the fact that the cogenerated fields lack the proper statistical structure.

* * * * * * * * * * *

It is obvious, from results of the preceding analyses, that there is an unresolved problem associated with the computer implementation of the two-dimensional algorithms, or possibly with the theoretical development, although the latter is less likely. Hopefully, in future work, the "bugs" in these two-dimensional experiments will be worked out.

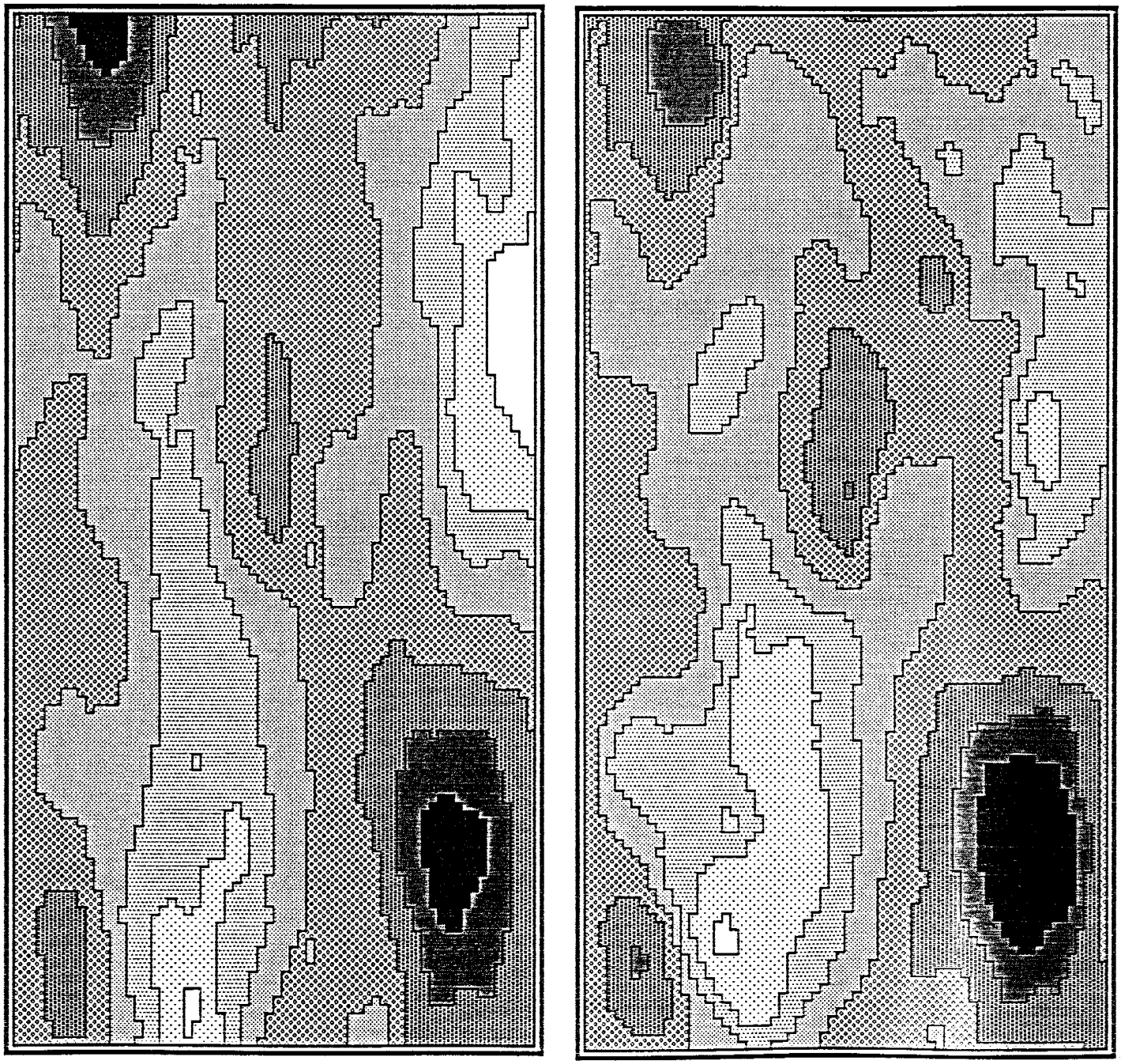


Figure 18. Shaded maps of head perturbations. Cogenerated head field (left) and deterministic solution with mean subtracted out (right).

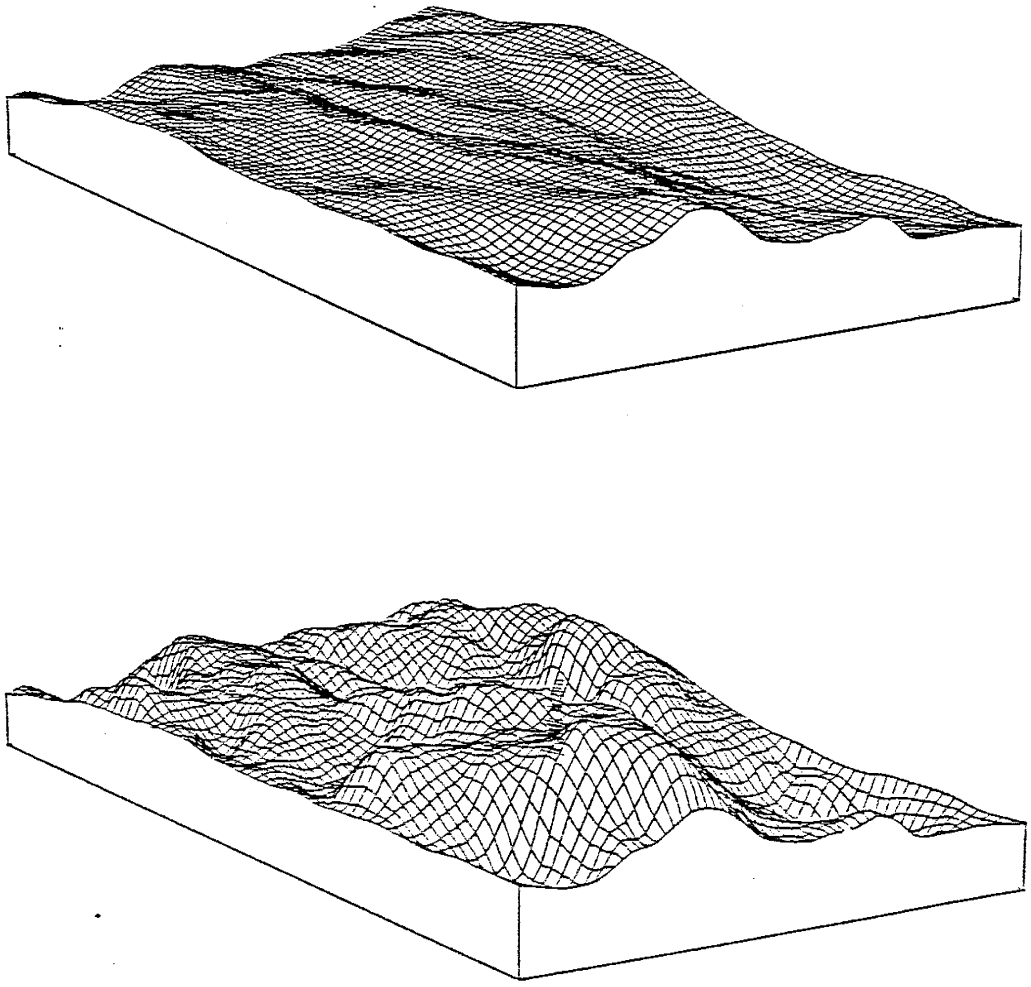


Figure 19. Perspective plot of the cosimulated (top) and deterministic (bottom) perturbed head fields illustrating the disparity in the variance between these two solutions.

# Summary and Conclusions

The theory for generating correlated random fields of head and log-transmissivity has been presented in a one-dimensional framework. The method is based on spectral-perturbation theory which is restricted to processes with low input variance and assumes that the cogenerated fields can both be represented as statistically homogeneous or stationary processes. Numerical algorithms for cogenerating discrete random fields are developed for one- and two-dimensions. A theoretical one-dimensional analysis of the discrete random field representations shows that the statistical properties are preserved and the governing differential equation is satisfied. The numerical algorithms are coded and discrete random fields of head and $lnT$ are cogenerated and analysed. The theory and procedural steps for *conditionally* cogenerating head and $lnT$ fields is also presented but is not implemented.

In one-dimension, the covariance properties of the discrete processes are well preserved. The perturbation analysis demonstrated the importance of carefully analysing the properties of the assumed form of the input ($lnT$ ) process. The $lnT$ hole-covariance function, while satisfying the requirements for stationary $lnT$ and head fields, is non-differentiable for continuous processes and yields a noisy $lnT$ field for the discrete case making it difficult to obtain accurate numerical derivatives. The bell-covariance function, which is continuously differentiable, satisfies the stationarity requirements and results in a much smoother more differentiable discrete $lnT$ process. The bell fields also do a much better job of satisfying the governing differential equation than the hole fields even at relatively coarse discretization levels and high input variances. An analysis to determine if the technique produces results which conserve mass is carried out via Darcy flux calculations in a finite difference framework. The results show that mass is conserved only for the low variance cases. Increasing the grid resolution improves the numerical derivative estimates and, consequently, also improves the mass balance. The results suggest that the linearized version of the

flow equation (1), obtained via the perturbation technique, is valid provided $\sigma_f^2 \ll 1$. A comparison with a deterministic solution shows that the spatial character of the cogenerated fields is correct except for localized areas where reverse head gradients appear in the high variance cases.

The two-dimensional results are, as of this writing, preliminary and of marginal value because the numerical algorithms have not yet been successfully implemented on the computer. Although we believe the theoretical development for the two-dimensional case to be correct, the results from numerical cosimulations so far have been somewhat disappointing. The discrete auto- and cross-covariance estimates do not accurately mimic the theoretical behavior, the global mass balance measures yield less than encouraging results, and interpretation of the local mass balance errors in quantitative terms is made difficult due to the disparity in the variances of the deterministic and cogenerated head solutions. Although the quantitative analyses fall far short of our expectations, the cogenerated fields, viewed qualitatively, do behave as expected. The auto- and cross-covariances have the right shape, and the fields exhibit the correct spatial character with respect to correlation scale, isotropy etc.

Results from these numerical analyses seem to suggest that higher order discretization may be necessary [see eg. *Dettinger and Wilson*, 1981]. The restriction of the perturbation approach to low input variances may mean that the governing equation needs to be solved exactly (ie. including the perturbation products) via convolution operations [see eg, *Emery*, 1988] in order to accomodate larger input variances. For the types of modeling problems envisioned for application of this technique (groundwater flow modeling, petroleum reservoir simulation), the stationarity requirement is not too restrictive.

The one-dimensional model is, in general, not a very good representation of any real (three-dimensional) process. The artificial complications it introduces (eg, no avenue for flow around low permeable zones, monotonic head gradient functions for steady state cases, etc.) must be given special consideration. The two-dimensional

framework, on the other hand, is more realistic but is also significantly more complicated and difficult to deal with, both in terms of developing and implementing the algorithms and in analysing the results. If the two-dimensional cosimulations had been more successful, additional performance measures, such as travel paths and travel times, would have been examined.

Under the assumption that the technique may yet be successful, the next step might be to cogenerate velocities directly. Even if mass will never be conserved with this method to the degree that it is in finite difference and finite element flow simulators, the technique may still be of practical value. For example, it might be used to generate an initial guess for a very large problem that will be solved deterministically via some iterative solution technique.

What is clear from the partial success of this work is that the method has tremendous potential and deserves further consideration.

# References


Abramowitz, M. and I. A. Stegun. 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards Applied Mathematics Series · 55, U.S. Government Printing Office, Wash. D.C.

Andersson, J. and A. M. Shapiro. 1983. Stochastic Analysis of One-Dimensional Steady State Unsaturated Flow: A Comparison of Monte Carlo and Perturbation Methods. *Water Resour. Res. 19*(1) 121–133.

Aziz, K. and A. Settari. 1979. *Petroleum Reservoir Simulation* Elsevier Science Publishing Co. Inc., New York.

Bakr, A. A. 1976. Stochastic Analysis of the Effects of Spatial Variations in Hydraulic Conductivity on Groundwater Flow, PhD. dissertation, N. Mex. Inst. of Mining and Technol., Socorro, New Mexico.

Bakr, A. A., Gelhar, L. W., Gutjahr, A. L. and J. R. MacMillan. 1978. Stochastic Analysis of Spatial Variability in Subsurface Flows 1. Comparison of One- and Three-Dimensional Flows. *Water Resour. Res. 14*(2) 263–271.

Barr, R. D. and P. W. Zehna. 1978. *Probability.* Brooks/Cole Publishing Company, Belmont, CA.

Clifton, P. M. and S. P. Neuman. 1982. Effects of Kriging and Inverse Modeling on Conditional Simulation of the Avra Valley Aquifer in Southern Arizona. *Water Resour. Res. 18*(4) 1215–1234.

Clifton, P. M., Sagar, B. and R. G. Baca. 1985. Stochastic Groundwater Traveltime Modeling Using a Monte Carlo Technique. Presented at the International Association of Hydrologists 17th International Congress, Tuscon, Arizona, January 7–12.

Coats, K. H., Dempsey, J. R., and J. H. Henderson. 1970. A New Technique for Determining Reservoir Description from Field Performance data. *Trans. SPE of AIME 249*, 66-74.

Coats, K. H., George, W. D. and B. E. Marcum. 1974. Three-dimensional Simulation of Steamflooding. *Trans. SPE of AIME 257*, 573–592.

Cole, C. R., Foote, H. P., Zimmerman, D. A., and C. S. Simmons. 1985. "Understanding, Testing and Development of Stochastic Approaches to Hydrologic Flow and Transport Through the use of the MultiGrid Method and Synthetic Data Sets." In: International Symposium on the Stochastic Approach to Subsurface Flow, Fontainbleau, France June 3–7, 1985.

Cole, C. R. and H. P. Foote. 1987. "Use of a Multigrid Technique to Study Effects of Limited Sampling of Heterogeniety on Transport Prediction." Presented at the Nat. Water Well Assoc. Conf., Solving Ground Water Problems with Models, Denver, Colorado, February 10–12.

Dagan, G. 1982. Stochastic Modeling of Groundwater Flow by Unconditional and Conditional Probabilities, 1. Conditional Simulation and the Direct Problem. *Water Resour. Res. 18*(4) 813–833.

Dagan, G. 1985. Stochastic Modeling of Groundwater Flow by Unconditional and Conditional Probabilities, The Inverse Problem. *Water Resour. Res. 21*(1) 65–72.

Delhomme, J. P. 1979. Spatial Variability and Uncertainty in Groundwater Flow Parameters: A Geostatistical Approach. *Water Resour. Res. 15*(2) 269–280.

de Marsily, G. 1986. *Quantitative Hydrogeology, Groundwater Hydrology for Engineers.* Academic Press, Inc., Orlando, Florida.

Dettinger, M. D. and J. L. Wilson. 1981. First Order Analysis of Uncertainty in Numerical Models of Groundwater Flow Part 1. Mathematical Development. *Water Resour. Res. 17*(1) 149–161.

Dudewicz, E. J. and T. G. Rally. 1981. *The Handbook of Random Number Generation and Testing with TESTRAND Computer Code.* American Sciences Press, Inc.

Emery, J. N. 1988. Investigation of an Iterative Solution for a Stochastic Differential Equation, Unpublished Independent Study, Math Dept., N. Mex. Inst. Mining and Technol., Socorro, New Mexico.

Freeze, R. A. 1975. A Stochastic-Conceptual Analysis of One-Dimensional Groundwater Flow in Nonuniform Homogeneous Media. *Water Resour. Res. 11*(5) 725–741.

Gelhar, L. W. 1974. Stochastic Analysis of Phreatic Aquifers. *Water Resour. Res. 10*(3) 539–545.

Gelhar, L. W., Wilson, J. L., and A. L. Gutjahr. 1979. Comments on 'Simulation of Groundwater Flow and Mass Transport Under Uncertainty' by D. H. Tang and G. F. Pinder. *Adv. Water Resour. Res. 2* p101.

Gelhar, L. W. 1986. Stochastic Subsurface Hydrology From Theory to Applications. *Water Resour. Res. 22*(9) 135S–145S.

Gill, P. E., Murray, W. and M. H. Wright. 1981. *Practical Optimization*, 401pp. Academic Press Inc., Orlando, FL.

Golub, G. H. and C. F. Van Loan. 1983. *Matrix Computations*, 476pp. John Hopkins University Press, Baltimore, MD.

Gutjahr, A. L., Gelhar, L. W., Bakr, A. A. and J. R. MacMillan. 1978. Stochastic Analysis of Spatial Variability in Subsurface Flows 2. Evaluation and Application. *Water Resour. Res. 14*(5) 953–959.

Gutjahr, A. L. and L. W. Gelhar 1981. Stochastic Models of Subsurface Flow: Infinite Versus Finite Domains and Stationarity. *Water Resour. Res. 17*(2) 337–350.

Hoeksema, R. J. and P. K. Kitanidis. 1984. An Application of the Geostatistical Approach to the Inverse Problem in Two-Dimensional Groundwater Modeling. *Water Resour. Res. 20*(7) 1003–1020.

