T-PROGS: Transition Probability Geostatistical Software, Version 2.1

Article · January 1999

CITATIONS
236

READS
1,048

1 author:

Steven F. Carle
Lawrence Livermore National Laboratory
62 PUBLICATIONS **2,597** CITATIONS

SEE PROFILE

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-SM-232880

T-PROGS: Transition Probability Geostatistical Software Version 2.1

S. F. Carle

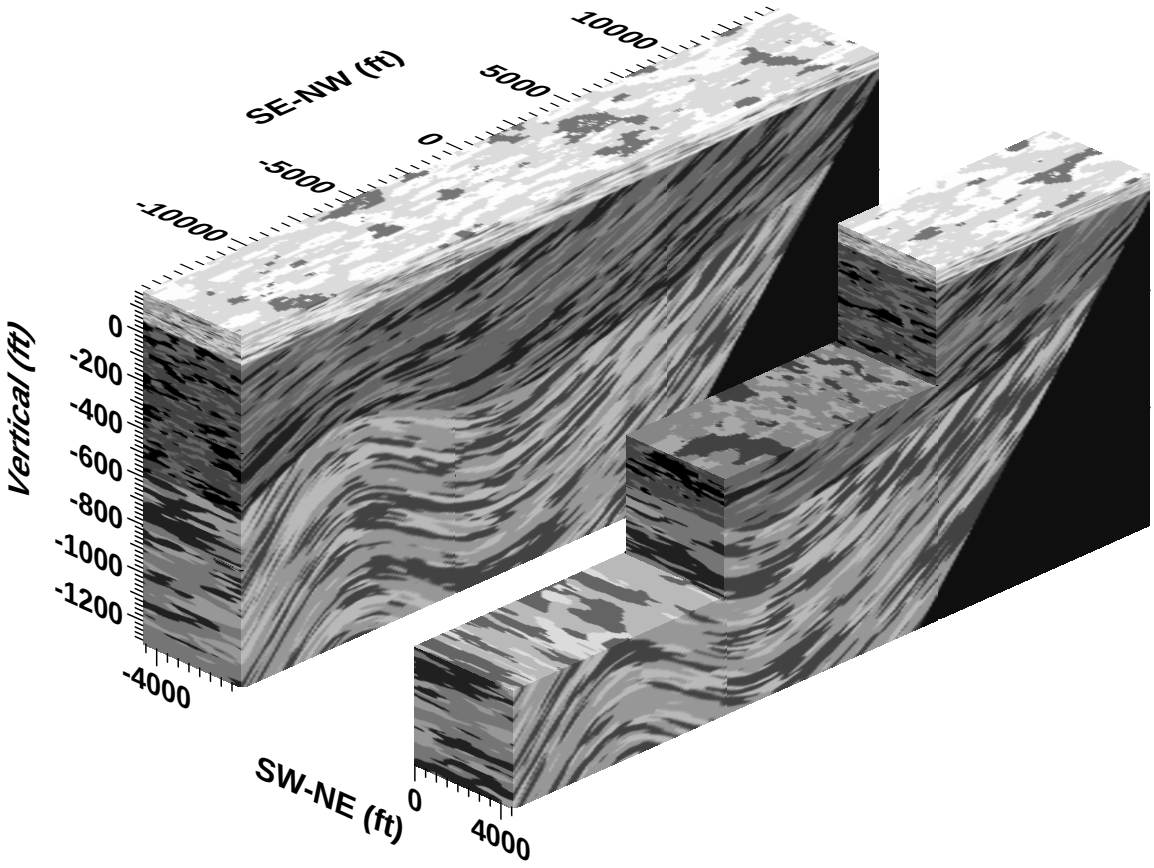
July 19, 2007

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

T-PROGS:

Transition Probability Geostatistical Software

Version 2.1



STEVEN F. CARLE
Hydrologic Sciences Graduate Group

University of California, Davis

Preface

The purpose of T-PROGS is to enable implementation of a transition probability/Markov approach to geostatistical simulation of categorical variables. In comparison to traditional variogram-based geostatistical methods, the transition probability/Markov approach improves consideration of spatial cross-correlations and facilitates the integration of geologic interpretation of facies architecture into the model development process. The manual was designed primarily for geostatistical practitioners, not theoreticians. In our experience, geostatistics is not the primary occupation of most users of geostatistical simulation codes. As such, the manual relies on references for much of the theoretical details. The T-PROGS computer source codes are provided without any warranty or guarantee of freedom from bugs. On the other hand, the accessibility of the source code frees the user to make any modifications as needed. An effort has been made to achieve a high degree of platform independence, however the responsibility rests upon the user to make any specific or system-dependent changes in the FORTRAN code or PostScript graphical output. The user should take responsibility for properly compiling the codes, checking dimensioning of arrays, constructing parameter files, understanding the theory behind the algorithms, and modifying input or output formats for interfacing with other programs. Questions not addressed in this manual as well as comments on the manual or code may be e-mailed to carle1@llnl.gov.

Contents

Acknowledgments

Large portions of T-PROGS originated from modified versions of the GSLIB codes. Clayton Deutsch, Andre Journel, and the other GSLIB contributors deserve much credit for developing much robust FORTRAN code that remains intact in T-PROGS. Many improvements and breakthroughs were incited by feedback from Graham Fogg, Eric Labolle, Gary Weissmann, and David Van Brocklin at UC Davis. Graham Fogg's steadfast promotion of T-PROGS has been greatly appreciated. This work was supported by U.S. Army Waterways Experimentation Station, Vicksburg, Mississippi; Lawrence Livermore National Laboratory, Livermore, California; N.I.E.H.S. Superfund Grant (ES-04599); U.S.G.S. Water Resources Research Grant (14-18-001-61909); and U.S. EPA (R819658) Center for Ecological Health Research at UC Davis. Although the information in this document has been funded in part by the United States Environmental Protection Agency, it may not necessarily reflect the views of the Agency, and no official endorsement should be inferred.

1 Summary

Transition Probability Geostatistical Software (T-PROGS) is a set of FORTRAN computer programs that implements a transition probability/Markov approach to geostatistical analysis and simulation of spatial distributions of categorical variables (e.g., geologic units, facies). Implementation of T-PROGS involves three main steps: (a) calculation of transition probability measurements, (b) modeling spatial variability with Markov chains, and (c) conditional simulation. These steps are accomplished by the following programs:

- **GAMEAS** computes bivariate statistics (e.g., transition probability, indicator cross-variogram, etc.).
- **MCMOD** develops one- and three-dimensional Markov chain models of spatial variability.
- **TSIM** generates three-dimensional, cross-correlated conditional simulations.

The transition probability/Markov approach was developed to facilitate incorporation of geologic interpretation and improve consideration for spatial cross-correlations (juxtapositional tendencies) in the development of geostatistical models. Further details on theory, examples, and comparison to other geostatistical methods are given in Carle (1996), Carle and Fogg (1996), Carle (1997a), Carle (1997b), Carle and Fogg (1997), and Carle and others (1998).

The graphical display of results may be produced with FORTRAN computer programs that generate "PostScript" (PS) graphics files (Adobe Systems Incorporated, 1990):

- **GRAFXX** plots a matrix of one-dimensional (along a single direction) bivariate statistics.
- **CHUNK** displays a three-dimensional perspective of the conditional simulation.

The T-PROGS implementation process, from data to producing simulation results and graphical output, is shown in Figure 1. The PS files may be converted to "Encapsulated PostScript" (EPS) using a program called **ps2eps.f(or)**, which facilitates inclusion into text-processing and graphics presentation programs. The PS and EPS files can also be printed directly to a printer having a PostScript driver or viewed on screen with a PostScript viewer such as "Ghostview."

The general style of the program execution is analogous to the Geostatistical Software Library (GSLIB) by Deutsch and Journel (1992), whereby parameter files are prepared to administer input data for the executable codes. Indeed, **GAMEAS** and **TSIM** originated from GSLIB codes, and **GRAFXX** and **CHUNK** contain aspects of GSLIB code as well. Two main data formats are used, one for point data and the other for gridded data. Point data, in particular coded lithologies located in an x, y, z coordinate system or bivariate statistics computed as a function of lag (variograms, transition probabilities, etc.), are stored in a free-format "GEOEAS" ASCII format. Grid data, in particular 3-D Markov chain models and conditional simulations, are stored in a compact binary format. The simulations can also be output in an ASCII format to promote portability. The PS and EPS graphics files are also produced in ASCII format, which

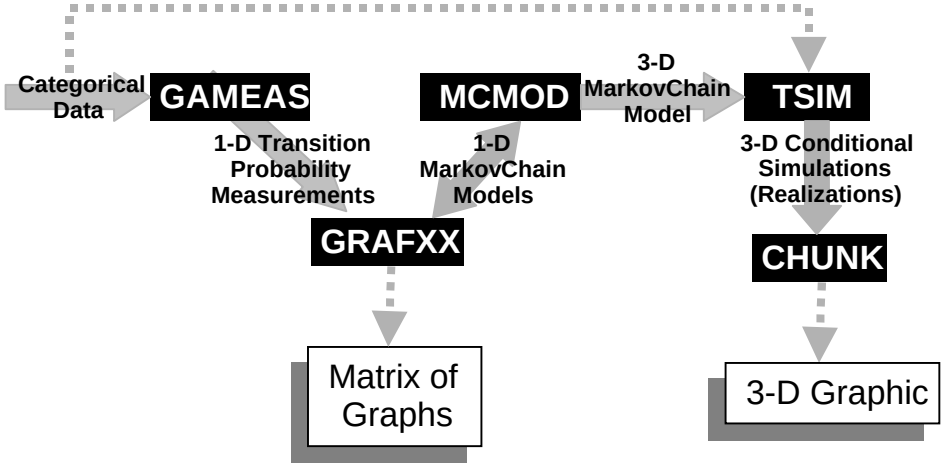


Figure 1. Schematic diagram showing implementation of T-PROGS.

provides opportunity for direct manipulation of graphical output given some understanding of PostScript.

The general style of this manual is designed to facilitate application to real problems. More often than not, the user will have a data set in mind, with a goal of developing a model of a heterogeneous geologic system. Therefore, the T-PROGS manual is organized to accommodate the chronological progression of a typical application.

2 Background

T-PROGS offers a transition probability-based geostatistical approach to stochastic *conditional simulation* of spatial distributions of categorical variables. T-PROGS can be used to analyze spatial variability and generate *realizations* of geologic units or *facies*. Importantly, the realizations attempt to honor existing data and display consistency with the spatial variability evident in data or other geologic observations.

The overall goal of T-PROGS is to simplify conceptual aspects of geostatistical modeling, yet maximize theoretical potential. Considering that potential users of T-PROGS will have varying backgrounds, here is some general advice:

- To those who are not familiar with geostatistics: Fear not! You do not need to know anything about variograms. T-PROGS emphasizes the extension of general and intuitive concepts from probability theory to spatial problems.
- To experienced geostatisticians: Be flexible! T-PROGS conceptualizes geostatistical models in a more interpretive framework than variogram-based geostatistical approaches. For example, the transition probability models are related to concepts of *proportions* and *mean length* as compared to the parameters of "sill" and "range" used in variogram modeling.