Hoeksema, R. J. and P. K. Kitanidis. 1985a. Analysis of the Spatial Structure of Properties of Selected Aquifers. *Water Resour. Res. 21*(4) 563–572.

Hoeksema, R. J. and P. K. Kitanidis. 1985b. Comparison of Gaussian Conditional Mean and Kriging Estimation in the Geostatistical Solution of the Inverse Problem. *Water Resour. Res. 21*(6) 825–836.

Jacobson, E. A. 1985. A Statistical Parameter Estimation Method Using Singular Value Decomposition with Application to Avra Valley Aquifer in Southern Arizona. PhD dissertation, Univ. Arizona, Tuscon, AZ.

Journel, A. G. and CH. J. Huijbregts. 1978. *Mining Geostatistics* Academic Press Inc., New York, NY.

Kitanidis, P. K. and E. G. Vomvoris. 1983. A Geostatistical Approach to the Inverse Problem in Groundwater Modeling (Steady State) and One-Dimensional Simulations. *Water Resour. Res. 19*(3) 677–690.

Longman, I. M. 1956. Note on a method for computing infinite integrals of oscillatory functions, *Proc. Cambridge Philos. Soc., 52*(4) 764–768.

Lumley, J. L. and H. S. Panofsky. 1964. *The Structure of Atmospheric Turbulence*, 239pp. John Wiley, New York.

Mantoglou, A. and J. L. Wilson. 1981. Simulation of Random Fields with the Turning Bands Method, *Tech. Rep. 264*, Ralph M. Parsons Lab., Dep. of Civ. Engr., Mass. Inst. of Technol., Cambridge, Mass.

Mantoglou, A. and J. L. Wilson. 1982. The Turning Bands Method for Simulation of Random Fields Using Line Generation by a Spectral Method. *Water Resour. Res. 18*(5) 1379–1394.

McKay, D. H. 1988. A Fast Fourier Transform Method for Generation of Random Fields. M.S. Thesis, N. Mex. Inst. Mining and Technol., Socorro, New Mexico.

Mizell, S. A., Gutjahr, A. L. and L. W. Gelhar. 1982. Stochastic Analysis of Spatial Variability in Two-Dimensional Steady Groundwater Flow Assuming Stationary and Nonstationary Heads. *Water Resour. Res. 18*(4) 1053–1067.

Naff, R. L. and A. V. Vecchia. 1986. Stochastic Analysis of Three-Dimensional Flow in a Bounded Domain. *Water Resour. Res. 22*(5) 695–704.

Neuman, S. P. 1982. Statistical Characterization of Aquifer Heterogenieties: An Overview, in Recent Trends in Hydrogeology, *Geol. Soc. Am. Spec. Pap., 184*, 81-102.

Neuman, S. P. and S. Yakowitz. 1979. A Statistical Approach to the Inverse Problem of Aquifer Hydrology, 1, Theory. *Water Resour. Res. 15*(4) 845–860.

Neuman, S. P., Fogg, G. E. and E. A. Jacobson. 1980. A Statistical Approach to the Inverse Problem of Aquifer Hydrology, 2, Case Study. *Water Resour. Res. 16*(1) 33–58.

Sagar, B. 1978. Galerkin Finite Element Procedure for Analysing Flow Through Random Media. *Water Resour. Res. 14*(6) 1035–1044.

Smith, L. and R. A. Freeze. 1979a. Stochastic Analysis of Steady State Groundwater Flow in a Bounded Domain 1. One-Dimensional Simulations. *Water Resour. Res. 15*(3) 521–528.

Smith, L. and R. A. Freeze. 1979b. Stochastic Analysis of Steady State Groundwater Flow in a Bounded Domain 1. Two-Dimensional Simulations. *Water Resour. Res. 15*(6) 1543–1559.

Smith, L. and F. W. Schwartz. 1981. Mass Transport. 3. Role of Hydraulic Conductivity Data in Prediction. *Water Resour. Res. 17*(5) 1463–1479.

Tang, D. H. and G. F. Pinder. 1977. Simulation of Groundwater Flow and Mass Transport Under Uncertainty." *Adv. Water Resour. Res. 1* 25–30.

Townley, L. R. and J. L. Wilson. 1983. Conditional Second Moment Analysis of Groundwater Flow: the Cumulative Effects of Transmissivity and Head Measurements, in *Papers of the International Conference on Groundwater and Man*, Australian Government Publishing Service, Canberra, Australia.

Townley, L. R. and J. L. Wilson. 1985. Computationally Efficient Algorithms for Parameter Estimation and Uncertainty Propagation in Numerical Models of Groundwater Flow. *Water Resour. Res. 21*(12) 1851–1860.

Vanmarke, E. 1984. *Random Fields: Analysis and Synthesis.* Cambridge, Massachusuetts: MIT Press.

Warren, J. E. and H. S. Price. 1961. Flow in Heterogeneous Media. *Soc. Petrol. Engr. J., 1*, 153–169.

Wilson, J. L., Kitanidis, P. K. and M. Dettinger. 1978. State and Parameter Estimation in Groundwater Models, paper presented at AGU Chapman Conference held at Univ. Pittsburgh, May 22–24, 1978, in *Applications of Kalman Filter to Hydrology, Hydraulics, and Water Resources*, edited by Chao-Lin Chiu, pub. by Stochastic Hydraulics Program, Dept. Civil Engr., Univ. Pittsburgh, Pittsburgh, PA.

Zimmerman, D. A. and J. L. Wilson. 1988. Description of and User's Manual for TUBA: A Computer Code for Generating Two-Dimensional Random Fields via the Turning Bands Method. Hydrology Program, N. Mexico Inst. Mining and Technol., Socorro, NM. (in press)

## APPENDIX A

In this appendix, we show why only half the output sequence from a finite Fourier Transform should be used in the computations. We begin with the discrete form of the spectal representation of $f(x)$ given in (13) as

$$f(x_j) \;=\; \sum_{m=-M}^{M-1} e^{i\omega_m x_j}\, dZ_f(\omega_m) \tag{A1}$$

where both space and frequency are discretized; $\omega_m \equiv (m+\frac{1}{2})\Delta\omega$ (giving $M$ positive and $M$ negative frequencies), $\Delta\omega \equiv \Omega/N$ where $\Omega$ is the maximum frequency and $N$=2$M$, and $x_j \equiv j\Delta x$, $j$=-$M$,$M$-1. Standard finite Fourier Transform algorithms use the form $e^{i2\pi\omega_m x_j}$ for the Fourier kernel. With the relationship between space and frequency defined as $\Delta x = \frac{2\pi}{\Omega} = \frac{2\pi}{N\Delta\omega}$, (A1) becomes

$$f(j\Delta x) \;=\; \sum_{m=-M}^{M-1} e^{i(m+\frac{1}{2})\Delta\omega \frac{2\pi j}{N\Delta\omega}}\, dZ_f((m+\tfrac{1}{2})\Delta\omega)\,. \tag{A2}$$

We now look at the covariance between $f(x_j)$ and $f(x_k)$ by substituting (A2) into the definition of the covariance as

$$Cov\Big(f(x_j), f(x_k)\Big) \;\equiv\; E\Big[f(x_j)f(x_k)\Big]$$

$$= E\Bigg[\sum_{m_1=-M}^{M-1}\sum_{m_2=-M}^{M-1} e^{i(m_1+\frac{1}{2})\Delta\omega \frac{2\pi j}{N\Delta\omega}}\; e^{i(m_2+\frac{1}{2})\Delta\omega \frac{2\pi k}{N\Delta\omega}}$$

$$dZ_f((\mathrm{m}_1+\tfrac{1}{2})\Delta\omega)\, dZ_f((\mathrm{m}_2+\tfrac{1}{2})\Delta\omega)\Bigg]$$

$$= \sum_{m_1=-M}^{M-1}\sum_{m_2=-M}^{M-1} e^{i(m_1+\frac{1}{2})\Delta\omega \frac{2\pi j}{N\Delta\omega}}\; e^{i(m_2+\frac{1}{2})\Delta\omega \frac{2\pi k}{N\Delta\omega}} \tag{A3}$$

$$\mathrm{E}\Big[dZ_f((\mathrm{m}_1+\tfrac{1}{2})\Delta\omega)\, dZ_f((\mathrm{m}_2+\tfrac{1}{2})\Delta\omega)\Big]\,.$$

Using the result (14) from the spectral representation theorem, we find that (A3) is non-zero only when $dZ_f((\mathrm{m}_1+\frac{1}{2})\Delta\omega) = dZ_f^*((\mathrm{m}_2+\frac{1}{2})\Delta\omega)$ with $m_1 = m_2$; in other words, when $(\mathrm{m}_1+\frac{1}{2})\Delta\omega = -(\mathrm{m}_2+\frac{1}{2})\Delta\omega$. In this case, (A3) reduces to

$$E\Big[f(x_j)f(x_k)\Big] \;=\; \sum_{m=-M}^{M-1} e^{i(m+\frac{1}{2})\Delta\omega(j-k)\frac{2\pi}{N\Delta\omega}}\, S_{ff}((m+\tfrac{1}{2})\Delta\omega)\Delta\omega\,. \tag{A4}$$

If we define $x_{j-k} \equiv (j\text{-}k)\Delta x = x_j - x_k = (\ j\text{-}k)\frac{2\pi}{N\Delta\omega}$ and $\omega_m \equiv ((m+\frac{1}{2})\Delta\omega)$, then we can write (A4) as

$$E\big[f(x_j)f(x_k)\big] \;=\; \sum_{m=-M}^{M-1} e^{i\omega_m(j-k)\Delta x}\, S_{ff}(\omega_m)\Delta\omega$$

$$\approx \int_{-\Omega/2}^{\Omega/2} e^{i\omega x_{j-k}}\, S_{ff}(\omega)\, d\omega \;\equiv\; R_{ff}(x_{j-k}) \;=\; R_{ff}(x_j - x_k) \qquad \text{(A5)}$$

where $R_{ff}(x_j - x_k)$ is the covariance of $f(x)$ at a lag or separation distance of $x_j - x_k$. Recall that the $j$ and $k$ indices range from -$M$ to $M$-1, ie. there are $2M{=}N$ discrete values in the $f(x)$ sequence. Assume for the moment that $j, k \le M$ so that $E\big[f(x_j)f(x_k)\big] = R_{ff}(x_j - x_k)$ is the covariance between two points separated by a distance $|x_j - x_k|$ denoted as $|j-k|$. Since $|j-k| \le M$, the separation distance is less than or equal to half the length of the output sequence from the FFT. Now consider covariances between points spanning greater than half the sequence by calculating

$$E\big[f(x_{N-j})f(x_k)\big] \;=\; \sum_{m=-M}^{M-1} e^{i\,(m+\frac{1}{2})\Delta\omega(N-j-k)\frac{2\pi}{N\Delta\omega}}\, S_{ff}((m+\tfrac{1}{2})\Delta\omega)\Delta\omega \qquad \text{(A6)}$$

$$= \sum_{m=-M}^{M-1} e^{i\,(m+\frac{1}{2})2\pi}\, e^{-i\,(m+\frac{1}{2})\Delta\omega(j+k)\frac{2\pi}{N\Delta\omega}}\, S_{ff}((m+\tfrac{1}{2})\Delta\omega)\Delta\omega\,.$$

Noting that $e^{i\,(m+\frac{1}{2})2\pi} = -1$ and using the result (A5), (A6) becomes

$$E\big[f(x_j)f(x_k)\big] \;=\; -\,R_{ff}(-\,(x_j + x_k)\,) \;=\; -\,R_{ff}(x_j + x_k)\,. \qquad \text{(A7)}$$

In other words, the covariance behavior of the output from a finite Fourier Transform what might be described as "anti-periodic" where

$$R_{ff}(x_{N-j} - x_k) \;=\; -\,R_{ff}(x_j + x_k), \qquad \text{(A8)}$$

ie. covariances over large lags (greater than half the length of the sequence) are equal but opposite in sign to covariances at small lags. For example, suppose $\Delta x = 1$ so that $x_j = j\Delta x = j$ and let $N = 100$, $j = 10$, $k = 0$. Then via (A8) $R_{ff}(x_{N-j}) = -R_{ff}(x_j)$ or $R_{ff}(90) = -R_{ff}(10)$. Because of this anti-periodicity, we should only use half the output sequence from the finite Fourier Transform.

# APPENDIX B

In the following analysis, we show that the Telis $lnT$ covariance model will give rise to stationary heads for the two-dimensional flow problem described herein. Substituting the Telis spectrum, (71), into (12) gives the Telis covariance function,

$$R_f(\xi) \;=\; \sigma_f^2 \left\{ I_0(b\xi) - L_0(b\xi) \;+\; b\xi \left[ I_1(b\xi) - L_{-1}(b\xi) \right] \right\}, \tag{B1}$$

where $I_0, I_1$ are modified Bessel functions of the first kind of order zero and one, respectively, and $L_0, L_{-1}$ are modified Struve functions of order zero and minus one, respectively [*Abramowitz and Stegun*, 1964]. It may be of interest, in light of the peculiar form of this covariance function, to understand how the Telis spectrum–covariance pair came to be considered for this purpose:

In their papers on the Turning Bands method of random field generation, *Mantoglou and Wilson*, [1981,1982] describe mathematical relationships between the covariance functions of two or three-dimensional random fields and the covariance function of one-dimensional processes generated along the Turning Band lines. The Turning Bands method is an efficient random field generation algorithm in which the two or three-dimensional field is generated by projecting the values of one-dimensional processes (generated along lines radiating from an arbitrary origin in space) into the two or three-dimensional domain. In this theory, the relationship between one and three dimensional isotropic covariances is given by

$$R_1(\xi) \;=\; \frac{d}{d\xi}\left[\xi R_3(\xi)\right] . \tag{B2}$$

In Bakr's three dimensional analysis of flow [*Bakr et al*, 1978], the exponential covariance function,

$$R_3(\xi) \;=\; \sigma_f^2 \, e^{-|\xi|/\lambda}, \tag{B3}$$

was used to model the $lnT$ covariance behavior and gave rise to stationary heads. Substituting (B3) into (B2) and differentiating gives

$$R_1(\xi) \;=\; \sigma_f^2 (1 - |\xi|/\lambda)\, e^{-|\xi|/\lambda} \tag{B4}$$

which is the $lnT$ covariance model Bakr used in his one-dimensional analysis and it also gave rise to stationary heads. Mantoglou and Wilson derived the relationship between one-dimensional covariances along the Turning Band lines and the isotropic covariance of the two-dimensional field,

$$\int_0^r \frac{R_1(\xi)}{(r^2-\xi^2)^{1/2}}\,d\xi \;=\; \frac{\pi}{2}\,R_2(r) \tag{B5}$$

where $\xi$ and $r$ are the lag distances in one and two-dimensions respectively. They reasoned that if $R_1(\xi)$ given by (B4) yields stationary heads in one-dimension, and $R_3(\xi)$ given by (B3) yields stationary heads in three-dimensions, it is possible that $R_2(\xi)$, given by substituting (B4) into (B5) and integrating, will yield stationary heads in two-dimensions. They derived $R_2(\xi)$ and labeled it the Telis covariance function. Here we prove that this function does indeed lead to a stationary head field in two-dimensions.

*Mizell et al*, [1982] showed that in order to obtain a stationary head process for the two-dimensional flow problem described herein,

$$\int_0^\infty \xi\, R_{ff}(\xi)\,d\xi \;=\; 0 \tag{B6}$$

must be true. Here, $R_{ff}(\xi)$ is the isotropic $lnT$ covariance function. Substituting (B1) into (B6) leads to formidable mathematical difficulties in evaluating the integral in that equation. An easier approach is to examine the head process and show that it has a finite variance and is therefore stationary. The variance of the heads can be calculated by integrating the head spectrum over all frequencies; we must therefore show

$$\sigma_h^2 \;=\; \int_{-\infty}^{+\infty} S_{hh}(\vec{\omega})\,d\vec{\omega} \;<\; \infty \tag{B7}$$

to prove that the head process is stationary. The isotropic $lnT$ spectrum corresponding to the Telis covariance is given by

$$S_{ff}(\omega) \;=\; \frac{2\sigma_f^2 b}{\pi^2}\,\frac{\omega}{(b^2+\omega^2)^2} \tag{B8}$$

where $b$ is a correlation parameter which is related to the $\lambda$ in equation (B3) by $b \equiv 1/\lambda$. Substituting (B8) into (70) and writing $\omega_1 = \omega cos(\theta)$, and $\omega^2 = \omega_1^2 + \omega_2^2$, leads to the directionally dependent head spectrum

$$S_{hh}(\omega, \theta) = \frac{2\sigma_f^2 J_1^2 b}{\pi^2} \frac{cos^2(\theta)}{\omega(b^2 + \omega^2)^2} . \tag{B9}$$

Substituting (B9) into (B7) and integrating gives

$$\begin{aligned} \sigma_h^2 &= \int_0^{\infty} \int_0^{2\pi} S_{hh}(\omega, \theta)\, \omega d\omega d\theta \\ \sigma_h^2 &= \frac{2\sigma_f^2 J_1^2 b}{\pi^2} \int_0^{\infty} \frac{d\omega}{(b^2 + \omega^2)^2} \int_0^{2\pi} cos^2(\theta)\, d\theta \\ \sigma_h^2 &= \frac{\sigma_f^2 J_1^2}{2b^2}, \end{aligned} \tag{B10}$$

which shows that $\sigma_h^2 < \infty$, therefore the head process is stationary. We note the similarity of (B10) with the two-dimensional result of *Mizell et al*, [1982] (equation (19) combined with equation (10) in that paper),

$$\sigma_h^2 = \frac{2\sigma_f^2 J_1^2}{\alpha^2},$$

where the head variance was calculated using the Whittle spectrum A as a model for the $lnT$ process covariance.

# APPENDIX C

In this appendix, we make use of some results from *Mizell et al*, [1982] to derive the theoretical anisotropic head covariance function corresponding to the head spectrum given by equation (89),

$$S_{hh}(\vec{\omega}) \;=\; \frac{2\sigma_f^2 J_1^2 b}{\pi^2}\,\frac{\omega_1^2}{\omega^3(b^2+\omega^2)^2} \tag{C1}$$

where $\omega^2 = \omega_1^2 + \omega_2^2$. The two-dimensional head covariance function, $R_{ff}(\vec{\xi})$, is obtained by taking the Fourier Transform of the head spectrum as

$$R_{ff}(\vec{\xi}) \;=\; \int_{-\infty}^{+\infty} e^{i\vec{\omega}\cdot\vec{\xi}}\, S_{hh}(\vec{\omega})\, d\vec{\omega}\,. \tag{C2}$$

Using (A9) of *Mizell et al*, [1982] (with $k$ replaced by $\omega$), the head covariance function can be written as

$$R_{ff}(\xi_1,\xi_2) \;=\; \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{i\,(\omega_1\xi_1+\omega_2\xi_2)}\;\omega_1^2\, g(\omega)\, d\omega_1 d\omega_2 \tag{C3}$$

$$\begin{aligned} R_{ff}(\xi,\chi) \;=\; & (2\cos^2\chi - 1)\int_0^{\infty}\int_0^{2\pi} e^{i\omega\xi\cos\theta}\,\omega^3 g(\omega)\cos^2\theta\, d\theta d\omega \\ & +\; 2\pi\sin^2\chi \int_0^{\infty} J_0(\omega\xi)\,\omega^3 g(\omega)\, d\omega \end{aligned} \tag{C4}$$

where $\omega_1^2 g(\omega) \equiv S_{hh}(\vec{\omega})$ in (C3), and $\xi^2 = \xi_1^2 + \xi_2^2$. $J_0(\;)$ is a Bessel function of the first kind of order zero, and $\chi = \cos^{-1}(\xi_1/\xi)$ is the angle between the mean flow direction and $\xi$ (We note a typographical error in (A9) of that paper in which the exponent on the $\cos\theta$ term is 3 instead of 2). Substituting (C1) into (C2) and using (C3), then coverting to polar coordinates as in (C4) gives

$$\begin{aligned} R_{hh}(\xi,\chi) \;=\; & \frac{2\sigma_f^2 J_1^2 b}{\pi^2}\Bigg[(2\cos^2\chi - 1)\int_0^{\infty}\int_0^{2\pi} e^{i\omega\xi\cos\theta}\,\frac{\cos^2\theta}{(b^2+\omega^2)^2}\, d\theta d\omega \\ & +\; 2\pi\sin^2\chi \int_0^{\infty}\frac{J_o(\omega\xi)}{(b^2+\omega^2)^2}\, d\omega\Bigg] \end{aligned} \tag{C5}$$

The integration on $\theta$ in (C5) is evaluated as follows:

$$\text{let}\quad I \;\equiv\; \int_0^{2\pi} e^{i\omega\xi\cos\theta}\,\frac{d\theta}{\omega^2(b^2+\omega^2)^2} \;=\; \frac{2\pi J_o(\omega\xi)}{\omega^2(b^2+\omega^2)^2} \tag{C6}$$

where the eqality on the right is taken from equation (A1) of *Mizell et al*, [1982]. Differentiating $I$ twice with respect to $\xi$ and negating gives

$$-\frac{d^2I}{d\xi^2} = \int_0^{2\pi} e^{i\omega\xi cos\theta}\frac{cos^2\theta}{(b^2+\omega^2)^2}\,d\theta = \frac{-2\pi J_0''(\omega\xi)}{(b^2+\omega^2)^2} \tag{C7}$$

where $J_0''(\omega\xi) \equiv \frac{d^2}{d\xi^2}J_o(\omega\xi)$. Substituting (C7) into (C5) gives

$$\begin{aligned} R_{hh}(\xi,\chi) = \frac{4\sigma_f^2 J_1^2 b}{\pi}\Bigg[(1-2cos^2\chi)\int_0^\infty \frac{J_0''(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega \\ + sin^2\chi\int_0^\infty \frac{J_o(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega\Bigg]. \end{aligned} \tag{C8}$$

The following equalities, taken from *Abramowitz and Stegun*, [1964],

$$\begin{aligned} J_0'(x) &= -J_1(x) && \text{eqn 9.1.28 p.361} \\ J_1'(x) &= J_0(x) - \tfrac{1}{x}J_1(x) && \text{eqn 9.1.27 p.361} \end{aligned}$$

enable us to write

$$J_0''(x) = \frac{d}{dx}\Big[J_0'(x)\Big] = \frac{d}{dx}\Big[-J_1(x)\Big] = \frac{1}{x}J_1(x) - J_0(x),$$

so that (C8) becomes

$$\begin{aligned} R_{hh}(\xi,\chi) = \frac{4\sigma_f^2 J_1^2 b}{\pi}\Bigg[(1-2cos^2\chi)\int_0^\infty \frac{J_1(\omega\xi)}{\omega\xi(b^2+\omega^2)^2}\,d\omega \\ + cos^2\chi\int_0^\infty \frac{J_0(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega\Bigg]. \end{aligned} \tag{C9}$$

Using equation 11.4.45 of *Abramowitz and Stegun*, [1964, p.488] we write

$$\int_0^\infty \frac{J_0(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega = \frac{\pi}{2b}\Big[I_0(b\xi) - L_0(b\xi)\Big] \tag{C10}$$

where $I_0(\ )$ and $L_0(\ )$ are modified Bessel functions and modified Struve functions respectively, of order 0. Taking the derivative of (C10) with respect to $b$ gives

$$-2b\int_0^\infty \frac{J_0(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega = \frac{\pi}{2}\frac{d}{db}\Big[\frac{I_o(b\xi)}{b} - \frac{L_0(b\xi)}{b}\Big],$$

which, after carrying out the differentiation on the right hand side, simplifying, and dividing through by $-2b$, leads to

$$\int_0^\infty \frac{J_0(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega = \frac{\pi}{4}\left\{ \frac{\xi}{b^2}\left[L_0'(b\xi) - I_0'(b\xi)\right] + \frac{1}{b^3}\left[I_0(b\xi) - L_0(b\xi)\right] \right\}. \tag{C11}$$

Use of the following equalities from *Abramowitz and Stegun*, [1964],

| | | |
|---|---|---|
| $I_0'(x) = I_1(x)$ | eqn 9.6.27 | p.376 |
| $2L_0'(x) = \frac{2}{\pi} + L_{-1}(x) + L_1(x)$ | eqn 12.2.25 | p.498 |
| $L_{-1}(x) = \frac{2}{\pi} + L_1(x)$ | eqn 12.2.24 | p.498 |

enables us to write (C11) as

$$\int_0^\infty \frac{J_0(\omega\xi)}{(b^2+\omega^2)^2}\,d\omega = \frac{\pi\xi}{4b^2}\left\{ \frac{2}{\pi} + \left[L_1(b\xi) - I_1(b\xi)\right] + \frac{1}{b\xi}\left[I_0(b\xi) - L_0(b\xi)\right] \right\}. \tag{C12}$$

Using equation 11.4.45 of *Abramowitz and Stegun*, [1964] again, we write the first integral in (C9) as

$$\int_0^\infty \frac{J_1(\omega\xi)}{\omega\xi(b^2+\omega^2)^2}\,d\omega = \frac{\pi}{2b^2\xi}\left[I_1(b\xi) - L_1(b\xi)\right]. \tag{C13}$$

Taking the derivative of (C13) with respect to $b$ and reducing in the same fashion as that shown in equations (C10) $\rightarrow$ (C11) leads to

$$\int_0^\infty \frac{J_1(\omega\xi)}{\omega\xi(b^2+\omega^2)^2}\,d\omega = \frac{\pi}{4b^3}\left\{ \left[L_1'(b\xi) - I_1'(b\xi)\right] + \frac{2}{b\xi}\left[I_1(b\xi) - L_1(b\xi)\right] \right\}. \tag{C14}$$

Using the following identities from *Abramowitz and Stegun*, [1964],

| | | |
|---|---|---|
| $I_1'(x) = I_0(x) - \frac{1}{x}I_1(x)$ | eqn 9.6.26 | p.376 |
| $2L_1'(x) = \frac{x}{2}\frac{\pi^{-1/2}}{\Gamma(5/2)} + L_0(x) + L_2(x)$ | eqn 12.2.5 | p.498 |
| $L_2(x) = L_0(x) - \frac{2}{x}L_1(x) - \frac{x}{2}\frac{\pi^{-1/2}}{\Gamma(5/2)}$ | eqn 12.2.4 | p.498 |

enables us to write $L_1'(x) = L_0(x) - \frac{1}{x}L_1(x)$ so that (C14) becomes

$$\int_0^\infty \frac{J_1(\omega\xi)}{\omega\xi(b^2+\omega^2)^2}\,d\omega \;=\; \frac{\pi}{4b^3}\left\{\left[L_0(b\xi) - I_0(b\xi)\right] \;+\; \frac{3}{b\xi}\left[I_1(b\xi) - L_1(b\xi)\right]\right\}. \tag{C15}$$

Finally, the two-dimensional head covariance function is obtained by substituting (C15) and (C12) into (C9) yielding

$$R_{hh}(\xi,\chi) \;=\; \frac{\sigma_f^2 J_1^2}{b^2}\left\{\frac{2b\xi}{\pi}cos^2\chi \;+\; (1-3cos^2\chi)\left[L_0(b\xi) - I_0(b\xi)\right] \;+\; \left(cos^2\chi\left[b\xi + \frac{6}{b\xi}\right] - \frac{3}{b\xi}\right)\left[L_1(b\xi) - I_1(b\xi)\right]\right\}. \tag{C16}$$

The variance of the head process is given by $R_{hh}(0,0)$ as

$$R_{hh}(0,0) \;=\; \frac{\sigma_f^2 J_1^2}{b^2}\left\{2 + \lim_{\xi\to 0}\left(\frac{3}{b\xi}\right)\left[L_1(b\xi) - I_1(b\xi)\right]\right\} \tag{C17}$$

where we have used Figure 9.7 and Figure 12.4 of *Abramowitz and Stegun*, [1964] to evaluate $L_0(0)$ and $I_0(0)$. Using the power series expansions for $L_1(x)$ and $I_1(x)$ given by equations 9.6.10 and 12.2.1 of *Abramowitz and Stegun*, [1964], we examine the behavior of $L_1(x) - I_1(x)$ for $x \ll 1$ and find that

$$L_1(x) - I_1(x) \;\approx\; \frac{x^2/2}{\Gamma(3/2)\,\Gamma(5/2)} \;-\; \frac{x/2}{\Gamma(2)}$$

so that

$$\lim_{x\to 0}\left(\frac{3}{x}\right)\left[L_1(x) - I_1(x)\right] \;=\; -\frac{3}{2}.$$

Hence, (C17) becomes

$$R_{hh}(0,0) \;\equiv\; \sigma_h^2 \;=\; \frac{\sigma_f^2 J_1^2}{2b^2}. \tag{C18}$$

Note that this result is consistent with equation (B10) where $\sigma_h^2$ was derived simply by integrating the spectral density function of the head process.

It appears, from (C16), that $R_{hh}(\xi,\chi)$ is linear in $\xi$, so we must check to see if $R_{hh}(\xi,\chi) \to 0$ as $\xi \to \infty$. The $\xi$ term that twice appears in the numerator position of (C16) as the coefficient to $cos^2\chi$ results from its being factored algebraically, but

originates in the expression (C12). Thus, using (C12), we examine the behavior of $R_{hh}(\xi,\chi)$ as $\xi \to \infty$ by evaluating

$$\lim_{\xi\to\infty} b\xi \left[\frac{2}{\pi} + L_1(b\xi) - I_1(b\xi)\right]. \tag{C19}$$

Using the asymptotic expansion for large arguments of $L_\nu(z)-I_\nu(z)$ given by equation 12.2.6 of *Abramowitz and Stegun*, [1964, p.498], we find that for $\nu = 1$

$$\lim_{z\to\infty} L_1(z) - I_1(z) \;=\; \frac{(-1)\,\Gamma(1/2)}{\pi\,\Gamma(3/2)} \;=\; -\frac{2}{\pi}, \tag{C20}$$

therefore,

$$\lim_{\xi\to\infty} R_{hh}(\xi,\chi) \;=\; 0. \tag{C21}$$

# APPENDIX D

In this appendix, we derive the theoretical cross-covariance function between the head and $lnT$ fields for the two-dimensional case. We begin by writing the two-dimensional version of (12) for the cross-covariance as

$$R_{fh}(\vec{\xi}) \;=\; \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{i\,\vec{\omega}\cdot\vec{\xi}}\, S_{fh}(\vec{\omega})\, d\vec{\omega}. \tag{D1}$$

Next we use $\chi$ to represent the angle between $\vec{\xi}$ and the mean flow direction, $\xi_1$, and write (D1) as

$$R_{fh}(\xi,\chi) \;=\; \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{i\xi(\omega_1 cos\chi+\omega_2 sin\chi)}\, S_{fh}(\omega_1,\omega_2)\, d\omega_1 d\omega_2 \tag{D2}$$

where $\xi_1 = \xi cos\chi$, $\xi_2 = \xi sin\chi$, and $\xi = |\vec{\xi}|$. It is perhaps easiest to evaluate (D2) by changing to polar coordinates, whereby we let $\omega_1 \equiv \omega cos\alpha$, $\omega_2 \equiv \omega sin\alpha$, where $\omega = |\vec{\omega}|$, and $\alpha$ is the angle between $\vec{\omega}$ and $\omega_1$; the $\omega_1$ and $\omega_2$ frequency directions correspond with the $\xi_1$ and $\xi_2$ spatial directions respectively (see Figure D.1). With these substitutions, (D2) becomes

$$R_{fh}(\xi,\chi) \;=\; \int_{0}^{\infty}\int_{0}^{2\pi} e^{i\omega\xi cos(\chi-\alpha)}\, S_{fh}(\omega\, cos\alpha,\, \omega\, sin\alpha)\, \omega d\omega d\alpha. \tag{D3}$$

We now let $\theta$ be the angle between $\vec{\xi}$ and $\vec{\omega}$, ie. $\theta = \chi - \alpha$, and write

$$R_{fh}(\xi,\chi) \;=\; \int_{0}^{\infty}\int_{0}^{2\pi} e^{i\omega\xi cos\theta}\, S_{fh}\Big(\omega\, cos(\chi-\theta),\, \omega\, sin(\chi-\theta)\Big)\, \omega d\omega d\theta. \tag{D4}$$

The cross-spectrum can be expressed via (91) in the form $S_{fh}(\vec{\omega}) = \frac{-Ji\omega_1}{\omega^2} S_{ff}(\omega)$ and substituted into (D4) giving

$$\begin{aligned} R_{fh}(\xi,\chi) \;&=\; -Ji \int_{0}^{\infty}\int_{0}^{2\pi} e^{i\omega\xi cos\theta}\, cos(\chi-\theta)\, S_{ff}(\omega)\, d\omega d\theta. \\ &=\; -Ji \int_{0}^{\infty}\int_{0}^{2\pi} e^{i\omega\xi cos\theta}\, \Big[cos\chi\, cos\theta + sin\chi\, sin\theta\Big]\, S_{ff}(\omega)\, d\omega d\theta. \end{aligned} \tag{D5}$$

where we have used $\omega_1 = \omega\, cos(\chi - \theta) = \omega\big[cos\chi cos\theta + sin\chi sin\theta\big]$. We now do the $\theta$-integration by writing two integrals, each of the form

$$\int_0^{2\pi} e^{i\omega\xi cos\theta}\, g(\theta)\, d\theta \;=\; \int_0^{\pi} e^{i\omega\xi cos\theta}\, g(\theta)\, d\theta \;+\; \int_{\pi}^{2\pi} e^{i\omega\xi cos\theta}\, g(\theta)\, d\theta \qquad \text{(D6)}$$

where $g(\theta) = cos\chi cos\theta$ or $g(\theta) = sin\chi sin\theta$. By substituting $\phi = 2\pi - \theta$ in the right-most integral in (D6), we obtain

$$\int_{\pi}^{2\pi} e^{i\omega\xi cos\theta}\, g(\theta)\, d\theta \;=\; \int_0^{\pi} e^{i\omega\xi cos(2\pi-\phi)}\, g(2\pi - \phi)\, d\phi. \qquad \text{(D7)}$$

Since the $sin(2\pi - \phi) = -sin\phi$, the $\theta$ integrals in (D6) cancel when $g(\theta) = sin\chi sin\theta$. Similarly, since $cos(2\pi - \phi) = cos\phi$, the $\theta$ integrals in (D6) add when $g(\theta) = cos\chi cos\theta$ giving $2\int_0^{\pi} e^{i\omega\xi cos\theta} cos\chi \cos\theta\, d\theta$; thus, (D5) becomes

$$R_{fh}(\xi, \chi) \;=\; -2Ji\, cos\chi \int_0^{\infty}\int_0^{\pi} e^{i\omega\xi cos\theta}\, cos\theta\, S_{ff}(\omega)\, d\theta d\omega. \qquad \text{(D8)}$$