To this end, T-PROGS is designed to appeal to geologists and geostatisticians alike.

The "Traditional" Approach

Consider that "traditional" geostatistics evolved from mining industry applications, where intensively sampled data sets abound. In this respect, the implementation of traditional geostatistical methods has adopted the following rather empirical approach:

1. Calculate values of a spatial statistic (usually the variogram) at regularly-spaced lags (separation vectors).
2. Fit a mathematical function (e.g., spherical, exponential) through the variogram measurements.
3. Implement various estimation (kriging) or simulation (sequential simulation, simulated annealing) procedures.

Geologic or "subjective" knowledge does not necessarily enter directly into this procedure.

In the application of geostatistics to other geologic disciplines involving more sparsely sampled variables, such as permeability, the procedures are not as straightforward. In many geologic applications, the parameter at the scale of interest is more conveniently interpreted in a categorical framework, for example:

- **petroleum** – lithologies indicating reservoir, source, trap, or non-oil bearing rocks
- **hydrogeology** – hydrofacies or hydrostratigraphic units indicating water-bearing zones (aquifer), aquitard, or aquiclude materials
- **mineral** – classifications based on grade, degrees of mineralization, or specific mineralization phases

"Indicator" geostatistical approaches were developed to address categorical applications, as well as to provide "non-parametric" models for continuous variables (Journel, 1983).

In the practical application of either the continuous or categorical geostatistical approaches, *geologic data sets rarely provide the necessary detail to directly implement the empirical variogram curve-fitting procedure traditionally employed.* If data are too sparse (or the geology is too complicated) to calculate meaningful variograms values, then how can one implement a geostatistical analysis? The usual advise is to infuse more understanding of the geology (e.g., characteristics of depositional systems, facies architecture, stratigraphy), for "...it is subjective interpretation .... that makes a good model; the data, by themselves, are rarely enough..."(Deutsch and Journel, 1992). However, the prevalent means for infusing geology into geostatistics has been to obtain a "reference image" or "training image" (e.g. Deutsch and Journel, 1992, p. 119, 161, 189; Almeida and Journel, 1994, Goovaerts, 1996), a picture of the geology which provides a surrogate for the exhaustive data set. With the training image at hand, the geostatistician can then implement the usual empirical curve-fitting variogram modeling procedure. However, not all applications are graced with a site-specific training image, particularly in 3-D.

Does this rule out the practical applicability of geostatistics to typically sparse geologic data sets? Geostatistics seems to offer a promising tool for addressing uncertainty and scaling issues that inevitably occur as a result of sparse data and geologic complexity. How then can subjective information be directly infused into the geostatistical modeling procedure?

The Transition Probability Approach

Some key answers to the problems of practical application of categorical (indicator) geostatistics can be found by linking model parameters to basic observable attributes, which, for categorical variables, are:

- volumetric proportions
- mean lengths (e.g., mean thickness in the vertical direction)
- juxtapositional tendencies (how one category tends to locate in space relative to another)
- anisotropy directions
- spatial variations of the above

In this light, T-PROGS was developed to encourage infusion of subjective interpretation by simplifying the relationship between observable attributes and model parameters. Understanding the impacts of model parameters will improve conditional simulation results whether data are abundant or sparse. The main simplification is to incorporate the transition probability instead of the indicator cross-variogram as the measure of spatial variability. The transition probability

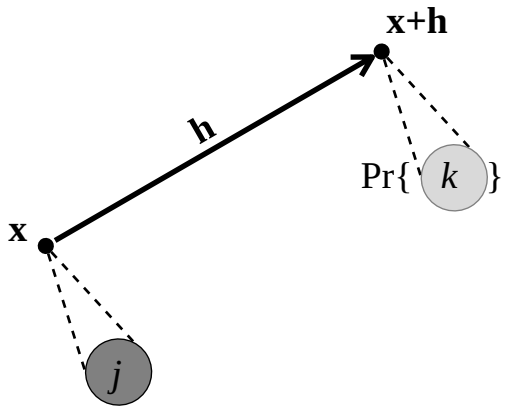
$t_{jk}(\mathbf{h})$ is defined by

$$t_{jk}(\mathbf{h}) = \Pr\{k \text{ occurs at } \mathbf{x}+\mathbf{h} \mid j \text{ occurs at } \mathbf{x}\} \tag{1}$$

where $\mathbf{x}$ is a spatial location, $\mathbf{h}$ is the lag (separation vector), and j,k denote mutually exclusive categories such as geologic units or facies. Indeed, the definition of the transition probability is simple enough to put into words:

> Given that a facies j is present at a location $\mathbf{x}$, what is the probability that another (or the same) facies k occurs at location $\mathbf{x}+\mathbf{h}$?

or, schematically:



The transition probability originates from the definition of a *conditional* probability

$$\Pr\{B'|A\} = \frac{\Pr\{A \text{ and } B'\}}{\Pr\{A\}} \tag{2}$$

where 'A' would represent $\{j$ occurs at $x\}$ and 'B'' would represent $\{k$ occurs at $x+h\}$.

Comparison to the Indicator (Cross-) Variogram

Traditional indicator geostatistics employs the indicator cross-variogram $\gamma_{jk}(\mathbf{h})$ bivariate statistic defined as

$$\gamma_{jk}(\mathbf{h}) = \frac{1}{2}E\{[I_j(\mathbf{x}) - I_j(\mathbf{x}+\mathbf{h})]\,[I_k(\mathbf{x}) - I_k(\mathbf{x}+\mathbf{h})]\} \tag{3}$$

where the indicator variable $I_j(\mathbf{x})$ denotes

$$I_j(\mathbf{x}) = \begin{cases} 1, & \text{if category } j \text{ occurs at } \mathbf{x} \\ 0, & \text{otherwise} \end{cases}$$

The transition probability can also be defined with respect to indicator variables as

$$t_{jk}(\mathbf{h}) = \frac{E\{I_j(\mathbf{x})I_k(\mathbf{x}+\mathbf{h})\}}{E\{I_j(\mathbf{x})\}}$$

With analogy to a conditional probability (2), the indicator cross-variogram $\gamma_{AB'}$ could be defined as

$$\gamma_{AB'} = \frac{1}{2}\left[\Pr\{A \text{ and } B\} - \Pr\{A \text{ and } B'\} - \Pr\{A' \text{ and } B\} + \Pr\{A' \text{ and } B'\}\right] \quad (4)$$

where 'A' would represent $\{j$ occurs at $x\}$, 'A'' would represent $\{j$ occurs at $x+h\}$, 'B' would represent $\{k$ occurs at $x\}$, and 'B'' would represent $\{k$ occurs at $x+h\}$. Although both the transition probability and indicator (cross-) variogram measures carry similar statistical information, the transition probability definitions (1) and (2) are simpler and, as will be demonstrated in later examples, more interpretable than the respective indicator variogram definitions (3) and (4).[1]

The transition probability approach further empowers the geostatistical method by considering all juxtapositional (cross-correlation) information, which has been otherwise considered tedious and impractical in the variogram approaches (Deutsch and Journel, 1992, p. 68-69, p. 82). The transition probability allows for the possibility of asymmetry, $t_{jk}(\mathbf{h}) \neq t_{jk}(-\mathbf{h})$, whereas the indicator cross-variogram assumes symmetry, $\gamma_{jk}(\mathbf{h}) = \gamma_{jk}(-\mathbf{h})$. Asymmetry would be evident in a stratigraphic sequence that displays juxtapositional tendencies of $ABCABC$, such as a fining-upward tendency, because the same sequence viewed in the reverse direction would appear as $CBACBA$. Considering that many geologic systems display asymmetries such as fining or coarsening-upward tendencies, the transition probability can be a more informative and diagnostic statistic than the indicator (cross-)variogram.

Markov Chain Analysis

Markov chains offer an interpretable and mathematically simple yet powerful stochastic model for categorical variables. In time-series applications, the Markov chain model assumes, in theory, that *the future depends on the present and not the past*. Analogously for 1-D spatial applications, the Markov chain assumes that spatial occurrences depend entirely on the nearest data. The Markov chain model is appealing for geostatistical applications because it offers straightforward means for developing "coregionalization" models to account for all spatial cross-correlations.

Embedded Markov Chains

Most geological applications of Markov chains have employed an *embedded* analysis, in which a matrix of vertical (z)-direction transition probabilities of *embedded* occurrences, i.e., from one *discrete* occurrence of a facies to another, is considered (e.g., Carr and others, 1966; Krumbein and Dacey, 1969; Doveton, 1971; Miall, 1973; Ethier, 1975).

To illustrate the concept of an embedded Markov chain analysis, Figure 2 shows a vertical succession of three categories, say $A = white$ (sand), $B = gray$ (silt), $C = black$ (clay), as

[1] Indicator geostatistics can also be formulated in yet simpler statistical terms by the *joint probability* defined as $E\{I_j(x)I_k(x+h)\}$ or $\Pr\{j$ occurs at x*and* k occurs at $x+h\}$ (Carle and Fogg, 1996). However, the transition probability is more interpretable (as a conditional probability) and has a long history of usage in the geosciences.

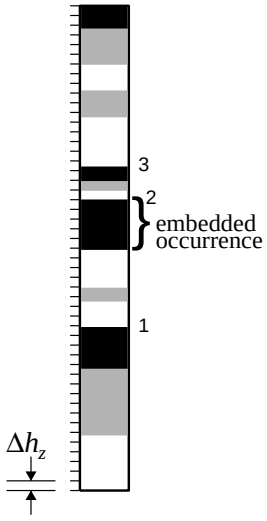


Figure 2. Diagram showing embedded occurrences of a three-category system with $1 = white$, $2 = gray$, $3 = black$. Count of embedded transitions from category 3 to category 1 show to right of $3 \rightarrow 1$ contact.

might be encountered in a borehole or a cliff face. To implement an embedded Markov chain analysis, one must:

1. Forget about lag or spatial dependency and relative thicknesses of the beds.
2. Record the succession of "embedded occurrences," that is, simply log each occurrence of sand, silt, or clay in the vertical succession, which would might look something like: $ABCABACABCABABC$.
3. Tally up the transition count matrix, which for the succession above would be

$$\begin{bmatrix} - & 5 & 1 \\ 2 & - & 3 \\ 3 & 0 & - \end{bmatrix}$$

   The diagonal elements are blank because "self-transitions," e.g. from A to A, are unobservable. That is, stacked beds of the same category are assumed not distinguishable from a single bed. The "embedded occurrence" term refers to the a discrete occurrence of A, which may consist of either a single bed or stacked beds.
4. Divide each row by the row sum to obtain the embedded transition probabilities.

$$\begin{bmatrix} - & 0.833 & 0.167 \\ 0.40 & - & 0.60 \\ 1.0 & 0 & - \end{bmatrix}$$

One of the goals of MCMOD, the 3-D Markov chain modeling program in T-PROGS, is to link the embedded Markov chain analysis to the development of *continuous-lag* (spatially

dependent) Markov chain models. The reason this is important is that geologists are more inclined to think and work in the embedded framework. In this example, there are no self-transitions because stacked beds of the same category are assumed to be indistinguishable from a single bed. It might be possible for geologists to distinguish individual beds associated with discrete depositional events, and an embedded Markov chain analysis can be performed in that context as well. However, for most data sets and practical applications, the self-transitions are considered "unobservable." In the context of modeling a flow system, whether the flow unit consists of one massive bed or stacked beds of the same facies usually would not make much difference.