Using equation 9.1.21 of *Abramowitz and Stegun*, [1964, p.360], we write

$$\int_0^{\pi} e^{i\omega\xi cos\theta}\, cos\theta\, d\theta \;=\; \frac{\pi}{i} J_1(\omega\xi) \qquad \text{(D9)}$$

where $J_1(\;\;)$ is a Bessel function of the first kind of order 1 (not to be confused with the mean hydraulic gradient, $J$). Thus, (D8) becomes

$$R_{fh}(\xi, \chi) \;=\; -2\pi J cos\chi \int_0^{\infty} J_1(\omega\xi) S_{ff}(\omega)\, d\omega. \qquad \text{(D10)}$$

Substituting the $lnT$ spectrum (71) into (D10) gives

$$R_{fh}(\xi, \chi) \;=\; \frac{-4J\sigma_f^2 b}{\pi} cos\chi \int_0^{\infty} J_1(\omega\xi) \frac{\omega}{(b^2 + \omega^2)^2}\, d\omega. \qquad \text{(D11)}$$

From equations 9.1.5 and 11.4.45 of *Abramowitz and Stegun*, [1964, p.358 and 488], we have

$$\int_0^{\infty} \frac{J_{-1}(at)}{t^{-1}(z^2 + t^2)}\, dt \;=\; -\int_0^{\infty} J_1(at)\, \frac{t}{(z^2 + t^2)}\, dt. \qquad \text{(D12)}$$

$$\;=\; \frac{\pi}{2}\big[I_{-1}(az) - L_{-1}(az)\big] \qquad \text{(D13)}$$

where $I_{-1}(\ )$ and $L_{-1}(\ )$ are modified Bessel functions and modified Struve functions of order $-1$ respectively. Taking the derivative of (D12) with respect to $z$ and using that result in (D11) with $a \leftrightarrow \xi$, $t \leftrightarrow \omega$, and $z \leftrightarrow b$, gives

$$\int_0^\infty J_1(\omega\xi)\,\frac{\omega}{(b^2+\omega^2)^2}\,d\omega \;=\; -\frac{\pi}{4b}\,\frac{d}{db}\Big[I_{-1}(b\xi) - L_{-1}(b\xi)\Big]. \tag{D14}$$

Using the following relations from *Abramowitz and Stegun*, [1964],

| | | |
|---|---|---|
| $I_{-n}(x) = I_n(x)$ | eqn 9.6.6 | p.375 |
| $I_1'(x) = I_0(x) - \frac{1}{x} I_1(x)$ | eqn 9.6.26 | p.376 |
| $2L_{-1}'(x) = L_{-2}(x) + L_0(x) + \frac{1}{x}\frac{2}{\pi}$ | eqn 12.2.5 | p.498 |
| $L_{-2}(x) = -\frac{2}{x} L_{-1}(x) + L_0(x) + \frac{1}{x}\frac{2}{\pi}$ | eqn 12.2.4 | p.498 |
| $L_1(x) = L_{-1}(x) - \frac{2}{\pi}$ | eqn 12.2.4 | p.498 |

enables us to evaluate the derivatives on the right-hand side of (D14) and obtain

$$\frac{d}{db}\Big[I_{-1}(b\xi) - L_{-1}(b\xi)\Big] \;=\; \xi\Bigg\{ I_0(b\xi) - L_0(b\xi) \;-\; \frac{1}{b\xi}\Big[ I_1(b\xi) - L_1(b\xi)\Big]\Bigg\}. \tag{D15}$$

Combining the constant terms in (D11) and (D14) with the result (D15) finally leads to the cross-covariance function,

$$R_{fh}(\xi,\chi) \;=\; J\sigma_f^2 \xi\, cos\chi \Bigg\{ I_0(b\xi) - L_0(b\xi) \;-\; \frac{1}{b\xi}\Big[ I_1(b\xi) - L_1(b\xi)\Big]\Bigg\}. \tag{D16}$$

At $\xi = 0$ $L_0(0) = L_1(0) = I_1(0) = 0$, $I_0(0) = 1$ so that $R_{fh}(0,\chi) = 0$, indicating that the heads and $lnT$ are uncorrelated at zero lag. For large separation distances we find, using the asymptotic expansion for large arguments of $L_\nu(x) - I_{-\nu}(x)$, equation 12.2.6 of *Abramowitz and Stegun*, [1964, p.498], that

$$\lim_{\xi\to\infty}\; I_1(x) - L_1(x) \;\approx\; \frac{2}{\pi} \tag{D17}$$

while

$$\lim_{\xi\to\infty}\; I_0(x) - L_0(x) \;\approx\; \frac{1}{x}\frac{2}{\pi}. \tag{D18}$$

Therefore,

$$\lim_{\xi\to\infty}\; R_{fh}(\xi,\chi) \;=\; 0. \tag{D19}$$

ie. the cross-covariance dies out at large lag distances as expected.

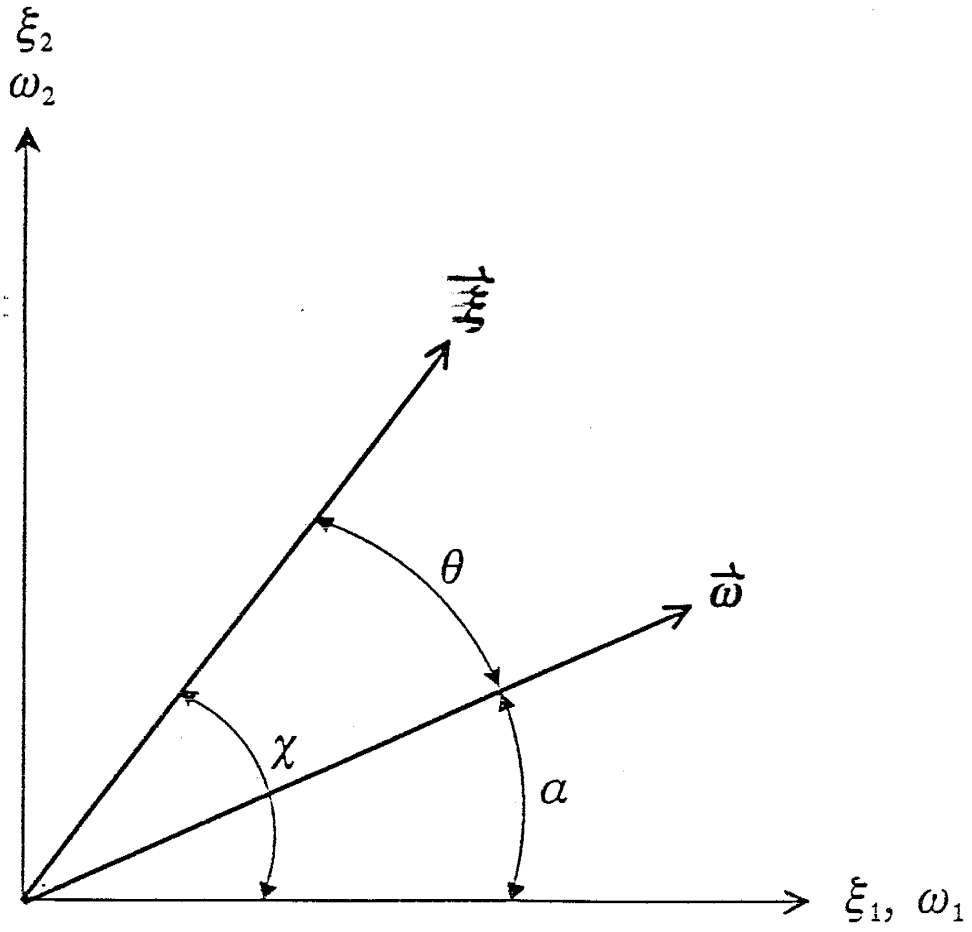



Figure D.1. Coordinate transformation definition sketch; schematic of frequency ($\omega$) and spatial ($\xi$) coordinate systems superimposed.

# Appendix E

Only the cogeneration codes and ancillary numerical analysis programs are listed here. The plotting programs are not listed except where they are imbedded within a "number crunching" code. Almost all of these routines take advantage of Fortran enhancements available on the VAX VMS Fortran version 4.1 compiler.