A real example of an embedded Markov chain analysis is given by Ethier (1975), who computed an embedded transition probability matrix for vertically successive occurrences of five rock units in the Pigeon-Grotto section of the Banff Formation, Alberta, Canada as

$$\mathbf{T}_z = \begin{bmatrix} t_{11} & \cdots & t_{1K} \\ \vdots & \ddots & \vdots \\ t_{K1} & \cdots & t_{KK} \end{bmatrix} = \begin{bmatrix} - & 0.087 & 0.391 & 0.326 & 0.196 \\ 0.357 & - & 0.143 & 0.0 & 0.500 \\ 0.643 & 0.143 & - & 0.0 & 0.214 \\ 1.0 & 0.0 & 0.0 & - & 0.0 \\ 0.364 & 0.318 & 0.318 & 0.0 & - \end{bmatrix}$$

where diagonal or "self" transitions are considered unobservable.

Spatial Markov Chains

A spatial dependency can also be incorporated into a Markov chain analysis. As such, Markov chains can be used as geostatistical models of spatial variability.

Most geological applications of spatial Markov chains have considered vertical (z)-direction transition probabilities at a fixed sampling interval or "discrete lag," say Δh_z as shown in Figure 2 (e.g. Krumbein and Dacey, 1969; Schwarzacher, 1969; Ethier, 1975). For the same Pigeon-Grotto section above, Ethier (1975) computed a transition probability matrix $\mathbf{T}(\Delta h_z) = \Pr\{k \text{ occurs at } x + \Delta h_z \mid j \text{ occurs at } x\}$ for a 5-ft sampling interval as

$$\mathbf{T}(\Delta h_z = 5 \text{ ft }) = \begin{bmatrix} t_{11}(\Delta h_z) & \cdots & t_{1K}(\Delta h_z) \\ \vdots & \ddots & \vdots \\ t_{K1}(\Delta h_z) & \cdots & t_{KK}(\Delta h_z) \end{bmatrix} = \begin{bmatrix} 0.63 & 0.11 & 0.0 & 0.04 & 0.22 \\ 0.16 & 0.48 & 0.04 & 0.0 & 0.32 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.14 & 0 & 0.14 & 0.57 & 0.14 \\ 0.05 & 0.11 & 0.03 & 0.0 & 0.81 \end{bmatrix}$$

The diagonal entries represent the transition probabilities from one category to itself, and the off-diagonal entries represent the transition probabilities from one category to another. As a matter of basic probability theory the row sums in any transition probability matrix should equal unity

$$\sum_{k=1}^{K} t_{jk}(h) = 1 \qquad \forall j$$

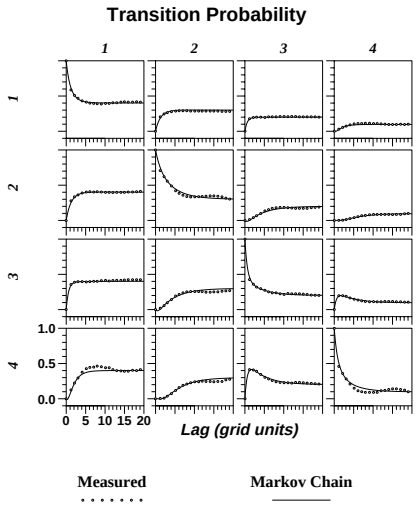


Figure 3. Matrix of transition probability measurements and models.

and, assuming stationarity, the column sums should obey

$$\sum_{k=1}^{K} p_j t_{jk}(h) = p_k \qquad \forall k$$

where p_j denotes the proportions or "marginal probabilities." Furthermore, the transition probability "sill," i.e. $\lim_{h\to\infty} t_{jk}(h)$, will converge on the column category proportion

$$\lim_{h\to\infty} t_{jk}(h) = p_k \tag{5}$$

for a stationary random field.

In one dimension, say along the vertical z, the complete set of spatial auto- and cross-correlations for K categories can be represented by a $K \times K$ matrix $\mathbf{T}(h_z)$ of transition probabilities as a function of lag h_z

$$\mathbf{T}(h_z) = \begin{bmatrix} t_{11}(h_z) & \cdots & t_{1K}(h_z) \\ \vdots & \ddots & \vdots \\ t_{K1}(h_z) & \cdots & t_{KK}(h_z) \end{bmatrix}$$

Thus, for a particular direction z, $\mathbf{T}(h_z)$ consists of a matrix of graphs representing transition probabilities from one category to another or to the same category as a function of lag separation, as shown in Figure 3 for the four-category system defined by Goovaerts (1996) from the 'true.dat' data set given in Deutsch and Journel (1992). The transition matrix can be made a function of a lag *vector* $\mathbf{h} = (h_x, h_y, h_z)$ as well, thus enabling application of the transition probability $\mathbf{T}(\mathbf{h})$ as a measure of 2- or 3-D spatial variability.

In theory, the discrete-lag Markov chain model assumes that the spatial variability can be characterized entirely by a transition probability matrix at a fixed lag interval, such as the 5-ft transition probability matrix for the Pigeon-Grotto Section above. Mathematically, the Markov

property is evident when $\mathbf{T}(\mathbf{h})$ depends entirely on *transition rates*, explained in more detail in Chapter 6. In practice, geologic data do not conform exactly to mathematical or probability theory, so that implementation and relevance of Markov chain models is not automatic. Nonetheless, the conceptual simplicity of Markov chains can facilitate and strengthen the application of geostatistics by

- making practical the development of coregionalization models,
- illuminating the relationship between model parameters and spatial structure, thus providing means for integrating geologic interpretation, and
- ensuring that the models of spatial variability are consistent with probability law.

Conditional Simulation

Conditional simulation is a process that creates multiple, equally probable spatial distributions of random variables or "realizations" that honor hard data at specified locations (Deutsch and Journel, 1992, p. 117). Although (co)kriging may be used in the algorithms, conditional simulation should not be confused with interpolation. From a geologic perspective, 2-D conditional simulation of categorical variables, such as geologic units, can be viewed as a quantitative approach to the classic problem of drawing a geologic cross-section that realistically represents geologic architecture between locations of control, such as outcrops or boreholes. In practice, construction of a geologic cross-section requires a reconciliation of the available data with an understanding of appropriate stratigraphic relationships in order to produce a plausible representation of the geologic system. The same requirements should also hold true for producing a geostatistical realization; the methodology should be able to reconcile patterns of spatial variability evident in the data and generate patterns of heterogeneity that are geologically plausible. Otherwise, the realizations obtained, although equally probable, may be highly *improbable*. Thus, the aim of conditional simulation, as illustrated in Figure 4 for the 'true.dat' data set examined by Goovaerts (1996), is to generate spatial distributions that honor hard data and exhibit a realistic pattern of spatial variability.

Either a hand-drawn cross-section or a conditional simulation may serve as a representation of geologic heterogeneity or, possibly, a template of hydraulic properties for flow and transport modeling. Whereas the manual approach is sometimes feasible in 2-D, the 3-D situation requires automated or computer-assisted methods. Yet automated methods should project some degree of geologic insight that a geologist would subjectively infuse into a hand-drawn cross-section. If a conditional approach can succeed in producing geologically plausible outcomes, two distinct advantages over a manual approach emerge: (1) applicability to 3-D problems, and (2) capability to produce an infinity of alternatives, thus providing a tool for assessing uncertainty.

In the petroleum industry, 3-D conditional simulations may serve as building blocks for "reservoir models" to evaluate efficiency and uncertainty in recovery schemes. Analogously in hydrogeology, conditional simulations may prove useful for developing realistic aquifer system models to evaluate impacts of heterogeneity on ground-water flow and contaminant transport.

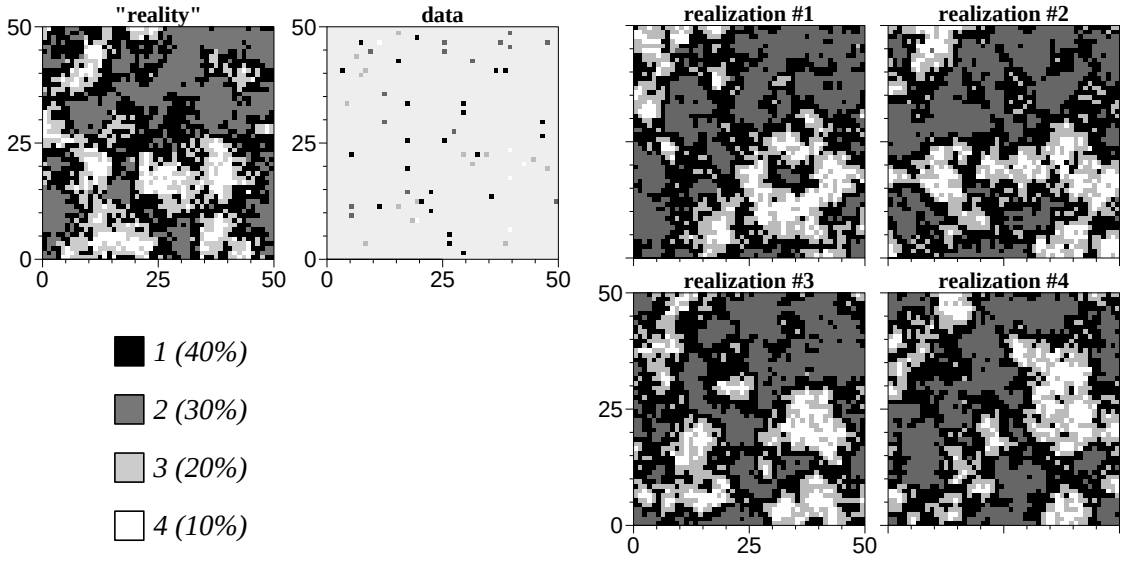


Figure 4. The concept of conditional simulation - to generate multiple “realizations” that honor data and exhibit a realistic pattern of spatial variability.

3 Data Formats

Data must be placed in a specific format to run the T-PROGS programs. Two formats are used exclusively, a "GEOEAS" format for point [x, y, z, attribute(s)] data and a binary format for grid (array) data.

GEOEAS

The "GEOEAS" format, also employed in GSLIB, handles point data with a flexible ASCII convention. T-PROGS uses this format for storing data locations and 1-D measured and modeled transition probability values as a function of lag. A ***.eas** filename suffix designation is recommended to signify a GEOEAS-format file. For example, Figure 5 shows an example GEOEAS-format data file excerpt which prescribes *x, y,* and *z* locations and probabilities (indicator values) for four ($K = 4$) categories as described in Table 1. The data from lines $(6 + K)$ to END is read in by free format in all of the T-PROGS programs and, thus, may be stored in various columnar formats.