## Appendix E Table of Contents

```
      PROGRAM COGEN_1DP
C-----------------------------------------------------------------------
C     SUFFIX "P"  -->  DOUBLE PRECISION FFT
C     COSIMULATION OF 1D RANDOM FIELDS - HOLE AND BELL-HOLE FUNCTIONS FOR LN(T)

      PARAMETER   (MXX=262150)
      REAL         SF(MXX),  SH(MXX), RX(0:MXX), CP(0:MXX)
      REAL         ZF(MXX),  ZH(MXX), RF(0:MXX), RH(0:MXX)
      REAL        SMF(MXX), SMH(MXX),  SMX(MXX), ER(MXX)
      COMPLEX*16  DZF(MXX), DZH(MXX)

      COMMON    /PARAMS/ VLNT,GRAD,DELX,FMAX,DELK,NHAR,NMAX,CMAX,APAR
      COMMON    /IFLAGS/ NBCN,IDIF,IHOL


C     READ INPUT PARAMETERS
      CALL RDINPT(NSIM,NPTS,MLAG)

C     BEGIN THE SIMULATIONS ...
      DO ISIM=1,NSIM
COMT    CALL PROGSS(ISIM,NSIM,'RANDOM FIELD SIMULATION NUMBER')

C       GOGENERATE RANDOM FIELDS
        CALL FFTCOG(ZF,ZH,DZF,DZH)

COMT    INTERMEDIATE COVARIANCE CALCULATIONS
COMT    CALL COVFHX(ZF,ZH,SMF,SMH,RF,RH,SMX,RX,CP,NPTS,MLAG)

C       CALCULATE MEAN AND VARIANCE OF RANDOM FIELDS
        CALL BARVAR(ZF,NMAX,FBAR,FVAR)
        CALL BARVAR(ZH,NMAX,HBAR,HVAR)
        WRITE( *,10) FBAR,HBAR,FVAR,HVAR
        WRITE(50,10) FBAR,HBAR,FVAR,HVAR
10      FORMAT('  ------------------------------------------------',
     1        /'  FBAR =',F8.2,'       HBAR =',F8.2,
     1        /'  FVAR =',F8.2,'       HVAR =',F8.2)

C       CHECK PERTURBATION EQTN AND MASS BALANCE
        CALL CHKPTB(ZF,ZH)
COMT    WRITE(50,'(1H1)')
        CALL CHKMAS(ZF,ZH)
      END DO

COMT  CALCULATE AND PRINT THE AUTO AND CROSS COVARIANCES
COMT  CALL RRRFHX(SMF,SMH,SMX,RF,RH,RX,CP,MLAG)
COMT  CALL PNTCOV(RF,RH,RX)

      STOP
      END




      SUBROUTINE PROGSS(K,KMAX,MSG)
C-----------------------------------------------------------------------
C     REPORT DO-LOOP COMPUTATION PROGRESS

      CHARACTER  MSG*(*)

      PCT = 100 * FLOAT(K)/FLOAT(KMAX)
      IPC = INT(PCT)
      PCD = PCT - PCL

      IF(MOD(IPC,5).EQ.0 .AND. PCD.GT.2) THEN
        PCT = AMIN1(PCT,99.9)
        PCL = PCT
        WRITE(6,10) MSG,INT(PCT)
10      FORMAT(1H$,A,' ... (',I3,' % ) ... ')
        IF(PCT.EQ.99.9) PCL = 0.0
      END IF

      RETURN
      END
```

```
      SUBROUTINE RDINPT(NSIM,NPTS,MLAG)
C-----------------------------------------------------------------------
C     READ INPUT PARAMETERS FOR 1D COGENERATION MODEL

      PARAMETER (PI=3.141592654)
      CHARACTER  OFILE*50,CFCN*12,COVF(2)*4,TNSF(2)*12
      COMMON    /PARAMS/ VLNT,GRAD,DELX,FMAX,DELK,NHAR,NMAX,CMAX,APAR
      COMMON    /IFLAGS/ NBCN,IDIF,IHOL
      COMMON    /ALGDLX/ NPRIM,IJOK
      DATA       COVF /'HOLE','BELL'/, VLNT,GRAD,APAR /1.0, 1.0, 1.0/
      DATA       TNSF /'(CONTINUOUS)','( DISCRETE )'/


      WRITE(*,*)'(1) - THE  HOLE FUNCTION FOR LN(T)'
      WRITE(*,*)'(2) - BELL-HOLE FUNCTION FOR LN(T)'
      READ (*,*) IHOL

C     WRITE(*,*)'ENTER THE LN(T) VARIANCE, THE HEAD GRADIENT, AND "A"'
C     READ(*,*)  VLNT,GRAD,APAR
      WRITE(*,*)'ENTER THE LN(T) VARIANCE'
      READ (*,*) VLNT

      WRITE(*,*)'ENTER MAX FREQUENCY, NHAR AND N-PRIME'
      READ(*,*)  FMAX,NHAR,NPRIM

      WRITE(*,*)'(1) - i*J/k  (continuous case)'
      WRITE(*,*)'(2) - cos etc. (discrete case)'
      READ(*,*) IJOK

C     DELX, DELK AND NMAX PARAMETERS FOR FFT METHOD
C     DELX = 2.0*PI/FMAX
      DELK = FMAX/FLOAT(NHAR)
      DELX = 2.0*PI / (FLOAT(NPRIM)*DELK)
      NMAX = NHAR/2
      NSIM = 1     ! MAKE PLURAL FOR COVARIANCE CALCULATIONS
      IDIF = 0     ! 0=CENTRAL DIFFERENCES, 1=FORWARD DIFFERENCES

      WRITE(*,*)'ENTER RANDOM NUMBER GENERATOR SEED'
      READ(*,*)  ISEED

      WRITE(*,*)'ENTER THE NAME OF THE OUTPUT DATA FILE'
      READ(*,'(A)') OFILE
      OPEN(UNIT=95,NAME=OFILE,STATUS='NEW',FORM='UNFORMATTED')
      WRITE(95) NMAX-2,VLNT,GRAD,APAR,DELX
      IDOT = INDEX(OFILE,'.')
      OFILE = OFILE(1:IDOT) // 'LST'
      OPEN(UNIT=50,NAME=OFILE,STATUS='NEW')

C     CALCULATE INTERNAL PARAMETERS
      CALL URNIT(ISEED)                  ! INITIALIZE RANDOM NUMBER GENERATOR
      NPPC = APAR/DELX + 0.5             ! NBR PTS/CL (B=1)
      DELK = FMAX/FLOAT(NHAR)            ! FREQUENCY SPACING
      CMAX = 6.25*APAR                   ! MAXIMUM NUMBER OF CORRELATION LENGTHS
      MLAG = CMAX/DELX + 1               ! MAXIMUM LAG CORRESPONDING TO CMAX
      NCOR = NMAX/NPPC                   ! NUMBER OF CORRELATION LENGTHS / REALIZATION
      NBCN = 50                          ! NUMBER OF "CHECK" NODES FOR LOCAL MASS BALANCE
      FLUX = EXP(-11.51+VLNT/2.)*GRAD    ! MEAN FLUX
      NPTS = NMAX
      CFCN = COVF(IHOL) // ' COV FCN'

      WRITE( *,10) ISEED,CFCN,APAR,TNSF(IJOK),FMAX,DELK,DELX,NPPC,VLNT,
     1             GRAD,FLUX,NHAR,NMAX,NPRIM,NCOR
      WRITE(50,10) ISEED,CFCN,APAR,TNSF(IJOK),FMAX,DELK,DELX,NPPC,VLNT,
     1             GRAD,FLUX,NHAR,NMAX,NPRIM,NCOR

10    FORMAT('  SEED =',I10, 24X,A /'  APAR =',F8.2, 26X,A,
     1  /'  FMAX =',F8.2   /'  DELK =',F10.4,
     1  /'  DELX =',F10.4,'    (NPPC =',I7,1H),
     1  /'  VLNT =',F8.2,  /'  GRAD =',F8.2,
     1  /'  MFLX =',1PE12.2,'  (MLnT = -11.51)',
     1  /'  NHAR =',I8,2X,'    (NPTS =',I7,')  (NPRIM =',I7,1H),
     1  /'  NCOR =',I8)

      RETURN
      END
```

```
      SUBROUTINE FFTCOG(ZF,ZH,DPZF,DPZH)
C-----------------------------------------------------------------------
C     CO-GENERATION OF HEAD AND LOG(T) FIELDS FOR 1D FLOW MODEL
C     CORRELATION PARAMETER L=1   (SEE BAKR, WRR 14(2) 1978)

      PARAMETER   (PI=3.141592654)
      REAL         ZF(*), ZH(*)
      COMPLEX     DZF,DZH, i, RNBS, CMPC, CNGL
      COMPLEX*16  DTOP, DPZF(*),DPZH(*)
      REAL*8      DARG,DBOT

      COMMON     /PARAMS/ VLNT,GRAD,DELX,FMAX,DELK,NHAR,NMAX,CMAX,APAR
      COMMON     /IFLAGS/ NBCN,IDIF,IHOL
      COMMON     /ALGDLX/ NPRIM,IJOK


      i    = (0.,1.)                                ! FOR COMPLEX ARITHMETIC
      ASQD = APAR*APAR                              ! CORR PARAMETER SQUARED
      CNST = 2.0*VLNT*APAR/PI                       ! CONSTANT TERM
      CMPC = -i*PI/FLOAT(NHAR)                      ! COMPLEX CONSTANT
      C2 = GRAD*DELX/2.0                            ! TRNSFR FCN CONSTANT TERM
      IF(IHOL.EQ.2) THEN                            ! BELL-HOLE FUNCTION
        R2PI = SQRT(2.0*PI)                         ! SQRT OF 2 PI
        ACUB = APAR*ASQD                            ! CORR PARAMETER CUBED
        CNST = VLNT*ACUB/R2PI                       ! CONSTANT TERM
      END IF

      DO M=1,NHAR
        OMEGA = (FLOAT(M)-0.5)*DELK                 ! FREQUENCY
        OMSQD = OMEGA*OMEGA                         ! FREQUENCY SQUARED
        OMSTF = OMSQD/(ASQD+OMSQD)**2               ! OMEGA STUFF
        IF(IHOL.EQ.2) THEN                          ! BELL-HOLE FUNCTION
          ARG   = 0.5*ASQD*OMSQD                    ! EXP ARGUMENT
          OMSTF = OMSQD*EXP(-ARG)                   ! OMEGA STUFF
        END IF
        SQDF = SQRT(CNST*OMSTF*DELK)                ! SQRT(DF)
        RNBS = URNV2() + i*URNV2()                  ! RANDOM NUMBERS
        DZF  = CEXP(M*CMPC) * SQDF*RNBS             ! DZF PROCESS
        DPZF(M) = DCMPLX(DZF)                       ! DOUBLE PRECISION DZF
        IF(IJOK.EQ.1) THEN
          DZH =(GRAD*i/OMEGA)* DZF                  ! DZH PROCESS (CONTINUOUS)
        ELSE
          SSS = SIN(OMEGA*DELX)                     ! DISCRETE CASE
C         CCC = COS(OMEGA*DELX)                     ! CCC-1. NO GOOD IN
C         DZH = DZF * (C2*i*SSS) / (CCC-1.)         ! SINGLE PRECISION, SO ...
          DARG = DBLE(OMEGA) * DBLE(DELX)
          DTOP = DCMPLX( C2*i*SSS )
          DBOT = DCOS(DARG) - 1.D0
          DZH  = DZF * DTOP / DBOT                  ! DZH PROCESS (DISCRETE)
        END IF
        DPZH(M) = DCMPLX(DZH)                       ! DOUBLE PRECISION DZH
COMT    CALL PROGSS(M,NHAR,'FFT FIELD GENERATION')
      END DO

C     PAD OUT THE SEQUENCE WITH ZEROS, THEN TRANSFORM
      DO M=NHAR+1,NPRIM
        DPZF(M) = DCMPLX(0.D0,0.D0)
        DPZH(M) = DCMPLX(0.D0,0.D0)
      END DO
      CALL DFFT(DPZF,NPRIM,-1)
      CALL DFFT(DPZH,NPRIM,-1)

      DO M=1,NHAR
        ZF(M) = 2.0 * REAL(DPZF(M))
        ZH(M) = 2.0 * REAL(DPZH(M))
        IF(IJOK.EQ.1) ZH(M) = -ZH(M)
      END DO

      RETURN
      END
```

```
      SUBROUTINE CHKPTB(ZF,ZH)
C-----------------------------------------------------------------------
C     CHECK MASS BALANCE CHECK FOR 1D FLOW MODEL  (PERTURBATION EQTN CHECK)

      REAL      ZF(*), ZH(*),   PF(-1:1), PH(-1:1)
      INTEGER   LOC(1000)
      COMMON   /PARAMS/ VLNT,GRAD,DELX,FMAX,DELK,NHAR,NMAX,CMAX,APAR
      COMMON   /IFLAGS/ NBCN,IDIF,IHOL

C     PICK OUT ARBITRARY NODES FOR LOCAL MASS BALANCE CHECK
      CALL PKNODS(LOC,NMAX,NBCN)
      NXT = 1
      WRITE(50,12)
12    FORMAT(/30X,'Perturbation Equation Check',
     1      //'  Node    d2h/dx2  +  J*df/dx  = PRTB EQTN',
     1        '  +  Df*Dh   =   D(TDH)    PP/DH   PP/DF',
     1       /'  ----   ---------   ---------   ---------',
     1        '   ---------   ---------   -----   -----')

      TUDX = 2.0*DELX
      DXSQ = DELX*DELX

      DO I=2,NMAX-1
        DO ICON=-1,1
          PF(ICON) = ZF(I+ICON)
          PH(ICON) = ZH(I+ICON)
        END DO

        IF(IDIF.EQ.0) THEN
          DELF = ( PF(+1) - PF(-1) ) / TUDX     ! CENTRAL DIFFERENCES
          DELH = ( PH(+1) - PH(-1) ) / TUDX
        ELSE
          DELF = ( PF(+1) - PF( 0) ) / DELX     ! FORWARD DIFFERENCES
          DELH = ( PH(+1) - PH( 0) ) / DELX
        END IF
        DSQH = ( PH(-1) - 2.0*PH(0) + PH(+1) ) / DXSQ

        GDLF = GRAD*DELF
        PERR = DSQH + GDLF
        DHDF = DELH*DELF
        TERR = PERR + DHDF
        HRAT = DHDF/DSQH
        FRAT = DHDF/GDLF
        HSSQ = HSSQ + DSQH*DSQH
        FSSQ = FSSQ + GDLF*GDLF
        ESSQ = ESSQ + PERR*PERR
        PSSQ = PSSQ + DHDF*DHDF
        TSSQ = TSSQ + TERR*TERR

C       THIS PARAGRAPH IS TO GET READY FOR THE NEXT "ARBITRARY CHECK NODE"
        IF(LOC(NXT).EQ.I) THEN
          WRITE(50,25) I,DSQH,GDLF,PERR,DHDF,TERR,HRAT,FRAT
25        FORMAT(I6,1PE12.3,4E12.3,0PF8.2,F8.2)
          NXT = NXT + 1
          LOC(NXT) = MAX(LOC(NXT),I+1)
          IF(NXT.EQ.NBCN+1)  NXT = 1
        END IF
      END DO
      WRITE(50,30)
30    FORMAT('  ----   ---------   ---------   ---------',
     1       '   ---------   ---------   -----   -----')

      PNTS = NMAX-2
      RMSE = SQRT(ESSQ/PNTS)
      RMSH = SQRT(HSSQ/PNTS)
      RMSF = SQRT(FSSQ/PNTS)
      RMSP = SQRT(PSSQ/PNTS)
      RMST = SQRT(TSSQ/PNTS)
      PRCT = 100*RMSE/SQRT(RMSH*RMSF)
      WRITE(50,35) RMSH,RMSF,RMSE,RMSP,RMST,PRCT
35    FORMAT('   RMS',1PE12.3,4E12.3,
     1  //9X,'100*RMSE/SQRT(D2H*JDF) =',0PF6.1,' %')

      RETURN
      END
```

```
      SUBROUTINE CHKMAS(ZF,ZH)
C-----------------------------------------------------------------------
C     MASS BALANCE CHECK FOR 1D FLOW MODEL

      REAL      ZF(*), ZH(*)
      INTEGER   LOC(1000)
      COMMON   /PARAMS/ VLNT,GRAD,DELX,FMAX,DELK,NHAR,NMAX,CMAX,APAR
      COMMON   /IFLAGS/ NBCN,IDIF,IHOL

C     STATEMENT FUNCTIONS FOR HARMONIC MEAN T AND HYDRAULIC GRADIENT
      HARM(T0,T1) = 2. / (1./T0 + 1./T1)
      DHDX(P0,P1) = (P1-P0) / DELX

C     PICK OUT ARBITRARY NODES FOR LOCAL MASS BALANCE CHECK
      CALL PKNODS(LOC,NMAX,NBCN)
      NXT = 1
      LOC(1) = 1
COMT  WRITE(50,12)
12    FORMAT(
     1   /'  Node      Q IN       Q OUT       ERROR      PCT ERR ',
     1   /'  ----   ---------   ---------   ---------   ---------')

      F = -11.51                        ! MEAN LN(T) = -11.51
      GKM = EXP(F+VLNT/2.)              ! MEAN TRANSMISSIVITY
      GDLX = GRAD*DELX                  ! GRADIENT * DELTA-X
      FLX = GKM*GRAD*(1-VLNT/2)         ! MEAN FLUX (Gutjahr et. al, [1978])
      T0 = EXP(F+ZF(2))                 ! THIS PARAGRAPH IS FOR
      T1 = EXP(F+ZF(1))                 ! CALCULATING THE FIRST
      P0 = GDLX + ZH(2)                 ! FLUX (IN AT THE LEFT)
      P1 = 0.00 + ZH(1)
      QL = HARM(T0,T1) * DHDX(P1,P0)
      SMFLX = 0.0                       ! SUM OF FLUX ERRORS
      SMPCT = 0.0                       ! SUM OF PCT  ERRORS
      SMPSQ = 0.0
      SMESQ = 0.0

      DO I=2,NMAX-1
        T1 = EXP(F+ZF(I+1))
        P1 = I*GDLX + ZH(I+1)
        QR = HARM(T0,T1) * DHDX(P0,P1)
        ER = QR - QL
        PE = 100.*ER/FLX
        SMFLX = SMFLX + ER              ! SUM UP THE FLUX ERRORS
        SMPCT = SMPCT + PE              ! AND THE PERCENT ERRORS
        SMESQ = SMESQ + ER*ER           ! SUM UP THE ERRORS**2
        SMPSQ = SMPSQ + PE*PE           ! SUM UP PCT ERRORS**2
        CDIST = FLOAT(I)*DELX/APAR      ! CORRELATION DISTANCE

C       WRITE STUFF OUT FOR LATER PLOTTING AND DETERMINISTIC SOLUTION
        WRITE(95) I,CDIST,ZF(I),P0,PE,T0,ZH(I)
COMT    CALL PROGSS(I,NMAX,'MASS BALANCE CALCULATIONS')

C       THIS PARAGRAPH IS TO GET READY FOR THE NEXT "ARBITRARY CHECK NODE"
        IF(LOC(NXT).EQ.I) THEN
          WRITE(50,25) I,QL,QR,ER,PE
25        FORMAT(I6,1PE12.3,2E12.3,0PF9.2,' %')
          NXT = NXT + 1
          LOC(NXT) = MAX(LOC(NXT),I+1)
          IF(NXT.EQ.NBCN+1)  NXT = 1
        END IF

        T0 = T1
        P0 = P1
        QL = QR
      END DO

      PTS = NMAX-2
      E_RMSE = SQRT(SMESQ/PTS)
      P_RMSE = SQRT(SMPSQ/PTS)
COMT  WRITE(50,30) SMFLX,SMPCT,E_RMSE,P_RMSE
30    FORMAT('  ----   ---------   ---------   ---------',