Data consist of the x, y, z locations in the first three columns and *probability values*, which should range from zero to unity, in the last four (K) columns. Thus, each data line records the location and probability that one of the four categories occurs at the location (Figure 5). If a datum is "hard," indicating the absolute presence of the floodplain unit (category 2), the probability values will consist of (0,1,0,0). This format leaves open the possibility of "soft"

```
Data
7
 x    = easting
 y    = northing
 z    = elevation above mean sea level
1 = debris flow
2 = floodplain
3 = levee
4 = channel
   2132.8    2487.4    137.07   0   1   0   0
   2132.8    2487.4    136.77   0   1   0   0
   2132.8    2487.4    136.47   0   1   0   0
   2132.8    2487.4    136.17   0   1   0   0
   2132.8    2487.4    135.87   1   0   0   0
   2132.8    2487.4    135.57   1   0   0   0
   2132.8    2487.4    132.27   0   1   0   0
   2132.8    2487.4    131.97   0   1   0   0
   2576.2    2695.5    186.48   0   1   0   0
   2576.2    2695.5    182.28   0   0   0   1
   2576.2    2695.5    181.98   0   0   0   1
   2576.2    2695.5    181.68   0   0   0   1
   2576.2    2695.5    181.38   0   0   0   1
   2576.2    2695.5    181.08   0   1   0   0
   2576.2    2695.5    175.98   1   0   0   0
   2576.2    2695.5    175.68   0   1   0   0
   2576.2    2695.5    175.38   0   1   0   0
   2576.2    2695.5    112.98   0   1   0   0
.
.
.
```

Figure 5. Example file showing GEOEAS format for storing data locations and probability values.

Line	Description
1	text describing the contents of the file or other relevant information
2	number of data columns ($= 3 + K$) for storing x, y, z locations and K data values
3 to $(5 + K)$	text describing the contents of each data
$(6 + K)$ to END	data: *x, y,* and *z* coordinates and K values associated with each data point

Table 1. Description of GEOEAS format for point data.

```
 0.0668 0.5623 0.1883 0.1821
17
 Lag
 1- 1 transition probability
 1- 2 transition probability
 1- 3 transition probability
 1- 4 transition probability
 2- 1 transition probability
 2- 2 transition probability
 2- 3 transition probability
 2- 4 transition probability
 3- 1 transition probability
 3- 2 transition probability
 3- 3 transition probability
 3- 4 transition probability
 4- 1 transition probability
 4- 2 transition probability
 4- 3 transition probability
 4- 4 transition probability
     0.000  1.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  ...
     0.300  0.7942  0.1571  0.0282  0.0205  0.0177  0.8968  0.0430  ...
     0.600  0.6182  0.2892  0.0529  0.0397  0.0325  0.8061  0.0787  ...
     0.900  0.4707  0.3897  0.0747  0.0648  0.0437  0.7358  0.1046  ...
     1.200  0.3592  0.4561  0.0896  0.0951  0.0517  0.6824  0.1261  ...
     1.500  0.2698  0.5042  0.1023  0.1237  0.0582  0.6402  0.1441  ...
     1.800  0.2119  0.5330  0.1102  0.1450  0.0625  0.6131  0.1561  ...
     2.100  0.1709  0.5461  0.1216  0.1614  0.0643  0.5935  0.1671  ...
     2.400  0.1437  0.5309  0.1437  0.1817  0.0648  0.5809  0.1764  ...
     2.700  0.1230  0.5195  0.1563  0.2011  0.0654  0.5741  0.1836  ...
.
.
.
```

Figure 6. Example file showing transition probability data in GEOEAS format.

or uncertain probability values lying between zero and one and summing to one, for example, (0.23, 0.34, 0.07, 0.36).

The GEOEAS format is also used for the output of 1-D transition probability data files produced by **GAMEAS** and **MCMOD**. For example, the example file excerpt shown in Figure 6 contains transition probabilities in the vertical (z)-direction computed by **GAMEAS**. To conform with the GEOEAS format, the transition probability files are generated as described in Table 2. Again, programs such as **GRAFXX** and **MCMOD** will read the transition probability values in "free format," so the user could provide data in other columnar forms. In all cases, the GEOEAS header is expected.

Binary Grid

A binary format is used to compactly store arrays of values for the 3-D Markov chain models generated by **MCMOD** and the 3-D conditional simulations generated by **TSIM**. Although the binary files do not provide direct access, there is usually no need to directly examine the

Line(s)	Description
1	proportions of the K categories
2	K^2+1, the number of data columns, which equals $4^2+1=17$ in the example
3	text describing the "lag"
4 to $(4+K^2)$	text labeling the category transitions, i.e., the "jk" in $t_{jk}(h)$.
$(5+K^2)$ to END	the lag and the transition probability values, cycling on k then j.

Table 2. Description of GEOEAS format for storing 1-D transition probability data.

Line	Description
1	the number of dimensions in the array
2	the sizes of each dimension
3	the array values, stored in one continuous stream

Table 3. Description of binary grid format.

contents of these files. The binary grid files are formatted as described in Table 3 whether values are integer or real. For example, the array values for a $2 \times 3 \times 4$ $(x \times y \times z) = 24$ node conditional simulation file generated by **TSIM** would consist of a continuous stream of 24 integers (each representing the category number) cycling in order of x, y, z. Such a file would appear as (if the binary were converted to text):

3

2 3 4

$\underbrace{1\ 1\ 1\ 3\ 3\ 4\ 4\ 2\ 2\ 2\ 2\ 3\ 3\ 3\ 3\ 1\ 1\ 1\ 2\ 2\ 4\ 4\ 4\ 4}_{\text{24 values}}$

A 3-D Markov chain model $\mathbf{T}(h_x, h_y, h_z)$ generated by **MCMOD** consists of a *five* -dimensional array of `real*4` (4 byte) values cycling on x, y, z, j, k. These format details do not need to be known to run the T-PROGS codes; they are given for informational purposes. However, the following details should be noted for future reference:

- Using the binary grid option, the output from **TSIM** consists of 1-byte integer values, which may range over [-128,127].
- A *negative* conditional simulation value indicates a grid block wherein at least one conditioning datum is present. For example, a negative simulation value (-k) signifies that a datum indicating category k is located within the grid block. A positive simulation value of (+k), on the other hand, signifies that no data were present within the grid block, and that category k was generated by the conditional simulation process.

ASCII Grid

The conditional simulation output files can also be generated in ASCII format to facilitate

portability. The ASCII format is identical to the binary grid, except that the simulation array values are written out with one value per line.

4 GAMEAS

The program **GAMEAS** calculates bivariate (two-point) spatial statistics such as the (cross-) variogram, (cross-) covariance, transition probability, or joint probability. **GAMEAS** was modified from the GSLIB program **GAMV3** (Deutsch and Journel, 1992) to permit computation of transition and joint probabilities and to produce output in the GEOEAS format. Before running **GAMEAS**, the user must:

1. Prepare a data file in GEOEAS format as previously described in the data formats section.
2. Set up a parameter file.
3. Check the array dimension settings in the "include" file called **gameas.inc**.

Parameter File

Figure 7 shows an example parameter file for calculating vertical-direction transition probabilities using **GAMEAS**. The input format preserves conventions found in **GAMV3** (Deutsch and Journel, 1992) as described in Table 4.

```
START OF PARAMETERS
data.eas                                /input file
1 2 3                                   /x,y,z columns
4 4 5 6 7                               /nvar, var1,2,3,... columns
-1. 2.                                  /vmin, vmax
datatpz.eas                             /output file
41                                      /# lags
0.3000                                  /lag spacing
0.1500                                  /lag tolerance
1                                       /ndir
0.0 90. 0.25 -90.0 22.50 0.25           /az,daz,azbw;dip,..,..
16                                      /# of bivariate statistics
1 1  11                                 /j,k, 11=tp
1 2  11
1 3  11
1 4  11
2 1  11
2 2  11
2 3  11
2 4  11
3 1  11
3 2  11
3 3  11
3 4  11
4 1  11
4 2  11
4 3  11
4 4  11
```

Figure 7. Example parameter file for GAMEAS.

Line	Description
1	a dummy line of text which keys the beginning of the file with "STAR"
2	input data file name [format `char*40`]
3	columns numbers in data file containing x, y, and z locations of data
4	# of variables (categories), followed by column #'s in data file containing those variables
5	minimum "vmin" and maximum "vmax" values used to screen extreme-valued data
6	output file name [format `char*40`] for bivariate spatial statistics, e.g., **datatpz.eas**
7	number of lags for which bivariate spatial statistics will be calculated
8	lag spacing
9	lag tolerance ($\pm$distance allowance used for defining data pairs)
10	loop of "ndir" directions (suggest keeping ndir=1)
11	azimuthal direction, tolerance, and bandwidth; dip direction, tolerance, and bandwidth
12	number of (cross-) correlations, which will be $K \times K$ to obtain all the entries in $\mathbf{T}(h)$
13 to END	tail variable, head variable, index for type of bivariate statistic

Table 4. Description of parameters for GAMEAS.

Implementation Notes

- Twelve types of bivariate statistics can be calculated, with 1 through 10 described in detail in GSLIB by Deutsch and Journel (1992, p. 40-42):

  1 = traditional variogram
  2 = traditional cross-variogram
  3 = non-ergodic covariance
  4 = non-ergodic correlogram
  5 = general relative variogram
  6 = pairwise relative variogram
  7 = variogram of logarithms
  8 = power variogram ($w = \frac{1}{2}$): rodogram
  9 = power variogram ($w = 1$): madogram
  10 = indicator variogram
  11 = transition probability $\frac{E\{V_j(x)V_k(x+h)\}}{E\{V_j(x)\}}$; for data defined as indicator variables, $V_j(x) = I_j(x)$
  12 = joint probability $E\{V_j(x)V_k(x+h)\}$

- As depicted in Figure 8, the azimuth angle is a clockwise rotation of the x-y plane, and the dip angle is a counter-clockwise rotation of the y-z plane. Figure 9 shows how the lag spacing, lag tolerance, angle tolerance, and bandwidth parameters are defined.

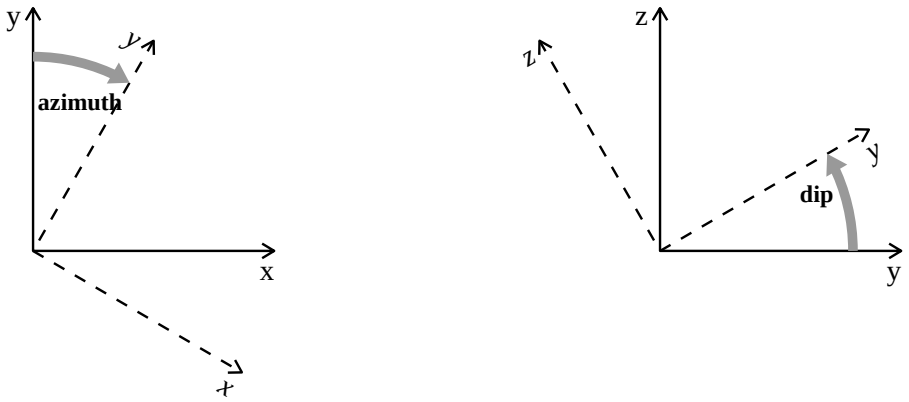


Figure 8. Azimuth and dip angles.