     1  /'                      SUMS -->',1PE12.3,0PF9.2,' %',
     1  /'                      RMSE -->',1PE12.3,0PF9.2,' %')

      RETURN
      END
```

```
      SUBROUTINE COVFHX(ZF,ZH,SMF,SMH,RF,RH, SMX,RX,CPL, NPTS,MLAG)
C-----------------------------------------------------------------------
C     CALCULATE ONE DIMENSIONAL AUTO AND CROSS COVARIANCES
C
C     ZF  AND  ZH = THE TWO 1D INPUT FIELDS
C     SMF,SMH,SMX = SUMMATION ARRAYS FOR EXPECTED VALUES
C     CPL  ARRAY  = TO ACCUMULATE THE NUMBER OF COUPLES
C     NPTS = LENGTH OF THE ZF AND ZH ARRAYS
C     MLAG = MAXIMUM LAG
C     RF, RH, RX  = AUTO AND CROSS COVARIANCE ARRAYS
C
C     CALL COVFHX REPEATEDLY AFTER EACH SIMULATION, THEN CALL RRRFHX TO
C     RETURN THE COVARIANCES CALCULATED OVER THE ENSEMBLE OF RANDOM FIELDS
C-----------------------------------------------------------------------
C >>> NOTE - THIS ASSUMES BOTH FIELDS ARE MEAN ZERO FIELDS <<<
C-----------------------------------------------------------------------
      REAL     ZF(*),ZH(*),RF(0:*),RH(0:*),RX(0:*)
      REAL     SMF(0:*),SMH(0:*),SMX(0:*),CPL(0:*)


      DO J=0,MLAG
        CPLS = NPTS-J
        SUMF = 0.0
        SUMH = 0.0
        SUMX = 0.0

C       REPORT COMPUTATION PROGRESS
        CALL PROGSS(J,MLAG,'COVARIANCE CALCULATIONS')

        DO I=1,CPLS
          SUMF = SUMF + ZF(I)*ZF(I+J)
          SUMH = SUMH + ZH(I)*ZH(I+J)
          SUMX = SUMX + ZF(I)*ZH(I+J)
        END DO

        SMF(J) = SMF(J) + SUMF
        SMH(J) = SMH(J) + SUMH
        SMX(J) = SMX(J) + SUMX
        CPL(J) = CPL(J) + CPLS
      END DO

      RETURN


      ENTRY RRRFHX(SMF,SMH,SMX,RF,RH,RX,CPL,MLAG)
C-----------------------------------------------------
C     FINAL COVARIANCE CALULATIONS (OVER THE ENSEMBLE)

      DO J=0,MLAG
        RF(J) = SMF(J) / CPL(J)
        RH(J) = SMH(J) / CPL(J)
        RX(J) = SMX(J) / CPL(J)
      END DO

      RETURN
      END
```

```
      SUBROUTINE DFFT(X,N,IFB)
C-----------------------------------------------------------------------
C     ONE DIMENSIONAL FAST FOURIER TRANSFORM ROUTINE  (FORWARD AND INVERSE)
C-----------------------------------------------------------------------
C     THIS ROUTINE, MODIFIED BY D. A. (TONY) ZIMMERMAN AT NEW MEXICO TECH
C     IN 1987 AND VERIFIED WITH IMSL ROUTINES, WAS TAKEN FROM PAGE 108 OF:
C
C               RAFAEL .C GONZALEZ AND PAUL WINTZ,  1987.
C                    ''DIGITAL IMAGE PROCESSING''
C                  ADDISON-WESLEY PUBLISHING COMPANY
C
C     X = COMPLEX SEQUENCE TO BE TRANSFORMED (INPUT)
C     X = COMPLEX TRANSFORMED ARRAY ON OUTPUT
C     N = NUMBER POINTS IN F TO BE TRANSFORMED
C     IFB = -1 FOR FORWARD TRANSFORM   ( EXP(-i*2PIux/N) )
C     IFB = +1 FOR INVERSE TRANSFORM   ( EXP(+i*2PIux/N) )
C-----------------------------------------------------------------------
C *** NOTE: THE INPUT SEQUENCE MUST BE OF LENGTH N = 2**M  FOR SOME M > 0  ***
C-----------------------------------------------------------------------
      PARAMETER   (PI=3.14159265400)
      REAL*8       A
      COMPLEX*16   X(*),U,W,T

      N2 = N/2
      LN = 1
      HMCS = N
199   HMCS = HMCS/2.
      LN = LN + 1
      IF(HMCS.GT.2.) GO TO 199

      IF(HMCS.NE.2.) THEN
        WRITE(*,*)'***** SUBROUTINE FFT, ARRAY TO BE TRANSFORMED'
        WRITE(*,*)'      MUST BE OF LENGTH = 2**N FOR SOME N'
        STOP
      END IF

      IF(IFB.GT.0) THEN
        DO 10 K=1,N
10      X(K) = DCONJG( X(K) )
      END IF

      J = 1
      DO 3 I=1,N-1
         IF(I.GE.J) GO TO 1
         T = X(J)
         X(J) = X(I)
         X(I) = T
1        K = N2
2        IF(K.GE.J) GO TO 3
         J = J-K
         K = K/2
         GO TO 2
3        J = J+K

      DO 5 L=1,LN
         LE  = 2**L
         LE1 = LE/2
         U = DCMPLX(1.D0,0.D0)
         A = PI/DFLOAT(LE1)
         W = DCMPLX(DCOS(A),-DSIN(A))
         DO 5 J=1,LE1
            DO 4 I=J,N,LE
               IP = I+LE1
               T  = X(IP)*U
               X(IP) = X(I)-T
4              X(I)  = X(I)+T
5           U = U*W

      IF(IFB.GT.0) THEN
        DO 20 K=1,N
20      X(K) = DCONJG( X(K) )
      END IF

      RETURN
      END
```

```
      SUBROUTINE PKNODS(LOC,NMAX,NBCN)
C-----------------------------------------------------------------------
C     PICK OUT NBCN ARBITRARY NODE LOCATIONS FOR LOOKING AT LOCAL MASS BALANCE

      INTEGER       LOC(*)

      DO I=1,NBCN
        LOC(I) = 1 + URN01()*(NMAX-2)
        IF(LOC(I).EQ.1) LOC(I) = 2
      END DO

C     SORT THEM INTO INCREASING ORDER
      DO I=1,NBCN-1
        DO J=I+1,NBCN
          IF(LOC(J) .LT. LOC(I)) THEN
            LOCTMP = LOC(J)
            LOC(J) = LOC(I)
            LOC(I) = LOCTMP
          END IF
        END DO
      END DO

      RETURN
      END




      SUBROUTINE PNTCOV(RF,RH,RX)
C-----------------------------------------------------------------------
C     COMPUTE THEORETICAL COVARIANCES AND PRINT ALONG WITH DISCRETE ONES

      REAL     RF(0:*),RH(0:*),RX(0:*)
      COMMON  /PARAMS/ VLNT,GRAD,DELX,FMAX,DELK,NHAR,NMAX,CMAX,APAR

      WRITE(50,30)
30    FORMAT(/'  LAG      THEORY-F  SAMPLE-F     THEORY-H ',
     1                     '   SAMPLE-H    THEORY-XC  SAMPLE-XC',
     1       /'  ----    --------  --------    ----------',
     1                     ' ----------  ---------  ---------')

      GRSQ = GRAD*GRAD                        ! HEAD GRADIENT SQUARED
      ASQD = APAR*APAR                        ! A PARAMETER SQUARED
      MX   = 59                               ! MX+1 OUTPUT POINTS
      A1   = ALOG10(CMAX/DELX)                ! OTHER STUFF FOR COVARIANCE OUTPUT
      A2   = ALOG10(FLOAT(MX))
      PWR  = A1/A2

      DO L=0,MX
        M = FLOAT(L)**PWR
        X = M * DELX
        ARG = X*X / (2.0*ASQD)
        EMA = EXP(-ARG)
        TRF = VLNT*(1.-2.0*ARG)*EMA           ! THEORETICAL F COVARIANCE
        TRH = VLNT*GRSQ*ASQD*EMA              ! THEORETICAL H COVARIANCE
        TRX = VLNT*GRAD*X*EMA                 ! THEORETICAL X-COVARIANCE
        RXM = -RX(M)                          ! SIGN CHANGE ON CROSS-COVARIANCE
        WRITE(50,40) X, TRF,RF(M), TRH,RH(M), TRX,RXM
40      FORMAT(F6.2, 2X, 2F10.4, 1X, 2(F12.4), 3X, 2F10.4)
      END DO

      RETURN
      END
```

```
      PROGRAM COSM2D_DP
C-----------------------------------------------------------------------
C     COSIMULATION OF TWO-DIMENSIONAL RANDOM FIELDS   ( LN(T) AND HEAD )
C-----------------------------------------------------------------------
C     SINCE THIS IS A RESEARCH CODE, IT IS WRITTEN USING SOME ENHANCED
C     FORTRAN FEATURES (VAX-VMS FORTRAN VERSION 4.1) WHICH ARE NOT PART
C     OF THE CURRENT ANSI STANDARD FORTRAN-77 LIBRARY.
C     PROGRAMMED TO USE THE TELIS COVARIANCE FUNCTION FOR LN(T) PROCESS.
C     URNIT = INITIALIZE RANDOM NUMBER GENERATOR
C     URNV2 = GENERATE MEAN-ZERO, VARIANCE 1/2 UNIFORM RANDOM NUMBERS
C     FFT = ONE-DIMENSIONAL FAST FOURIER TRANSFORM ROUTINE
C     BARVAR = CALCULATE MEAN AND VARIANCE OF AN ARRAY
C-----------------------------------------------------------------------
      PARAMETER    (PI=3.141592654, MX1=64,MX2=128, LF=50,LH=51)
      CHARACTER     FILE*40,SEED*12
      COMPLEX       DZFP, DZHP, CARG
      COMPLEX*16    TEMPF(2*MX2), TEMPH(2*MX2), DARG
      COMPLEX*16    DZF(2*MX1,2*MX2), DZH(2*MX1,2*MX2)
      REAL          F(MX1,MX2), H(MX1,MX2)
      COMMON       /STUFF/ M1,M2,GRAD,DELK,BSQD,SQRC,VLNT,VHDS


      WRITE(*,*)'ENTER THE LN(T) VARIANCE'
      READ (*,*) VLNT
      WRITE(*,*)'ENTER THE MAXIMUM FREQUENCY'
      READ (*,*) FMAX
      WRITE(*,*)'ENTER NHAR-X AND NHAR-Y'
      READ (*,*) M1,M2
      WRITE(*,*)'ENTER RANDOM NBR GEN SEED'
      READ (*,*) ISEED

C     REQUIRED FMAX=10 FOR Ln(T) PROCESS (CAPTURES 99% OF TELIS SPECTRUM)
C     BUT FMAX=64 YIELDS 10 POINTS/CLEN WHEN CLEN=1
C     AND FMAX=44 YIELDS  7 POINTS/CLEN WHEN CLEN=1
      DUMMY = URNIT(ISEED)          ! INITIALIZE RANDOM NUMBER GENERATOR
      SQRT2 = SQRT(2.0)             ! NAMED CONSTANT
      POV2 = PI/2.0                 ! NAMED CONSTANT
      GRAD = 1.0                    ! MEAN HYDRAULIC GRADIENT
      CLEN = 1.0                    ! CORRELATION LENGTH
C     FMAX = 44.                    ! MAXIMUM FREQUENCY
      DELK = FMAX/MIN(M1,M2)        ! FREQUENCY SPACING
      DELX = 2.0*PI/FMAX            ! SPACE DISCRETIZATION
      PPCL = CLEN/DELX              ! NBR POINTS/CORR LENGTH
      BSQD = 1.0/(CLEN*CLEN)        ! NEEDED FOR SPECTRAL DENSITY FUNCTION
      CNST = 2.0*VLNT/(CLEN*PI*PI)  ! CONSTANT TERM IN SPECTRAL DENSITY FCN
      SQRC = SQRT(CNST)             ! NEEDED IN SQDF FORMULA
      VHDS = VLNT*GRAD**2/(2*BSQD)  ! THEORETICAL HEAD VARIANCE

C     REFLECT INPUT DATA AND INTERNAL PARAMETERS TO SCREEN AND LIST FILE
      NC = ALOG10(FLOAT(ISEED)) + 1
      WRITE(SEED,2) ISEED
2     FORMAT(I<NC>)
      FILE = 'T_' // SEED(1:NC) // '.CSM_LST'
      OPEN(UNIT=1,NAME=FILE,STATUS='NEW')
      WRITE(1,5) ISEED,GRAD,CLEN,DELX,PPCL,VLNT,FMAX,DELK,M1,M2
      WRITE(6,5) ISEED,GRAD,CLEN,DELX,PPCL,VLNT,FMAX,DELK,M1,M2
5     FORMAT(/' TELIS.FOR INPUT:'/,
     1 /' SEED =',I9  /' GRAD =',F8.1,/' CLEN =',F8.1,
     1 /' DELX =',F12.5/' NPPC =',F8.1,
     1 /' VLNT =',F9.2/' FMAX =',F9.2 /' DELK =',F12.5,
     1 /'  M1  =',I6/'  M2  =',I6)
```

```
      WRITE(*,*)
      DO K2=1,M2
        CALL LPLST(K2,M2,'BUILDING DZF AND DZH MATICES')
        DO K1=1,M1
          CALL DZPCSS(K1,K2,DZFP,DZHP)
C         LOAD QUADRANT I
          DZF(K1,K2) = DCMPLX( DBLE(DZFP), 0.D0 )
          DZH(K1,K2) = DCMPLX( DBLE(DZHP), 0.D0 )

C         LOAD QUADRANT II
          K1PM1 = K1+M1
          K1NEG =-M1+K1-1
          CALL DZPCSS(K1NEG,K2,DZFP,DZHP)
          DZF(K1PM1,K2) = DCMPLX( DBLE(DZFP), 0.D0 )
          DZH(K1PM1,K2) = DCMPLX( DBLE(DZHP), 0.D0 )

C         LOAD QUADRANTS III AND IV
          DZF(  K1, K2+M2) = DCMPLX( 0.D0, 0.D0 )
          DZH(K1PM1,K2+M2) = DCMPLX( 0.D0, 0.D0 )
        END DO
      END DO

      WRITE(*,*)
C     FIRST DO THE ROW TRANSFORMS ...
      DO K2=1,M2
        CALL LPLST(K2,2*M2,'DOING THE ROW FFT TRANSFORMS')
        CALL DFFT(DZF(1,K2),2*M1,-1)
        CALL DFFT(DZH(1,K2),2*M1,-1)
      END DO

      WRITE(*,*)
C     NOW DO THE COLUMN TRANSFORMS ...
      DO K1=1,M1
        CALL LPLST(K1,M1,'DOING THE COL FFT TRANSFORMS')
        DO K2=1,2*M2
          TEMPF(K2) = DZF(K1,K2)
          TEMPH(K2) = DZH(K1,K2)
        END DO
        CALL DFFT(TEMPF,2*M2,-1)
        CALL DFFT(TEMPH,2*M2,-1)
        DO K2=1,2*M2
          DZF(K1,K2) = TEMPF(K2)
          DZH(K1,K2) = TEMPH(K2)
        END DO
      END DO

      WRITE(*,*)
      DO K2=1,M2
        CALL LPLST(K2,M2,'TAKING 2 TIMES THE REAL  ETC')
        DO K1=1,M1
          CARG = DCMPLX( 0.0, -POV2*(FLOAT(K1)/M1+FLOAT(K2)/M2) )
          DARG = DBLE(CARG)
          DZF(K1,K2) = DZF(K1,K2)*2.D0*CDEXP(DARG)
          DZH(K1,K2) = DZH(K1,K2)*2.D0*CDEXP(DARG)
          F(K1,K2) = REAL( DZF(K1,K2) )
          H(K1,K2) = REAL( DZH(K1,K2) )
        END DO
      END DO
```

```
C       NOW WRITE THE FIELDS TO OUTPUT FILES AND CALCULATE STATS ...
        FILE = FILE(1:NC+2) // '.LNT_1'
        OPEN(UNIT=LF,NAME=FILE,STATUS='NEW',FORM='UNFORMATTED')
        FILE = FILE(1:NC+2) // '.PHD_1'
        OPEN(UNIT=LH,NAME=FILE,STATUS='NEW',FORM='UNFORMATTED')
        WRITE(LF) ((F(K1,K2),K1=1,M1),K2=1,M2)
        WRITE(LH) ((H(K1,K2),K1=1,M1),K2=1,M2)
        CALL STATS(LF,LH,F,H,M1,M2)

        CLOSE(UNIT=LF)
        CLOSE(UNIT=LH)

        FILE = FILE(1:NC+2) // '.LNT_2'
        OPEN(UNIT=LF,NAME=FILE,STATUS='NEW',FORM='UNFORMATTED')
        FILE = FILE(1:NC+2) // '.PHD_2'
        OPEN(UNIT=LH,NAME=FILE,STATUS='NEW',FORM='UNFORMATTED')
        WRITE(*,*)
        DO K2=1,M2
          CALL LPLST(K2,M2,'2ND FIELD VIA IMAGINARY PART')
          DO K1=1,M1
            F(K1,K2) = DIMAG( DZF(K1,K2) )
            H(K1,K2) = DIMAG( DZH(K1,K2) )
          END DO
        END DO
        WRITE(LF) ((F(K1,K2),K1=1,M1),K2=1,M2)
        WRITE(LH) ((H(K1,K2),K1=1,M1),K2=1,M2)
        CALL STATS(LF,LH,F,H,M1,M2)

        CALL EXIT
        END
```

```
      SUBROUTINE DZPCSS(K1,K2,DZFP,DZHP)
C-----------------------------------------------------------------------
C     GENERATE THE 2D DZF AND DZH PROCESSES

      COMPLEX  DZFP,DZHP,TFCN
      COMMON  /STUFF/ M1,M2,GRAD,DELK,BSQD,SQRC,VLNT,VHDS


      OMGA1 = K1*DELK
      OM1SQ = OMGA1 * OMGA1

      OMGA2 = K2*DELK
      OM2SQ = OMGA2*OMGA2

      OMSUM = OM1SQ + OM2SQ
      OM4TH = OMSUM**(0.25)
      SQDF = SQRC * OM4TH/(BSQD+OMSUM) * DELK

      UA = URNV2()
      UB = URNV2()

      DZFP = CMPLX( SQDF*UA, SQDF*UB )
      TFCN = CMPLX( 0.0, -GRAD*OMGA1/OMSUM )
      DZHP = TFCN*DZFP

      RETURN
      END



      SUBROUTINE STATS(LF,LH,F,H,N1,N2)
C-----------------------------------------------------------------------
C     CALCULATE STATS OF OUTPUT FIELDS

      REAL      F(*), H(*)
      COMMON   /STUFF/ M1,M2,GRAD,DELK,BSQD,SQRC,VLNT,VHDS

      REWIND LF
      READ(LF) (F(I),I=1,N1*N2)
      CALL BARVAR(F,N1*N2,FBAR,FVAR)

      REWIND LH
      READ(LH) (H(I),I=1,N1*N2)
      CALL BARVAR(H,N1*N2,HBAR,HVAR)

      WRITE(6,10) FBAR,FVAR,VLNT,HBAR,HVAR,VHDS
      WRITE(1,10) FBAR,FVAR,VLNT,HBAR,HVAR,VHDS
10    FORMAT(/' OUTPUT STATISTICS:',
     1     //' FBAR =',F10.4/' FVAR =',F8.2,/' VLNT =',F8.2,