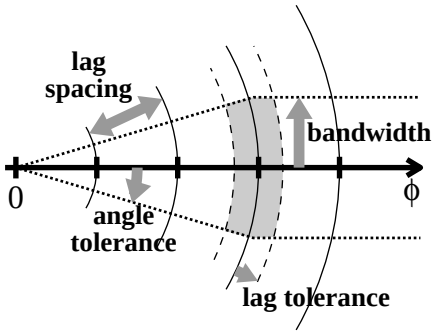


Figure 9. Lag spacing, lag tolerance, angle (azimuth or dip) tolerance, and bandwidth parameters.

Include File

The file **gameas.inc** sets dimensions of arrays in **gameas.f**. If not familiar with the dimension settings, the user should check **gameas.inc**, reset dimensions (as appropriate to the application), and recompile **gameas.f**.

Output

The output from **GAMEAS** consists of a GEOEAS-format file such as the file of calculated vertical transition probabilities shown in Figure 6. Note that a large record length will be produced because each lag contains the $K \times K$ entries needed to describe the full transition probability matrix.

GAMEAS will also produce a debugging file called **gameas.dbg** that contains diagnostic information about the computed spatial statistics, as given also by **GAMV3** of GSLIB (Deutsch and Journel, 1992, p. 53-60). This information includes lag number, mean lag distance, number of pairs, and mean values for tail and head variables.

5 GRAFXX

GRAFXX plots a *matrix* of graphs, such as 1-D transition probability, cross-variogram, cross-covariance, or cross-correlation matrix values as a function of lag. After calculating one-dimensional transition probabilities using **GAMEAS**, it is recommended to graph the matrix of measured transition probabilities using **GRAFXX** before embarking on the development of a spatial variability model. The graphs are useful for assessing data quality, interpreting juxtapositional relationships and trends, and preparing the implementation of the Markov chain modeling procedures described in Chapter 6. **GRAFXX** is also used later to compare measured transition probabilities with Markov chain models.

Before implementing **GRAFXX**, the user must

1. Generate one or more GEOEAS-format data files containing transition probability values as a function of lag.
2. Set up a parameter file.

Parameter File

Figure 10 shows an example parameter file for **GRAFXX** as described in Table 5. The resulting PostScript graphical output is shown in Figure 11. The number of lines will vary depending on the number of input data files (line 4) and the number of categories K (e.g., line 10).

Implementation Notes

- If the flag on line 20 equals 1 (instead of zero), then $4 \times 4 = 16$ (K^2) text lines will be expected below line 20 instead of the $8 = 4 \times 2$ $(K \times 2)$ lines as presented in the example of Figure 10.
- If any text lines are not needed, insert a blank line as presented in the example on lines 29 and 30.
- Use **GRAFXX** to plot other square matrices of graphs, such as the indicator cross-variogram or joint probability.

```
0                                          /symmetry check (1=sym)
0.20 0.20                                  /dx dy between plots (inches)
0                                          /line at zero? 1=yes
2                                          /number of input files
../../llnl/tp/llnl1195tpz.eas              /input file 1
-10  0.55 0 1.                            /file 1: marker, lw, dash, gray
../../llnl/tp/llnltpzm2.eas                /input file 2
0 1.5 0 0.                                 /file 2: marker, lw, dash, gray
../../manual/figs/llnl1195tpz2.ps          /output file
4                                          /number of categories
0. 6. 0.0 1.0                              /Xmin,Xmax,Ymin,Ymax
0 1                                        /# of x,y decimal places
5.4   0.9                                  /X,Y scales (units/inch)
1.0                                        /Data scale factor
0. 0. 0.                                   /axes color
3.  0.5                                    /X,Y label increments
3    5                                     /X,Y tics per label
Lag (m)                                    /X title
Transition Probability                     /Y title
0                                          /1= titles for each plot
debris fl                                  /X title variable 1
debris fl                                  /Y title variable 1
floodplain                                 /X title variable 2
floodplain                                 /Y title variable 2
levee                                      /X title variable 3
levee                                      /Y title variable 3
channel                                    /X title variable 4
channel                                    /Y title variable 4
                                           /title, line 1
                                           /title, line 2
1                                          /1=plot legend
7.5                                        /width of legend (inches)
Measured                                   /label for file 1 data
Markov Chain                               /label for file 2 data
```

Figure 10. Example parameter file for GRAFXX.

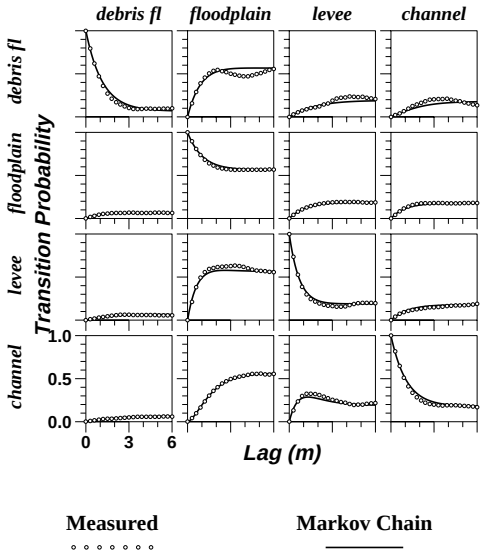


Figure 11. Postscript graphical output produced by GRAFXX using the example parameter file.

Line	Description
1	a flag: 0 = display full matrix, 1 = show only lower triangle (if symmetric)
2	X,Y spacing in inches between each of the graphs (matrix entries)
3	a flag: 1 indicates put a horizontal line the ordinate (Y-axis) value of zero
4	number of input files (data sets)
5	1st input file name: e.g., a data file of vertical transition probabilities
6	line attributes, file 1: marker, width (72/inch), dash, and gray (0=black, 1=white).
7	2nd input file name: e.g., a Markov chain model
8	line attributes, file 2:
9	encapsulated PostScript output file name: **tpz.eps**
10	number of categories
11	X minimum, X maximum, Y minimum, Y maximum values for graphs
12	number of decimal places in X, Y labels
13	X, Y scales in *units per inch*
14	data scale factor (multiplier)
15	axes gray level (0.0 = black, 1.0 = white)
16	X, Y label increments
17	X, Y tics per label
18	X axis title
19	Y axis title
20	flag: 0 = column-row (X-Y) titles; 1 = titles for each graph
21 to 28	column 1, row 1, ..., column 4, row 4 titles
29	title, line 1
30	title, line 2
31	flag: 1 = plot legend
32	width of legend in inches
33	label for file 1 data
34	label for file 2 data

Table 5. Description of parameters for GRAFXX.

Dash Code

The dash code (used in the line attributes for lines 6 and 8 in Table 5) specifies the type of dash used in drawing a line that connects data.

0 =	no dash
1 to 10 =	dash size proportionate to number

Marker Code

The marker code (used in the line attributes for lines 6 and 8 in Table 5) specifies the type of marker used to plot a data point.

Code	**Result**
0	line with no markers
negative	markers with no line
positive	markers and line
±1	cross
±2	diamond
±3	×
±4	box
±5	3-point star
±6	triangle
±7	5-point star
±8	pentagon
±9	6-point star
±10	circle
±11	sphere
±12	filled circle

If the marker code is zero (0), only a line connecting the data values is plotted. If the marker code is negative, say (-10), the data values are plotted as circles with no connecting lines. If the marker code is positive, say (+10), the data values are plotted as circles with connecting lines.

6 MCMOD

MCMOD provides several means for generating 1-D and 3-D Markov chain models of spatial variability. The Markov chain is an important theoretical model for cross-correlated categorical variables. It has shown remarkable applicability to many categorical geological data sets, particularly vertical stratigraphic successions. Three-dimensional Markov chain models are generated in **MCMOD** by interpolating models for each of the principal directions, say x, y, and z or stratigraphic strike, dip, and vertical (upward).

Before running MCMOD, the user must:

1. Have a rudimentary understanding of the transition probability and Markov chain models.
2. Set up a parameter file.
3. Check the array dimension settings in the **mcmod.inc** include file.
4. If using option 2 (see below), prepare a GEOEAS-format *transition probability* data file (as calculated from **GAMEAS**).

The resulting 3-D Markov chain file in binary grid format is used to prescribe the model of spatial variability for the conditional simulation program **TSIM** (Chapter 7).

Theory

Markov chain models applied to time series assume that the future depends on the present and not the past. For a one-dimensional spatial application, a Markov chain model assumes that an outcome at a specified location depends entirely on the nearest datum. A three-dimensional Markov chain model assumes that spatial variability in any one direction can be characterized by a one-dimensional Markov chain (Lin and Harbaugh, 1984; Politis, 1994). Although the Markov chain is defined very simply in theoretical and mathematical terms, it has shown remarkable applicability to characterization of spatial variability of facies (or hydrostratigraphic units) in alluvial and fluvial depositional systems (Carle and Fogg, 1996; Carle 1996; Carle and Fogg, 1997; Carle and others, 1998). Mathematically, it can be shown that the Markov chain consists of linear combinations of exponential structures, although non-exponential-looking "Gaussian" and "hole-effect" structures can be generated.

Matrix Exponential Form

Mathematically, a Markov chain model applied to one-dimensional categorical data in a direc-

tion ϕ assumes a *matrix exponential* form

$$\mathbf{T}(h_\phi) = \exp(\mathbf{R}_\phi h_\phi) \tag{6}$$

where h_ϕ denotes a lag in the direction ϕ, and $\mathbf{R}_\phi$ denotes a transition *rate* matrix

$$\mathbf{R}_\phi = \begin{bmatrix} r_{11,\phi} & \cdots & r_{1K,\phi} \\ \vdots & \ddots & \vdots \\ r_{K1,\phi} & \cdots & r_{KK,\phi} \end{bmatrix}$$

with entries $r_{jk,\phi}$ representing the rate of change from category j to category k (conditional to the presence of j) per unit length in the direction ϕ (Krumbein, 1968).

An eigenvalue analysis *must* be carried out in order to evaluate $\exp(\mathbf{R}_\phi h_\phi)$, because the matrix exponential is *not* computed merely by computing the exponential of the matrix entries, that is, $t_{jk,\phi}(h_\phi) \neq \exp(r_{jk,\phi} h_\phi)$. Letting $h = h_\phi$ and $\mathbf{R} = \mathbf{R}_\phi$ for notational simplification, $\exp(\mathbf{R}h)$ is either approximated by an infinite series or, better yet, exactly determined by

$$\exp(\mathbf{R}h) = \sum_{i=1}^{K} \exp(\lambda_i h)\, \mathbf{Z}_i$$

where λ_i and $\mathbf{Z}_i$ denote the eigenvalues and spectral component matrices, respectively, of $\mathbf{R}$. The mathematical details are given in Agterberg (1974) and Carle and Fogg (1997) and Carle and others (1998). One eigenvalue, say λ_i, is inherently zero and is associated with a spectral component matrix having the proportions along each column. Thus, for a four-category system, the continuous lag Markov chain model written out completely consists of

$$\begin{aligned} \exp(\mathbf{R}h) \;=\; & (1.0) \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_1 & p_2 & p_3 & p_4 \\ p_1 & p_2 & p_3 & p_4 \\ p_1 & p_2 & p_3 & p_4 \end{bmatrix} + \exp(\lambda_2 h) \begin{bmatrix} z_{11,2} & z_{12,2} & z_{13,2} & z_{14,2} \\ z_{21,2} & z_{22,2} & z_{23,2} & z_{24,2} \\ z_{31,2} & z_{32,2} & z_{33,2} & z_{34,2} \\ z_{41,2} & z_{42,2} & z_{43,2} & z_{44,2} \end{bmatrix} \\ & + \exp(\lambda_3 h) \begin{bmatrix} z_{11,3} & z_{12,3} & z_{13,3} & z_{14,3} \\ z_{21,3} & z_{22,3} & z_{23,3} & z_{24,3} \\ z_{31,3} & z_{32,3} & z_{33,3} & z_{34,3} \\ z_{41.3} & z_{42,3} & z_{43,3} & z_{44,3} \end{bmatrix} + \exp(\lambda_4 h) \begin{bmatrix} z_{11,4} & z_{12,4} & z_{13,4} & z_{14,4} \\ z_{21,4} & z_{22,4} & z_{23,4} & z_{24,4} \\ z_{31,4} & z_{32,4} & z_{33,4} & z_{34,4} \\ z_{41,4} & z_{42,4} & z_{43,4} & z_{44,4} \end{bmatrix} \end{aligned} \tag{7}$$

where the $z_{\alpha\beta,i}$ are coefficients of the spectral component matrices $\mathbf{Z}_i$ determined in the eigensystem analysis. Thus, the Markov chain model for each entry $t_{jk}(h)$ in $\mathbf{T}(h)$ consists of a *linear combination* of $K-1$ exponential structures added to the column category proportion. For example, in the four-category case given in (7)

$$t_{jk}(h) = p_k + z_{jk,2} \exp(\lambda_2 h) + z_{jk,3} \exp(\lambda_3 h) + z_{jk,4} \exp(\lambda_4 h)$$

Comparison to Discrete-Lag Form

Markov chain models are often formulated by the "discrete-lag" approach by successive multiplication of a transition probability matrix $\mathbf{T}(\Delta h_\phi)$ at discrete lag Δh_ϕ

$$\begin{aligned} \mathbf{T}(1\Delta h_\phi) &= \mathbf{I}\mathbf{T}(\Delta h_\phi) \\ \mathbf{T}(2\Delta h_\phi) &= \mathbf{T}(\Delta h_\phi)\mathbf{T}(\Delta h_\phi) \\ &\vdots \\ \mathbf{T}(n\Delta h_\phi) &= \mathbf{T}[(n-1)\Delta h_\phi]\mathbf{T}(\Delta h_\phi) \end{aligned} \tag{8}$$

where $\mathbf{T}(0) = \mathbf{I}$. The discrete-lag approach generates transition probabilities at only discrete lag multiples $1\Delta h_\phi$, $2\Delta h_\phi$, ..., $n\Delta h_\phi$. However, any discrete-lag Markov chain can be converted to a continuous-lag Markov chain by computing

$$\mathbf{R}_\phi = \frac{\ln\left[\mathbf{T}(\Delta h_\phi)\right]}{\Delta h_\phi} \tag{9}$$

which involves an eigensystem analysis (Agterberg, 1974; Carle, 1996; Carle and Fogg, 1997).

Eigensystem Analysis

MCMOD performs an eigensystem analysis because development of a continuous-lag Markov chain as a geostatistical model of spatial variability may require the following mathematical calculations:

- evaluate the *matrix exponential* form of Markov chain given by (6),
- evaluate the *matrix logarithm* of a transition probability matrix given by (9), and
- convert a discrete-lag Markov chain to a continuous-lag Markov chain by combining (6) and (9).

As shown above, (6) and (9) cannot be computed directly from the matrix entries. In either situation, the key step is to find the eigenvalues of $\mathbf{R}_\phi$ or $\mathbf{T}(h_\phi)$, which can be computed using codes for real general matrices as given by Smith and others (1976) or Press and others (1992).

For notational simplicity, let lag $h = h_\phi$ and $\mathbf{R} = \mathbf{R}_\phi$. A square $(K \times K)$ matrix such as $\mathbf{R}$ can be expressed in diagonal form with respect to its eigenvalues by

$$\mathbf{R} = \sum_{k=1}^{K} \lambda_k \mathbf{Z}_k \tag{10}$$

where the λ_k for $k = 1, ..., K$ denote the eigenvalues of $\mathbf{R}$, and $\mathbf{Z}_k$ denotes a spectral component matrix associated with each eigenvalue λ_k. The spectral component matrices $\mathbf{Z}_k$ can be determined directly from the eigenvalues and matrix $\mathbf{R}$ by

$$\mathbf{Z}_k = \frac{\prod\limits_{m \neq k} (\lambda_m \mathbf{I} - \mathbf{R})}{\prod\limits_{m \neq k} (\lambda_m - \lambda_k)} \qquad k = 1, ..., K \tag{11}$$

where $\mathbf{I}$ denotes the identity matrix. The continuous-lag Markov chain (6) then can be computed from

$$\mathbf{T}(h) = \sum_{k=1}^{K} \exp(\lambda_k h)\, \mathbf{Z}_k \tag{12}$$

through application of Sylvester's theorem (Agterberg, 1974, p. 406-412). The value of one eigenvalue of $\mathbf{R}$ will be zero, and the remaining eigenvalues will be negative (to ensure the negative diagonal transition rates). Recognizing that (12) represents a canonical form of $\mathbf{T}(h)$, two useful conclusions can be drawn for the Markov chain model:

1. The eigenvalues $\theta_k(h)$ of $\mathbf{T}(h)$ relate to the eigenvalues λ_k of $\mathbf{R}$ by

$$\theta_k(h) = \exp(\lambda_k h) \quad \text{or} \quad \lambda_k = \frac{\ln \theta_k(h)}{h} \qquad \forall k = 1, ..., K \tag{13}$$

2. Both $\mathbf{R}$ and $\mathbf{T}(h)$ have identical spectral component matrices $\mathbf{Z}_k$.

As a result, if a Markov chain model is assumed, a transition probability matrix $\mathbf{T}(\Delta h)$ for a discrete lag Δh can be used to compute $\mathbf{R}$ by applying (13) to (10) to obtain

$$\mathbf{R} = \sum_{k=1}^{K} \frac{\ln \theta_k(\Delta h)}{\Delta h} \mathbf{Z}_k \tag{14}$$

where $\theta_k(\Delta h)$ and $\mathbf{Z}_k$ are the eigenvalues and spectral component matrices, respectively, corresponding to $\mathbf{T}(\Delta h)$. Application of (14) to (6) yields

$$\mathbf{T}(h) = \sum_{k=1}^{K} \theta_k(\Delta h)^{h/\Delta h} \mathbf{Z}_k \tag{15}$$

which represents a continuous-lag version of the more commonly used discrete-lag Markov chain model (8). The clear advantage of (15) over (8) is the continuous functional representation of the model, that is, the ability to calculate $\mathbf{T}(h)$ at any h, not just integer multiples of Δh. Expression (12) shows that a Markov chain model corresponds to a linear combination of exponential functions. Nonetheless, rather nonexponential looking structures can be obtained from a Markov chain model, as evident in some of the off-diagonal transition probabilities for the examples given.

Considering that one eigenvalue of $\mathbf{R}$, say λ_1, has a value of zero, the corresponding eigenvalue $\theta_1(h)$ of $\mathbf{T}(h)$ has a value of unity for all h. The entries of the spectral component matrix $\mathbf{Z}_1$ correspond to the proportion p_k of the column category such that

$$\mathbf{Z}_1 = \begin{bmatrix} p_1 & \cdots & p_K \\ \vdots & & \vdots \\ p_1 & \cdots & p_K \end{bmatrix}$$

Considering (12) and that the other eigenvalues $\lambda_2, ..., \lambda_K$ are negative such that $\lim_{h \to \infty} \exp(\lambda_k h) = 0$ for $k = 2, ..., K$, then $\mathbf{Z}_1$ establishes the sill of the Markov chain model as given by (5).

Properties

The transition rate matrix has some important theoretical properties useful in model development:

- The transition rate corresponds to the *slope* of the transition probability as it approaches lag zero
$$r_{jk,\phi} = \frac{\partial t(\mathbf{h} \rightarrow 0)}{\partial h_\phi} \tag{16}$$
- The diagonal entries are negative ($r_{jj,\phi} < 0$), and the off-diagonal entries are (usually) non-negative ($r_{jk,\phi} \geq 0 \quad \forall k \neq j$), which ensures that $0 \leq t_{jk}(h_\phi) \leq 1$.
- The diagonal entries $r_{jj,\phi}$ are related to $\overline{L}_{j,\phi}$, the mean length of category j in the direction ϕ, by
$$r_{jj,\phi} = -\frac{1}{\overline{L}_{j,\phi}} \tag{17}$$
For example, the mean "thickness" [mean length in the vertical (z) direction] of category j corresponds to $\overline{L}_{j,z}$, so that a diagonal transition rate $r_{jj,z}$ can be established by
$$r_{jj,z} = -\frac{1}{\overline{L}_{j,z}}$$
- The row sums must equal zero
$$\sum_{k=1}^{K} r_{jk,\phi} = 0 \qquad \forall j \tag{18}$$
such that the diagonal entry is equivalent to the negative of the sum of the off-diagonal row entries
$$r_{jj,\phi} = -\sum_{k \neq j}^{K} r_{jk,\phi} \qquad \forall j$$
which ensures that $\sum_{k \neq j}^{K} t_{jk}(h_\phi) = 1$ for all j, k according to probability law.
- The column sums must obey
$$\sum_{j=1}^{K} p_j r_{jk,\phi} = 0 \qquad \forall k \tag{19}$$
which ensures that the transition matrix converges on the specified proportions, $t_{jk}(h_\phi \rightarrow \infty) = p_k$, as expected for a stationary Markov chain.

Background Category

Probability law and knowledge of proportions can be exploited in **MCMOD** by specifying

one category as "background." Usually proportions are known *a priori*, such that the row and column summing constraints (18) and (19) can be applied to eliminate the need to specify one column and one row of transition rates. For example, in a four-category system, only $3 \times 3 = 9$ transition rates need to be established instead of $4 \times 4 = 16$.