     1     //' HBAR =',F10.4/' HVAR =',F8.2,/' VHDS =',F8.2)

      RETURN
      END



      SUBROUTINE LPLST(K,KMAX,MSG)
C-----------------------------------------------------------------------
C    "LOOP-LIST"   REPORT DO-LOOP COMPUTATION PROGRESS

      CHARACTER  MSG*(*)


      ICHK = MAX(KMAX/20,1)

      IF(MOD(K,ICHK).EQ.0) THEN
        IPCT = 100 * FLOAT(K)/FLOAT(KMAX)
        WRITE(6,10) MSG,IPCT
10      FORMAT(1H$,A,' ... (',I3,' % ) ... ')
      END IF

      RETURN
      END
```

```
      PROGRAM RESIDUALS
C-----------------------------------------------------------------------
C     CALCULATE RESIDUALS AND GENERATE RESIDUAL MAP

      PARAMETER    (MXX=512*1024)
      REAL          T(MXX),H(MXX),E(MXX)
      CHARACTER*40  TFILE,PFILE,OFILE
      COMMON       /PROMPT/ NX,NY


      CALL SPRMPT('ENTER  INPUT .TRN FILE > ',' ',TFILE,NC,1,IER)
      CALL SPRMPT('ENTER  INPUT .PHI FILE > ',' ',PFILE,NC,0,IER)
      CALL IPRMPT('ENTER NODES-X, NODES-Y > ',NX,2,N,0,IER)
      CALL RPRMPT('ENTER THE DELTA-X VALU > ',DELX,1,N,0,IER)

C     WRITE(*,*)'(1) - Standard residual calculation'
C     WRITE(*,*)'(2) - Scale residual by conductivity'
C     READ (*,*) ISCAL
      ISCAL = 1

      OPEN(UNIT=1, NAME=TFILE,STATUS='OLD',FORM='UNFORMATTED')
      OPEN(UNIT=2, NAME=PFILE,STATUS='OLD',FORM='UNFORMATTED')

      READ(1) (T(I),I=1,NX*NY)
      READ(2) (H(I),I=1,NX*NY)

      CALL BARVAR(T,NX*NY,TBAR,TVAR)
      CALL BARVAR(H,NX*NY,HBAR,HVAR)

      WRITE(*,10) TBAR,TVAR,HBAR,HVAR
10    FORMAT(/' THE TRN MEAN =',1PE12.5,
     1       /' TRN VARIANCE =',  E12.5,
     1      //' THE PHI MEAN =',  E12.5,
     1       /' PHI VARIANCE =',  E12.5/)

      CALL CHKMAS(T,H,E,NX,NY,DELX, ISCAL)
      WRITE(99) (E(I),I=1,NX*NY)
      CALL BARVAR(E,NX*NY,BAR,VAR)

      SME = BAR*NX*NY
      WRITE(*,20) SME,BAR,VAR
20    FORMAT(/' SUM RESIDUAL =',1PE12.5,
     1       /' THE ERR MEAN =',  E12.5,
     1       /' ERR VARIANCE =',  E12.5)

      STOP 'ERROR MAP IN FOR099.DAT'
      END
```

```
      SUBROUTINE CHKMAS(T,H,E,NX,NY,DELX, ISCAL)
C-------------------------------------------------------------------------------
C     MAS BAL CHECK ALGORITHM

      REAL   T(NX,NY),H(NX,NY),E(NX,NY), PT(-1:1),PH(-1:1)

C     STATEMENT FUNCTION FOR HARMONIC MEAN T
      HARMT(T0,T1) = 2.0/(1./T0 + 1./T1)

      DO J=2,NY-1
        CALL LPLST(J,NY,'MASS BALANCE CHECK')
        DO I=2,NX-1
          SUMM = 0.0
          DO IAXS=1,2
C           FILL THE ARRAYS FOR THIS DIRECTION
            DO ICON = -1,+1
              CALL INCRMT(I,J,IAXS,ICON,II,JJ)
              PT(ICON) = T(II,JJ)
              PH(ICON) = H(II,JJ)
            END DO
C           NOW CALCULATE THE FLUXES
            DO ICON = -1,+1,2
              DELH = ( PH(ICON) - PH(0) ) / DELX
              HARM = HARMT( PT(ICON),PT(0) )
              SUMM = SUMM + HARM*DELH
            END DO
          END DO
C         SCALING IS ACTUALLY 100*[SUMM*(Tmean/T(i,j))]/(Tmean*Grad)
C         WHICH IS PERCENT OF MEAN FLOW (Grad=1)
          IF(ISCAL.EQ.2) SUMM = 100.*SUMM/T(I,J)
          E(I,J) = SUMM
        END DO
      END DO

      RETURN
      END



      SUBROUTINE INCRMT(I,J,IAXS,IADD,II,JJ)
C-------------------------------------------------------------------------------
C     INCREMENT INDEX ALONG IAXS BY AMOUNT IADD

      II = I
      JJ = J
      IF(IAXS.EQ.1) II = II + IADD
      IF(IAXS.EQ.2) JJ = JJ + IADD

      RETURN
      END
```

```
      PROGRAM MAK_FBC
C-----------------------------------------------------------------------
C     MAKE FDM MODEL BOUNDARY CONDITIONS AND CALCULATE BOUNDARY FLUXES

      PARAMETER    (MXX=512*1024)
      REAL          T(MXX),H(MXX),B(MXX)
      CHARACTER*40  REC(36),TFILE,PFILE,OFILE
      COMMON       /INDATA/ REC
      COMMON       /PROMPT/ NX,NY,GRAD,DELX


      CALL SPRMPT('ENTER  INPUT .TRN FILE > ',' ',TFILE,NC,1,IER)
      CALL SPRMPT('ENTER  INPUT .PHI FILE > ',' ',PFILE,NC,0,IER)
      CALL IPRMPT('ENTER NODES-X, NODES-Y > ',NX,2,N,0,IER)
      CALL RPRMPT('ENTER  GRAD  AND  DELX > ',GRAD,2,N,0,IER)
      CALL SPRMPT('ENTER  OUTPUT FILENAME > ',' ',OFILE,NC,0,IER)

      OPEN(UNIT=1, NAME=TFILE,STATUS='OLD',FORM='UNFORMATTED')
      OPEN(UNIT=2, NAME=PFILE,STATUS='OLD',FORM='UNFORMATTED')
      OPEN(UNIT=50,NAME=OFILE,STATUS='NEW',CARRIAGECONTROL='LIST')

      CALL READU(1,T,NX,NY)
      CALL READU(2,H,NX,NY)

      CALL BND_SUB(T,H,B,NX,NY)


COMT  WRITE(*,20)
20    FORMAT(/' TRN IN FOR033.DAT, HEADS IN FOR045.DAT',
     1       /' BOTH DIMENSIONED AT NX-2 BY NY-2')

      END
```

```
      SUBROUTINE BND_SUB(T,H,B,NX,NY)
C-----------------------------------------------------------------------
C     CALCULATE AND WRITE OUT BOUNDARY CONDITIONS

      REAL       T(NX,NY),H(NX,NY),B(NX,NY), LFTFLX
      DATA .     IFACE,K,IVM /0,1,1/
      COMMON    /PROMPT/ MX,MY,GRAD,DELX


C     NOW CALCULATE AND WRITE OUT THE BOUNDARY CONDITIONS
C***      WRITE(50,*)'BOUNDARY FLUXES'
C***      NBFLX = 2*(NX-2) + 2*(NY-4)
C***      WRITE(50,5) NBFLX
5     FORMAT(5I5,1PE12.3)
6     FORMAT(4I5,1PE12.3)

      NBFLX = 2*(NY-2-2)
COMT      WRITE(50,7) NBFLX
7     FORMAT(I6,2X,'; number of prescribed fluxes')
8     FORMAT(3I5/1PE12.5)

C     LOWER BOUNDARY
      J = 2
      DO 10 I=2,NX-1
      BFLX = FLUX(T,H,NX,NY,I,J,K)
      B(I,J) = BFLX
      BOTFLX = BOTFLX + BFLX
C***10    WRITE(50,5) I-1,J-1,K,IFACE,IVM,BFLX
10    CONTINUE

C     UPPER BOUNDARY
      J = NY-1
      DO 20 I=2,NX-1
      BFLX = FLUX(T,H,NX,NY,I,J,K)
      B(I,J) = BFLX
      TOPFLX = TOPFLX + BFLX
C***20    WRITE(50,5) I-1,J-1,K,IFACE,IVM,BFLX
20    CONTINUE

C     LEFT BOUNDARY
      I = 2
      DO 30 J=3,NY-2
      BFLX = FLUX(T,H,NX,NY,I,J,K)
      B(I,J) = BFLX
      LFTFLX = LFTFLX + BFLX
C***30    WRITE(50,5) I-1,J-1,K,IFACE,IVM,BFLX
COMT      WRITE(50,8) I-1,J-1,K,BFLX
30    CONTINUE

C     RIGHT BOUNDARY
      I = NX-1
      DO 40 J=3,NY-2
      BFLX = FLUX(T,H,NX,NY,I,J,K)
      B(I,J) = BFLX
      RHTFLX = RHTFLX + BFLX
C***40    WRITE(50,5) I-1,J-1,K,IFACE,IVM,BFLX
COMT      WRITE(50,8) I-1,J-1,K,BFLX
40    CONTINUE
```

```
C-------------------------------------------------------
C     THIS PARAGRAPH LISTS OUT THE BOUNDARY FLUX IMBALANCE
      SUMFLX = TOPFLX + BOTFLX + RHTFLX + LFTFLX
      WRITE(*,45) TOPFLX,BOTFLX,RHTFLX,LFTFLX,SUMFLX
45    FORMAT(/' TOPFLX = ',1PE12.5,
     1       /' BOTFLX = ',  E12.5,
     1       /' RHTFLX = ',  E12.5,
     1       /' LFTFLX = ',  E12.5,
     1       /' SUMFLX = ',  E12.5/)

C     OPEN(UNIT=99,NAME='EDG_FLX.MAP',STATUS='NEW',FORM='UNFORMATTED')
C     WRITE(99) ((B(I,J),I=2,NX-1),J=2,NY-1)


C     CALCULATE AND WRITE OUT THE DIRICHLET BOUNDARY CONDITIONS

C     WRITE(50,18) 2*(NY-2)
      WRITE(50,19) 2*NX + 2*(NY-2)
18    FORMAT(I6)
19    FORMAT(I6,2X,'; number of prescribed heads')

C NOT       HMAX = NX*DELX*GRAD / 2.0
C USD       HMIN = -HMAX

C     WRITE(*,*)' TOP DIRICHLET BOUNDARY CONDITIONS'
      DO 100 I=1,NX
100   WRITE(50,8) I,NY,1,H(I,NY)

C     WRITE(*,*)' BOT DIRICHLET BOUNDARY CONDITIONS'
      DO 101 I=1,NX
101   WRITE(50,8) I, 1, 1, H(I,1)

      WRITE(*,*)' RGHT BOUNDARY CONDITIONS ...'
      DO 200 J=2,NY-1
        BCRHT = H(NX,J)
C***    WRITE(50,50) NX-2,J-1,1, 2, 1,BCRHT, 0.0
        WRITE(50,51) NX,J,1,BCRHT
200    CONTINUE

      WRITE(*,*)' LEFT BOUNDARY CONDITIONS ...'
      DO 300 J=2,NY-1
         BCLFT = H(1,J)
C***     WRITE(50,50)  1,J-1,1, 1, 1,BCLFT, 0.0
         WRITE(50,51)  1,J,1,BCLFT
300    CONTINUE

50    FORMAT(3I4,2X,2(2X,I2),4X,F6.2,F4.0)
51    FORMAT(3I5/1PE12.5)

C     WRITE OUT THE TRANSMISSIVITY FILE
COMT      WRITE(33) ((T(I,J),I=2,NX-1),J=2,NY-1)
COMT      WRITE(45) ((H(I,J),I=2,NX-1),J=2,NY-1)

      RETURN
      END



      SUBROUTINE INCRMT(I,J,K,IAXS,IADD,II,JJ,KK)
C-------------------------------------------------------------------------------
C     INCREMENT INDEX ALONG IAXS BY AMOUNT IADD

      II = I
      JJ = J
      KK = K

      IF(IAXS.EQ.1) II = II + IADD
      IF(IAXS.EQ.2) JJ = JJ + IADD
      IF(IAXS.EQ.3) KK = KK + IADD

      RETURN
      END
```

```
      FUNCTION FLUX(T,P,NX,NY,I,J,K)
C-----------------------------------------------------------------------
C     CALCULATE BOUNDARY FLUX

      REAL       T(NX,NY),P(NX,NY)
      COMMON    /PROMPT/ MX,MY,GRAD,DELX

C     STATEMENT FUNCTION FOR HARMONIC MEAN T
      HARMT(T0,T1) = 2.0/(1.0/T0 + 1.0/T1)


C     SPECIAL CORNER FLAG
      ICORN = 0

C     LOWER BOUNDARY
      IF(J.EQ. 2 .) THEN
        ICON1 = -1
        ICON2 = -1
        IAXS1 =  2
        IAXS2 =  2
        IF(I.EQ. 2) IAXS1 = 1
        IF(I.EQ.NX-1) ICORN = 1     ! LOWER RIGHT CORNER
      END IF

C     UPPER BOUNDARY
      IF(J.EQ.NY-1) THEN
        ICON1 = 1
        ICON2 = 1
        IAXS1 = 2
        IAXS2 = 2
        IF(I.EQ.NX-1) IAXS1 = 1
        IF(I.EQ.2) ICORN = 2        ! UPPER LEFT CORNER
      END IF

C     LEFT AND RIGHT BOUNDARIES
      IF(J.GT.2 .AND. J.LT.NY-1) THEN
        IAXS1 =  1
        IAXS2 =  1
        IF(I.EQ.2) THEN
          ICON1 = -1
          ICON2 = -1
        ELSE IF(I.EQ.NX-1) THEN
          ICON1 =  1
          ICON2 =  1
        END IF
      END IF

      T0 = T(I,J)
      P0 = P(I,J)
      FLUX = 0.0

      DO IAXS=IAXS1,IAXS2
        DO ICON=ICON1,ICON2,2
          CALL INCRMT(I,J,K,IAXS,ICON,II,JJ,KK)
          T1 = T(II,JJ)
          P1 = P(II,JJ)
          HMT = HARMT(T0,T1)
          DLP =(P1-P0)/DELX
          FLUX = FLUX + HMT*DLP
        END DO
      END DO

C     ADD FLUXES FROM THESE NASTY CORNERS SEPARATELY ...
      IF(ICORN.EQ.1) CALL INCRMT(I,J,K,1, 1,II,JJ,KK)
      IF(ICORN.EQ.2) CALL INCRMT(I,J,K,1,-1,II,JJ,KK)
      IF(ICORN.GT.0) THEN
        T1 = T(II,JJ)
        P1 = P(II,JJ)
        HMT = HARMT(T0,T1)
        DLP =(P1-P0)/DELX
        FLUX = FLUX + HMT*DLP
      END IF

      RETURN
      END
```

```
      PROGRAM PCT_DET
C-----------------------------------------------------------------------
C     CALCULATED MASS BALANCE ERRORS FOR COGENERATED FIELDS (IN 2D)
C     USING DETERMINISTIC SOLUTION FOR REFERENCE

      PARAMETER     (MXX=512*1024)
      CHARACTER*40   TFILE,CFILE,DFILE,EFILE
      REAL           TRN(MXX),COG(MXX),DET(MXX),PCT(MXX)
      COMMON         NX,NY,DELX


      CALL SPRMPT('ENTER THE TRN FILENAME > ',' ',TFILE,NC,1,IER)
      CALL SPRMPT('ENTER THE COG FILENAME > ',' ',CFILE,NC,0,IER)
      CALL SPRMPT('ENTER THE DET FILENAME > ',' ',DFILE,NC,0,IER)
      CALL SPRMPT('ENTER THE PCT FILENAME > ',' ',EFILE,NC,0,IER)
      CALL IPRMPT('ENTER NMBR NODES NX,NY > ',NX,2,N,0,IER)
      CALL RPRMPT('ENTER THE DELTAX VALUE > ',DELX,1,N,0,IER)

      OPEN(UNIT=1,NAME=TFILE,STATUS='OLD',FORM='UNFORMATTED')
      CALL READU(1,TRN,NX,NY)
      CLOSE(UNIT=1)

      OPEN(UNIT=1,NAME=CFILE,STATUS='OLD',FORM='UNFORMATTED')
      CALL READU(1,COG,NX,NY)
      CLOSE(UNIT=1)

      OPEN(UNIT=1,NAME=DFILE,STATUS='OLD',FORM='UNFORMATTED')
      CALL READU(1,DET,NX,NY)
      CLOSE(UNIT=1)

      OPEN(UNIT=9,NAME=EFILE,STATUS='NEW',FORM='UNFORMATTED')

      DO 10 I=1,NX*NY
10    PCT(I) = 0.0
      CALL MASCHK(TRN,COG,DET,PCT,NX,NY)
      CALL WRITU(9,PCT,NX,NY)

      CALL STATS(PCT,NX*NY,NEX,BAR,SDV,PMN,PMX)
      WRITE(*,20) NEX,BAR,SDV,PMN,PMX
20    FORMAT(/' NBR. ZEROS =',I7,
     1       /' MEAN VALUE =',F12.4,
     1       /' SDEV VALUE =',F12.4,
     1       /' MIN. VALUE =',F12.4,
     1       /' MAX. VALUE =',F12.4)

      END
```

```
      SUBROUTINE MASCHK(TRN,COG,DET,PCT,NX,NY)
C-------------------------------------------------------------------------------
C     CALCULATE MASS BALANCE ERRORS IN PERCENT

      REAL     TRN(NX,NY),COG(NX,NY),DET(NX,NY),PCT(NX,NY)
      COMMON   MX,MY,DELX

C     STATEMENT FUNCTION FOR HARMONIC MEAN T
      HARMT(T0,T1) = 2.0/(1.0/T0 + 1.0/T1)

      DO J=2,NY-1
        CALL PROGSS('PCT MASS BAL ERR CALCULATIONS',J,NY)
        DO I=2,NX-1
          T0 = TRN(I,J)
          C0 = COG(I,J)
          D0 = DET(I,J)
          DPOS = 0.0
          DNEG = 0.0
          SUMQ = 0.0
          DO IAXS=1,2
            DO ICON=-1,1,2
              CALL INCRMT(I,J,IAXS,ICON,II,JJ)
              T1 = TRN(II,JJ)
              C1 = COG(II,JJ)
              D1 = DET(II,JJ)
              HMT = HARMT(T0,T1)
              CFLX = -HMT * (C1-C0)/DELX
              DFLX = -HMT * (D1-D0)/DELX
              IF(DFLX.GT.0) DPOS = DPOS + DFLX
              IF(DFLX.LT.0) DNEG = DNEG - DFLX
              SUMQ = SUMQ + CFLX
            END DO
          END DO
          GFLX = SQRT(DPOS*DNEG)

C         CHECK DETERMINISTIC STUFF JUST TO BE SURE ...
          DIFF = DPOS - DNEG
          DPCT = 100. * DIFF/DPOS
          DNCT = 100. * DIFF/DNEG
          IF(DPCT.GT.1 .OR. DNCT.GT.1) THEN
            WRITE(99,199) I,J,DPCT,DNCT
199         FORMAT(' ***** DET MASS ERR !!!    (I,J) =',2I5,
     1             5X,'POS PCT, NEG PCT =',2F12.4)
          END IF

          PERR = 100. * SUMQ/GFLX
          PCT(I,J) = PERR
        END DO
      END DO

      RETURN
      END


      SUBROUTINE INCRMT(I,J,IAXS,IADD,II,JJ)
C-------------------------------------------------------------------------------
C     INCREMENT INDEX I OR J ALONG IAXS BY AMOUNT IADD; RETURN IN II AND JJ

      II = I
      JJ = J

      IF(IAXS.EQ.1) II = II + IADD
      IF(IAXS.EQ.2) JJ = JJ + IADD

      RETURN
      END
```

```