Multidimensional Markov Chains

2-D or 3-D Markov chain models can be developed by assuming that spatial variability in any direction can be characterized by a 1-D Markov chain (Switzer, 1965; Lin and Harbaugh, 1984; Politis, 1994). Although this may seem like a tenuous theoretical leap, the assumption here is merely that Markov chains might characterize spatial variability not only in the vertical but in other stratigraphic directions such as dip or strike. In a typical geologic application, data coverage usually is inadequate to directly develop a 1-D Markov chain model for each of the infinity of directions. Alternatively, model development can focus on the principal directions, say the strike (x), dip (y) and vertical (z). Then 1-D Markov chain models for any direction can be interpolated from the principal direction models.[2]

Considering that the transition probability matrix $\mathbf{T}(h_\phi)$ for an arbitrary direction ϕ depends entirely on $\mathbf{R}_\phi$, the interpolation of Markov chain models can be accomplished by ellipsoidally interpolating entries in the transition rate matrices for the principal x, y and z directions by

$$|r_{jk,\phi}| = \sqrt{\left(\frac{h_x}{h_\phi} r_{jk,x}\right)^2 + \left(\frac{h_y}{h_\phi} r_{jk,y}\right)^2 + \left(\frac{h_z}{h_\phi} r_{jk,z}\right)^2} \qquad \forall j,k \neq \beta \qquad (20)$$

where β denotes the background category, h_x, h_y and h_z are the x, y and z direction components of $h_\phi = \sqrt{h_x^2 + h_y^2 + h_z^2}$. The remaining entries in $\mathbf{R}_\phi$ involving j or $k = \beta$ can be determined by applying (18) and (19). For the negative lag vector components, say h_{-x}, entries from the rate matrix $\mathbf{R}_{-x}$ corresponding to the opposite direction $-x$ are defined by

$$r_{jk,-x} = \left(\frac{p_k}{p_j}\right) r_{kj,x}$$

and used in (20) in place of entries for $\mathbf{R}_x$, in accordance with the backward Kolmogorov differential equation (Agterberg, 1974, p. 455-456).

The Determinant - A Measure of Statistical Closeness

The lateral extent of the 3-D Markov chain model output by **MCMOD** must be finite, with limits that consider statistical closeness. Kriging-based algorithms, which do not consider cross-correlations, easily rank statistical closeness by the magnitude of the variogram (or covariance) model or a prescribed search radius with anisotropy ratios. However, the ranking of a full cross-correlation matrix for multiple categories is not so straightforward.

[2] We make no claim that non-negative definiteness is guaranteed for the three-dimensional Markov chain models. However, 1-D non-negative definiteness is ensured for each of the principal direction ϕ models by maintaining real and non-positive eigenvalues for $\mathbf{R}_\phi$. Our experience has shown that the transition probability-based cokriging equations implemented in TPSIM, although singular, are solvable by singular value decomposition.

A ranking of statistical closeness is needed not only in **MCMOD**, but also for the search and simulated quenching algorithms in **TSIM** (next chapter). A generalized method is needed for the following reasons:

- Different categories will have different correlation lengths and anisotropy ratios.
- Experience has shown that users tend to undervalue lateral:vertical anisotropy ratios when manually specifying degree of spatial continuity.

As such, T-PROGS utilizes a mathematically-based measure of statistical closeness, the determinant.

The determinant of a transition probability matrix $\mathbf{T}(h)$, is the product of its eigenvalues $\theta_k(h)$

$$\det\left[\mathbf{T}(h)\right] = \prod_{k=1}^{K} \theta_k(h)$$

For a Markov chain model, $\theta_k(h) = \exp\left(\lambda_k h\right)$, where λ_k are the $k = 1, ..., K$ eigenvalues of the transition rate matrix $\mathbf{R}$. Considering that one λ_k is zero and others are negative (for the real part), then the non-zero eigenvalues of $\mathbf{R}$ obey $\lim_{h\to\infty} \theta_k(h) = 0$, such that

$$\lim_{h\to\infty} \det\left[\mathbf{T}(h)\right] = 0$$

At lag zero ($h = 0$) all eigenvalues of $\mathbf{T}(0)$ equal unity, such that

$$\det\left[\mathbf{T}(0)\right] = 1$$

Therefore, for any lag vector $\mathbf{h}$

$$\det\left[\mathbf{T}(0)\right] \geq \det\left[\mathbf{T}(\mathbf{h})\right] \geq \det\left[\mathbf{T}(\infty)\right]$$

and

$$1 \geq \det\left[\mathbf{T}(\mathbf{h})\right] \geq 0$$

which suggests that $\det\left[\mathbf{T}(\mathbf{h})\right]$ can be used to rank the statistical closeness of two locations separated by a lag vector $\mathbf{h}$. A determinant value near unity indicates an strong correlation between the two points, whereas as determinant value near zero indicates a lack of correlation.

In practice **MCMOD** and **TSIM** utilize the $(K-1)$ root of $\det\left[\mathbf{T}(\mathbf{h})\right]$ (the geometric mean of the eigenvalues not associated with the sill) instead of the actual determinant. This reduces dependency of the closeness statistic on the number of categories. Values of $\left(\det\left[\mathbf{T}(\mathbf{h})\right]\right)^{\frac{1}{K-1}}$ ranging between 0.1 and 0.01 are recommended as limits for the 3-D model.

Application

Development of a Markov chain model of spatial variability focuses on establishing the entries $r_{jk,\phi}$ in the transition rate matrix $\mathbf{R}_\phi$. In straightforward mathematical terms, this can be viewed as estimation of the slopes $\frac{\partial t_{jk}(\mathbf{h}\to 0)}{\partial h_\phi}$ (or tangents at the origin) for all entries $t_{jk}(h_\phi)$ in the transition probability matrix $\mathbf{T}\left(h_\phi\right)$.

Selecting the Background Category

As stated above, application of the background category concept eliminates the need to specify one row and column of transition rates. In general, the background category may be selected according to geologic interpretation as the category that fills in the space not occupied by other categories. For example, in a fluvial depositional system consisting of *lag*, *channel*, *levee*, and *flood plain* deposits, the *flood plain* facies would be a logical choice for background because it has the lowest energy of deposition and, therefore, fills in accommodation space not otherwise occupied by higher energy facies.

Choosing the Approach

MCMOD can generate one-dimensional Markov chains by either (1) direct quantitative means, (2) estimation of $\frac{\partial t_{jk}(\mathbf{h}\rightarrow 0)}{\partial h_\phi}$ corresponding to the slope of the transition probability as lag approaches zero, (3) direct fitting to data, or (4) interpretation of juxtapositional tendencies. As a result, five different modeling approaches can be implemented with **MCMOD**:

1. **Transition Rates** – Prescribe the actual transition rates.
2. **Discrete Lag** – Honor transition probability data for a particular (discrete) lag .
3. **Embedded Transition Probabilities –** Interpret transition rates relative to an embedded transition probability matrix.
4. **Embedded Transition Frequencies** – Interpret transition rates relative to an embedded transition frequency matrix.
5. **Independence** - Interpret transition rates relative to “independent” or “maximum entropy (disorder)” juxtapositional tendencies.

The choice of approach will depend on the particular application or style of interpretation. The fact that many approaches are available exemplifies the flexibility of the Markov chain as a model of spatial variability. Various modeling situations are given below, for which the most conducive approaches are recommended.

Sparse Data

Most practical data sets yield noisy looking transition probability (or indicator cross-variogram) measurements, particularly for the lateral directions. The traditional geostatistical model development approach of empirical curve-fitting can easily lead to overcomplicated structures and inconsistencies with mathematical and probability theory. Alternatively, the Markovian model soundly addresses mathematical and probability theory while offering an interpretive framework for defining model parameters. The assumption of a Markov chain may be viewed as a conceptual simplification, that the spatial variability depends on values at nearest locations (first-order stochastic). One can develop a Markov chain (first-order) model from parameters conducive to integration of geologic insight: proportions, mean length, and juxtapositional tendencies. Noisy data typically do not support a more complicated (higher-order) model, unless supported by ancillary or interpretive information.

Recommendation:

Embedded Transition Probabilities - Prescribe cross (off-diagonal) transition rates in terms of conditional probabilities of embedded occurrences. For example, "Given an embedded occurrence of clay, what is the probability that sand occurs directly above?" Prescribe auto (diagonal) transition rates by mean lengths.

Transition Rates - Infer the slope or fit (interpolate) the tangent line of transition probability curve as lag approaches zero as per (16); these slope values directly translate to transition rates. Use this approach in conjunction with the embedded transition probability approach (through examination of the debugging file) to infer whether the prescribed transition rates are geologically plausible.

No Data

The Markovian framework is particularly conducive to development of models of spatial variability from conceptual information and, thus, is well-suited to situations lacking any data at all. For example, one can use geologic information or other insights on facies proportions, mean lengths, and juxtapositional tendencies to establish a geologically plausible model of spatial variability.

Recommendation:

Embedded Transition Probabilities - Prescribe cross-transition rates by estimating conditional probabilities of embedded occurrences according to geologically plausible juxtapositional tendencies. Prescribe auto-transition rates by estimated mean lengths.

Abundant Data

Given abundant data, the measured transition probabilities may display Markovian properties and define a smooth curve (without scatter). This situation might occur for numerous, finely-spaced data, such as continuous logs obtained from multiple boreholes penetrating the same geologic system.

Recommendation:

Discrete Lag – Use transition probability data at one (discrete) lag to establish the model at all lags.

Transition Rates – Infer the slope of the transition probability as the lag approaches zero; the slope values directly translate to transition rates as indicated by (16).

Interpretation Relative to Statistical Independence or Maximum Entropy (Disorder)

A main motivation for performing statistical analysis of bedding successions has been to quantify interpretation of juxtapositional tendencies, to address questions such as, "Does siltstone tend to occur above sandstone (versus claystone or conglomerate)." A standard is needed for judging whether a juxtapositional tendency is greater or lesser than "random." This can be based on statistical "independence," for which the frequency of occurrence of a pairs of events depends on the product of the marginal frequencies of the two events. Theoretically, statistical independence is identical to the maximum entropy concept, wherein a spatial arrangement of a given number of categories exhibits a state of maximum disorder.

Recommendation:

Independence – Set cross-transition rates relative to the independent (maximum entropy) model. Set auto-transition rates according to mean lengths. Use this approach primarily to interpret whether the data or model exhibit significantly nonrandom juxtapositional tendencies.

Transition Frequency/Count Data

The raw data used in an embedded Markov chain analysis consists of transition counts, for example, the number of observations of siltstone occurring over sandstone. These values may be normalized by the sum of the entire matrix to obtain transition frequencies, or the row sum to obtain transition probabilities. Recall that transition frequencies, rather than transition probabilities, are used in the assessment of statistical independence and, thus, represent a more fundamental statistic.

Recommendation:
Transition Frequencies – Prescribe transition frequencies or counts for off-diagonal entries, mean lengths for diagonal entries.

Direct Quantitative Interpretation

Transition rates can be interpreted directly in terms of a conditional rate of change per unit length. The auto-transition rates are negative because the auto-transition probability at an infinitesimal lag is less than unity (the auto-transition probability at lag zero). Similarly, the transition rates to other categories are usually positive because the cross-transition probabilities at an infinitesimal lag are expected to be greater than zero (the cross-transition probability at lag zero). Specifically, $r_{jk,\phi}$ denotes, given an occurrence of j, the rate at which j transitions to k per unit length in a direction ϕ. For example, if j is very continuous in the direction ϕ (j possesses a very large mean length), $r_{jj,\phi}$ will be negative and very small in magnitude, and $r_{jk,\phi}$ for $k \neq j$ will be positive but smaller in magnitude than $r_{jj,\phi}$, particularly if k tends not to occur adjacent to j. The summing constraint (18) maintains adherence to probability law by prescribing that the auto-transition rates equal the negative of the sum of the cross-transition rates.