      PROGRAM XY_XCOR
C-----------------------------------------------------------------------
C     CROSS-CORRELATION IN X OR Y DIRECTION BETWEEN TWO RANDOM FIELDS

      PARAMETER    (MXX=512*1024,MXY=1024,MXL=512)
      REAL          A(MXX), B(MXX), R(0:MXL),S(0:MXL)
      REAL          TA(MXY),TB(MXY)
      REAL          SMF(0:MXL),SMH(0:MXL),SMX(0:MXL),CPLS(0:MXL)
      REAL          RF(0:MXL), RH(0:MXL), RX(0:MXL)
      CHARACTER*40  FIL1,FIL2,OFIL
      COMMON       /PROMPT/ NX,NY,DX,ML


      CALL SPRMPT('ENTER INPUT FILENAME 1 > ',' ',FIL1,NC,1,IER)
      CALL SPRMPT('ENTER INPUT FILENAME 2 > ',' ',FIL2,NC,0,IER)
      CALL IPRMPT('ENTER NODES-X, NODES-Y > ',NX,2,N,0,IER)
      CALL RPRMPT('ENTER THE DELTA-X VALU > ',DX,1,N,0,IER)
      CALL IPRMPT('ENTER MAXIMUM NBR LAGS > ',ML,1,N,0,IER)
      CALL SPRMPT('ENTER OUTPUT  FILENAME > ',' ',OFIL,NC,0,IER)
      WRITE(*,*)'(1) - X  DIRECTION CROSS-COVARIANCE'
      WRITE(*,*)'(2) - Y  DIRECTION CROSS-COVARIANCE'
C     WRITE(*,*)'(3) - DO BOTH DIRECTIONS SEQUENTIALLY'
      READ (*,*) IDIR

      OPEN(UNIT=1,NAME=FIL1,STATUS='OLD',FORM='UNFORMATTED')
      CALL READU(1,A,NX,NY)
      CLOSE(UNIT=1)

      OPEN(UNIT=1,NAME=FIL2,STATUS='OLD',FORM='UNFORMATTED')
      CALL READU(1,B,NX,NY)
      CLOSE(UNIT=1)

      OPEN(UNIT=50,NAME=OFIL,STATUS='NEW',CARRIAGECONTROL='LIST')


      IF(IDIR.EQ.1 .OR. IDIR.EQ.3) THEN
        DO J=1,NY
          CALL LPLST(J,NY,'CROSS-COVARIANCE, X-DIRECTION')
          IADR = (J-1)*NX + 1
          CALL COVFHX(A(IADR),B(IADR),SMF,SMH,SMX,RF,RH,RX,
     1              CPLS,NX,ML)
        END DO
        CALL RRRFHX(SMF,SMH,SMX,RF,RH,RX,CPLS,ML)
        CALL PNTCOV(RF,RH,RX)
      END IF

      IF(IDIR.EQ.2 .OR. IDIR.EQ.3) THEN
        DO I=1,NX
          CALL LPLST(I,NX,'CROSS-COVARIANCE, Y-DIRECTION')
          DO J=1,NY
            IADR = (J-1)*NX + I
            TA(J) = A(IADR)
            TB(J) = B(IADR)
          END DO
          CALL COVFHX(TA,TB,SMF,SMH,SMX,RF,RH,RX,CPLS,NY,ML)
        END DO
        CALL RRRFHX(SMF,SMH,SMX,RF,RH,RX,CPLS,ML)
        CALL PNTCOV(RF,RH,RX)
      END IF

      END
```

```
      SUBROUTINE COVFHX(ZF,ZH,SMF,SMH,SMX,RF,RH,RX,CPL,NPTS,MLAG)
C-----------------------------------------------------------------------
C     CALCULATE ONE DIMENSIONAL AUTO AND CROSS COVARIANCES
C
C     ZF  AND  ZH = THE TWO 1D INPUT FIELDS
C     SMF,SMH,SMX = SUMMATION ARRAYS FOR EXPECTED VALUES
C     CPL  ARRAY  = TO ACCUMULATE THE NUMBER OF COUPLES
C     NPTS = LENGTH OF THE ZF AND ZH ARRAYS
C     MLAG = MAXIMUM LAG
C     RF, RH, RX  = AUTO AND CROSS COVARIANCE ARRAYS
C
C     CALL COVFHX REPEATEDLY AFTER EACH SIMULATION, THEN CALL RRRFHX TO
C     RETURN THE COVARIANCES CALCULATED OVER THE ENSEMBLE OF RANDOM FIELDS
C-----------------------------------------------------------------------
C >>> NOTE - THIS ASSUMES BOTH FIELDS ARE MEAN ZERO FIELDS <<<
C-----------------------------------------------------------------------
      REAL     ZF(*),ZH(*),RF(0:*),RH(0:*),RX(0:*)
      REAL     SMF(0:*),SMH(0:*),SMX(0:*),CPL(0:*)


      DO J=0,MLAG
        CPLS = NPTS-J
        SUMF = 0.0
        SUMH = 0.0
        SUMX = 0.0

        DO I=1,CPLS
          SUMF = SUMF + ZF(I)*ZF(I+J)
          SUMH = SUMH + ZH(I)*ZH(I+J)
          SUMX = SUMX + ZF(I)*ZH(I+J)
        END DO

        SMF(J) = SMF(J) + SUMF
        SMH(J) = SMH(J) + SUMH
        SMX(J) = SMX(J) + SUMX
        CPL(J) = CPL(J) + CPLS
      END DO

      RETURN


      ENTRY RRRFHX(SMF,SMH,SMX,RF,RH,RX,CPL,MLAG)
C-----------------------------------------------------
C     FINAL COVARIANCE CALULATIONS (OVER THE ENSEMBLE)

      DO J=0,MLAG
        RF(J) = SMF(J) / CPL(J)
        RH(J) = SMH(J) / CPL(J)
        RX(J) = SMX(J) / CPL(J)
      END DO

      RETURN
      END
```

```
      SUBROUTINE PNTCOV(RF,RH,RX)
C-----------------------------------------------------------------------
C     COMPUTE THEORETICAL COVARIANCES AND PRINT ALONG WITH DISCRETE ONES

      REAL      RF(0:*),RH(0:*),RX(0:*)
      COMMON   /PROMPT/ NX,NY,DX,ML


C     WRITE(50,30)
30    FORMAT(/'  LAG     THEORY-F  SAMPLE-F     THEORY-H ',
     1                    '   SAMPLE-H    THEORY-XC  SAMPLE-XC',
     1       /'  ----    --------  --------    ----------',
     1                    ' ----------  ---------  ---------')

      CMAX = DX*(ML-1)                    ! MAX NBR OF CORRELATION LENGTHS
      MX   = 59                           ! MX+1 OUTPUT POINTS
      A1   = ALOG10(CMAX/DX)              ! OTHER STUFF FOR COVARIANCE OUTPUT
      A2   = ALOG10(FLOAT(MX))
      PWR  = A1/A2

      WRITE(50,49)
49    FORMAT(/'        X        AUTO(1)     AUTO(2)   CROSS(1,2)',
     1       /'   ----------  ----------  ----------  ----------')

      DO L=0,MX
        M = FLOAT(L)**PWR
        X = M * DX
        WRITE(50,50) X,RF(M),RH(M),-RX(M)
50      FORMAT(4F12.4)
C        ARG = X*X / (2.0*ASQD)
C        EMA = EXP(-ARG)
C        TRF = VLNT*(1.-2.0*ARG)*EMA          ! THEORETICAL F COVARIANCE
C        TRH = VLNT*GRSQ*ASQD*EMA             ! THEORETICAL H COVARIANCE
C        TRX = VLNT*GRAD*X*EMA                ! THEORETICAL X-COVARIANCE
C        RXM = -RX(M)                         ! SIGN CHANGE ON CROSS-COVARIANCE
C        WRITE(50,40) X, TRF,RF(M), TRH,RH(M), TRX,RXM
C40      FORMAT(F6.2, 2X, 2F10.4, 1X, 2(F12.4), 3X, 2F10.4)
      END DO

      RETURN
      END
```

```
      PROGRAM RFF_PLOT
C-------------------------------------------------------------------------
C     PLOT THEORETICAL AND EXPERIMENTAL TELIS COVARIANCE FUNCTION
C     LINK  RFF_PLOT,OLD-IMSLD/LIB
C-------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
      DIMENSION       GMA(0:1),BMS(0:1)
      REAL*4          XE(99),CE(99), XT(99),CT(99), XINC,XMAX
      CHARACTER       REC*80,FILE*40,COMT*40
      EXTERNAL        F
      COMMON   /PASS/ X,ORDER

      IO = 6                          ! OUTPUT CHANNEL FOR ERROR MESSAGES
      A = 0.D0                        ! LOWER LIMIT OF INTEGERATION
      GMA(0) =  1.77245385100         ! GAMMA OF (+1/2)
      GMA(1) = -3.54490770200         ! GAMMA OF (-1/2)
      TRTPI = 1.12837916700           ! 2.0/SQRT(PI)
      TOVPI = 0.63661977200           ! 2.0/PI

      WRITE(*,*)'ENTER VARGRM OUTPUT FILENAME'
      READ (*,1) FILE
1     FORMAT(A)
      OPEN(UNIT=1,NAME=FILE,STATUS='OLD')

      WRITE(*,*)'ENTER DELTA-X'
      READ (*,*) DELX
      VLNT = 1.0
      GRAD = 1.0
      ZERO = 0.0

      WRITE(*,*)'ENTER COMMENTS'
      READ (*,1) COMT
      NC = NCHR(COMT)
      COMT(NC+1:NC+1) = 'S'

C     READ AWAY VARGRM PRINTER PLOT ETC TO GET TO DATA ...
C     ALSO CONVERT XE UNITS FROM NODES TO X-SPACING ...
      DO WHILE(INDEX(REC,'I  LAG').EQ.0)
        READ(1,1) REC
      END DO
      READ(1,*)
      DO I=1,99
        READ(1,*,END=2) TOSS,XE(I),TOSS,CE(I)
        XE(I) = DELX*XE(I)
      END DO
2     NPTS = I-1
      WRITE(*,10) XE(NPTS)
10    FORMAT(' MAX X =',F5.1,'  ENTER XINC AND XMAX FOR PLOT')
      READ (*,*) XINC,XMAX

C     CALCULATE THE THEORETICAL CURVE OVER NP+1 POINTS
      NP = 50
      DX = XE(NPTS)/FLOAT(NP)
      XT(1) = 0.0
      CT(1) = 1.0
      DO I=2,NP
        CALL LPLST(I,NP,'CALCULATING THEORETICAL CURVE')
        X =(I-1)*DX
        DO K=0,1
          ORDER = -DFLOAT(K)
          GAMMA = GMA(K)
          CNST = TRTPI*(X/2.D0)**ORDER/GAMMA
          CALL LONGMN(IO,F,A,N,SUM)
          BMS(K) = CNST * SUM
          IF(K.EQ.1) BMS(K) = BMS(K) + TOVPI
        END DO
        R11 = BMS(1) - TOVPI
        RFF = VLNT * ( BMS(0) + X*R11 )
        XT(I) = SNGL(X)
        CT(I) = SNGL(RFF)
      END DO
```

```
C       DISSPLA PLOT SETUP AND CURVE DRAW PREPARATION
11      CALL PSETUP(IDEV,XINC,XMAX,COMT)
        CALL GRACE(0.0)
        CALL THKCRV(0.024)
        CALL CURVE(XT,CT,NP,0)
        CALL GRACE(0.2)
        CALL SCLPIC(0.9)
        CALL MARKER(15)
        CALL CURVE(XE,CE,NPTS,-1)
        CALL ENDPL(0)
        CALL DONEPL
        IF(IDEV.NE.2) GO TO 11
99      STOP

        END



        SUBROUTINE PSETUP(IDEV,XINC,XMAX,COMT)
C-------------------------------------------------------------------------
C       DISSPLA PLOT SET UP

        CHARACTER    COMT*40

        WRITE(*,*)'(1) - VAX STATION'
        WRITE(*,*)'(2) - DISK FILE'
        READ(*,*,END=99) IDEV
        IF(IDEV.EQ.1) CALL UIS
        IF(IDEV.EQ.2) CALL COMPRS

        OPEN(UNIT=99,NAME='TOSS.MSG',STATUS='NEW')
        CALL SETDEV(99,99)
        CALL PAGE(8.5,11.)
        CALL NOBRDR
C       CALL AREA2D(7.0,4.5)
C       CALL BANGLE(90.)
C       CALL BSHIFT(5.75,-0.75)
        CALL AREA2D(4.5,6.0)

        CALL TRIPLX
        CALL HEIGHT(.16)
        CALL HEADIN(%REF(COMT),100,0.75,3)
        CALL HEADIN(' $',100,1.0,3)
        CALL HEADIN('2D Ln(T) Covariance Function$',100,1.,3)
        CALL XNAME('Lag$',100)
        CALL YNAME('Covariance$',100)

        CALL GRAF(0.0,XINC,XMAX, -0.2,0.2,1.0)
        CALL THKFRM(0.015)
        CALL FRAME
        CALL DOT
        CALL RLVEC(0.,0.,XMAX,0.,0)
        CALL RESET('DASH')

        RETURN
99      STOP
        END



        FUNCTION F(T)
C-------------------------------------------------------------------------
C       MONOTONICALLY DECREASING SIGN-ALTERNATING FUNCTION

        IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
        COMMON   /PASS/ X,ORDER                         ! FUNCTION SPECIFIC


        F = SIN(X*T) * (1+T*T)**(ORDER-0.5D0)           ! FUNCTION SPECIFIC

        RETURN
        END
```

```
      SUBROUTINE LONGMN(IO,F,A,N,SUM)
C-----------------------------------------------------------------------
C     PAUL HSIEH'S VERSION OF LONGMAN'S METHOD FOR INTEGRATING A MONOTONICALLY
C     DECREASING ALTERNATING SERIES FROM X=A TO X=INFINITY

C     IO = OUTPUT CHANNEL FOR ERROR MESSAGES
C     F  = NAME OF SIGN-ALTERNATING FUNCTION
C     A  = LOWER LIMIT OF INTEGRATION
C     N  = NUMBER OF HALF CYCLES INTEGRATED OVER FOR CONVERGENCE
C     SUM = THE INTEGRAL OF F FROM A TO INFINITY
C     ERR,EPS1 = CONVERGENCE CRITERIA FOR LONGMAN SUM OR DIRECT SUM
C     DCADRE = IMSL ROUTINE FOR CAUTIOUS ADAPTIVE ROMBERG INTEGRATION

      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
      PARAMETER       NMAX=100
      EXTERNAL        F
      DIMENSION       D1(NMAX),D2(NMAX)
      DATA            AERR,RERR,EPS1,ERR /0.D0, 1.D-04, 1.D-04, 1.D-06/

      N = 0
      X1 = A
      SUMD = 0.D0
      SUML = 0.D0
      CALL  LIMITS(IO,F,X0,X1,N)
      UU0 = DCADRE(F,X0,X1,AERR,RERR,ERROR,IERR)
      CALL IERCHK(IO,IERR)
      SUMD = UU0
100   CALL LIMITS(IO,F,X0,X1,N)
      UU = DCADRE(F,X0,X1,AERR,RERR,ERROR,IERR)
      CALL IERCHK(IO,IERR)
      IF(UU.LT.0.D0)              GO TO 110
      IF(DABS(UU).LT.DABS(UU0))   GO TO 200
110   SUMD = SUMD + UU
      UU0 = UU
      GO TO 100                 ! GO TO 100
200   UUU = UU0*UU
      IF(UUU.GE.0.D0)             GO TO 270
      D1(1) = UU
      SUML = SUMD + D1(1)/2.D0
      SUMD = SUMD + UU
      ID = 1
      DENOM = -4.D0
220   CALL LIMITS(IO,F,X0,X1,N)
      UU = DCADRE(F,X0,X1,AERR,RERR,ERROR,IERR)
      CALL IERCHK(IO,IERR)
      SUMD = SUMD + UU
      D2(1) = DABS(UU)
      DO 230 I=1,ID
230   D2(I+1) = D2(I)-D1(I)
      IDP1 = ID + 1
      VV = D2(IDP1)/DENOM
      SUML = SUML + VV
      IF(DABS(UU).LT.ERR)         GO TO 300
      IF(DABS(2.D0*VV).LT.ERR)    GO TO 310
      IF(N.EQ.NMAX)               GO TO 260
      DO 250 I=1,IDP1
250   D1(I) = D2(I)
      ID = IDP1
      DENOM = DENOM * (-2.D0)
      GO TO 220                 ! GO TO 220
260   SUM = SUMD
      WRITE(IO,261) NMAX
261   FORMAT(/' ***** NMAX CYCLES REACHED IN LONGMN',I5)
      RETURN
270   SUMD = SUMD + UU
      DIFF = SUMD - UU0
      IF(DIFF.LE.EPS1)            GO TO 271
      UU0 = SUMD
      GO TO 100                 ! GO TO 100
271   SUM = SUMD
      RETURN
300   SUM = SUMD
      RETURN
310   SUM = SUML
      RETURN
      END
```

```
      SUBROUTINE IERCHK(IO,IERR)
C-----------------------------------------------------------------------
C     CHECK IERR FLAG FROM IMSL ROUTINE DCADRE

      IF(IERR.NE.0 .AND. IERR.NE.65)  THEN
        WRITE(IO,10) IERR
10      FORMAT(/' ***** DCADRE FAILED, IERR =',I4)
        STOP
      END IF

      RETURN
      END



      SUBROUTINE LIMITS(IO,F,X0,X1,N)
C-----------------------------------------------------------------------
C     RETURN LIMITS OF INTERVAL BOUNDING A HALF CYCLE OF ALTERNATING SERIES

C     IO = OUTPUT CHANNEL FOR ERROR MESSAGE
C     F  = NAME OF SIGN-ALTERNATING FUNCTION
C     X0 = CURRENT ZERO OF FUNCTION F (INPUT)
C     X1 = NEXT ZERO FOR FUNCTION F  (OUTPUT)
C     N  = NUMBER OF ZEROS COMPUTED SO FAR

      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
      PARAMETER       PI=3.14159265400          ! FUNCTION SPECIFIC
      EXTERNAL        F

      COMMON   /PASS/ X,ORDER                   ! FUNCTION SPECIFIC


      N  = N + 1                                ! THESE TWO STATEMENTS MUST
      X0 = X1                                   ! REMAIN IN THIS ORDER

      X1 = PI*N/X                               ! FUNCTION SPECIFIC

C     CALL NXTZRO(IO,F,X0,DX,X1)                ! WHEN THE NEXT ZERO CANNOT
                                                ! BE COMPUTED ANALYTICALLY
                                                ! DX WILL NEED TO BE PASSED

      RETURN
      END
```