Recommendation:
Transition Rates – Directly prescribe transition rates in terms of conditional rate of change per unit length or by estimation of the slopes $\frac{\partial t_{jk}(\mathbf{h}\to 0)}{\partial h_\phi}$ (or tangents at the origin).

Description of Approaches

1. Transition Rates

The transition rates are the entries $r_{jk,\phi}$ in $\mathbf{R}_\phi$ of the equation describing a continuous-lag Markov chain (6). The transition rates can be interpreted as the slope of the tangent of the transition probability curve as the lag approaches zero, as indicated by (16). Thus, one approach to developing a transition rate matrix would be to estimate the slopes $\frac{\partial t_{jk}(\mathbf{h}\to 0)}{\partial h_\phi}$ indicated by transition probability data. For example, a 4-category (*debris flow, floodplain, levee, channel*) vertical transition rate matrix could be established as

$$\mathbf{R}_z = \begin{bmatrix} -0.87 & * & 0.10 & 0.066 \\ * & * & * & * \\ 0.030 & * & -1.23 & 0.12 \\ 0.039 & * & 0.79 & -0.82 \end{bmatrix} \text{m}^{-1} \tag{21}$$

by estimation of the slopes $\frac{\partial t_{jk}(\mathbf{h}\to 0)}{\partial h_\phi}$. Recall that row and column sums of $\mathbf{R}_\phi$ should obey (18) and (19), which can be achieved by employing the background category concept. The entries for the row and column involving category 2, the background category, need not be specified. Note that the diagonal entries are negative, and the off-diagonal entries are non-negative. To avoid negative or above-unity probabilities, these sign conventions are recommended! Figure 12 shows the Markov chain model resulting from this transition rate matrix.

2. Discrete-lag Approach using Transition Probability Data at a Particular Lag

MCMOD employs an eigensystem analysis of (9) to establish a transition rate matrix from transition probability data at a particular lag, where Δh_ϕ would be chosen within the range of correlation. For example, the vertical (z)-direction transition probability matrix

$$\mathbf{T}(\Delta h_z = 0.6 \text{ m}) = \begin{bmatrix} 0.6182 & 0.2892 & 0.0529 & 0.0397 \\ 0.0325 & 0.8061 & 0.0787 & 0.0826 \\ 0.0192 & 0.3817 & 0.5258 & 0.0727 \\ 0.0168 & 0.0995 & 0.2359 & 0.6478 \end{bmatrix}$$

was used to compute the transition rates in (21) by (9) to obtain the model shown in Figure 12.

3. Transition Probabilities of an Embedded Markov Chain Analysis

An embedded Markov chain analysis evaluates the conditional probabilities of discrete occurrences of geologic units occurring adjacent to others in a particular direction (see Figure 2). For example, the *embedded* transition probabilities $\pi_{jk,z}$ in the vertical (z) direction are defined as

$$\pi_{43,z} = \Pr\{levee \text{ occurs above} \mid channel \text{ occurs below}\}$$

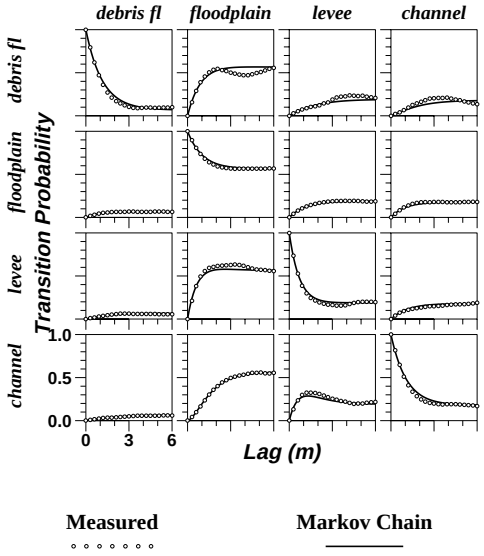


Figure 12. Matrix of transition probability data fit by Markov chain using discrete-lag approach at 0.6 m lag.

Consequently, an embedded transition probability matrix $\mathbf{\Pi}_z$ could be constructed as

$$\mathbf{\Pi}_z = \begin{bmatrix} - & 0.803 & 0.124 & 0.073 \\ 0.176 & - & 0.390 & 0.434 \\ 0.026 & 0.846 & - & 0.128 \\ 0.045 & 0.058 & 0.896 & - \end{bmatrix} \tag{22}$$

Note that auto (diagonal)-transitions are considered unobservable, thus, the diagonal entries are absent. From an interpretive standpoint, note in (22) that $\pi_{43,z} >> \pi_{41,z}$ and $\pi_{43,z} >> \pi_{42,z}$, which indicates that *levee* tends to occur above *channel*.

With the additional information of mean length $\overline{L}_{j,z}$, the entries of an embedded transition probability matrix can be translated into entries in a transition rate matrix by

$$r_{jk,z} = \frac{\pi_{jk,z}}{\overline{L}_{j,z}} \tag{23}$$

The transition rates in (21) are related to the embedded transition probabilities in (22) by (23). Thus, one approach to developing a transition rate matrix can be to (a) establish an embedded transition probability matrix from either data or geologic interpretation of juxtapositional tendencies, (b) establish mean lengths, and (c) convert the embedded transition probabilities to transition rates by (23).

Note that the off-diagonal entries $r_{jk,z}$ defined by (23) satisfy (17) and (18) because

$$\sum_{k=1}^{K} \pi_{jk,z} = 1$$

If a background category is assumed, *the row and column entries involving the background*

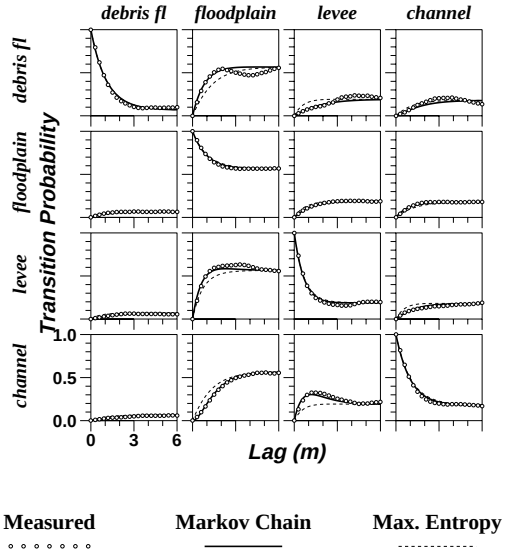


Figure 13. Markov chain model fit to matrix of transition probability measurements by using option 3 to adjust embedded transition probabilities. Markov chain model based on independent or "maximum entropy" juxtapositional tendencies shown by dashed line.

category do not need to be specified (set them to zero). A revised $\mathbf{R}_z$ was established from embedded transition probabilities and mean lengths by

$$\begin{bmatrix} \overline{L}_{1,z} = 1.15 & * & 0.12 & 0.075 \\ * & * & * & * \\ 0.025 & * & \overline{L}_{3,z} = 0.82 & 0.10 \\ 0.04 & * & 0.96 & \overline{L}_{4,z} = 1.24 \end{bmatrix} \tag{24}$$

where the diagonal entries are converted to transition rates by (17), the off-diagonal entries are converted to transition rates by (23), and category 2 is assumed as background (why row 2 and column 2 entries are set to any number). The resulting Markov chain model shown in Figure 13 fits the transition probability measurements slightly better than the initial model shown in Figure 12.

4. Transition Frequencies of an Embedded Markov Chain Analysis

An embedded Markov chain analysis may also be performed in terms of embedded transition frequencies $f_{jk,\phi}$ defined, for example in the vertical (z) direction, as

$$f_{43,z} = \Pr\{levee \text{ occurs above and } channel \text{ occurs below}\}$$

An embedded transition frequency matrix $\mathbf{F}_z$ could be formulated as

$$\mathbf{F}_z = \begin{bmatrix} (0.0841) & 0.0677 & 0.0101 & 0.0063 \\ 0.0672 & (0.3468) & 0.1264 & 0.1713 \\ 0.0085 & 0.2971 & (0.3395) & 0.0340 \\ 0.0085 & 0.0000 & 0.2031 & (0.2115) \end{bmatrix}$$

where the diagonal entries (in parenthesis) are the row and column sums corresponding to marginal frequencies $f_{j,z}$ of embedded occurrences of category j

$$f_{j,z} = \sum_{k \neq j}^{K} f_{jk,z} = \sum_{k \neq j}^{K} f_{kj,z} \tag{25}$$

With the additional information of mean length $\overline{L}_{j,z}$, the transition frequencies can be converted to transition rates by

$$r_{jk,z} = \frac{f_{jk,z}}{f_{j,z}\overline{L}_{j,z}} \qquad \forall k \neq j \tag{26}$$

Analogous to (24), **MCMOD** can establish the rate matrix from transition frequencies in the off-diagonal entries by

$$\begin{bmatrix} \overline{L}_{1,z} = 1.15 & * & 0.0101 & 0.0063 \\ * & \overline{L}_{1,z} = 2.27. & * & * \\ 0.0085 & * & \overline{L}_{3,z} = 0.82 & 0.0340 \\ 0.0085 & * & 0.2031 & \overline{L}_{4,z} = 1.24 \end{bmatrix}$$

Although off-diagonal entries for the background row and column do not need to be specified, *this approach requires a mean length for the background category* because the marginal frequencies depend on the proportions and mean lengths for all categories.

5. "Independent" or "Maximum Entropy" (Disorder) Transition Frequencies

The juxtapositional tendencies in a geologic system reflect some degree of order (or disorder) in the bedding sequences. The disorder of the juxtapositional tendencies in a particular direction, say ϕ, can be quantified by the entropy S_ϕ of bed-to-bed transition frequencies $f_{jk,\phi}$, the probabilities that one bed occurs next to another, by

$$S_\phi = -\sum_{j}\sum_{k} f_{jk,\phi} \ln f_{jk,\phi} \tag{27}$$

(Hattori, 1976). Consider the question, "What would the bedding look like for a maximally disordered system?" From this state of reference, one might be able to judge whether the observed bedding sequence exhibits nonrandom juxtapositional tendencies, that is, some preferential *order*.

In applying *embedded* Markov chain analyses, geologists have been interested in quantifying their interpretations of facies successions, particularly in the vertical direction. "Random" or independent transition frequencies for a succession involving four categories of beds would

obey

$$\begin{bmatrix} f_1f_1 & f_1f_2 & f_1f_3 & f_1f_4 \\ f_2f_1 & f_2f_2 & f_2f_3 & f_1f_4 \\ f_3f_1 & f_3f_2 & f_3f_3 & f_3f_4 \\ f_4f_1 & f_4f_2 & f_4f_3 & f_4f_4 \end{bmatrix} \tag{28}$$

where f_i denotes the "marginal" frequency that a bed of category i occurs in the succession (the direction ϕ is implied for notational simplicity). However, the problem is not so straightforward in practice. Usually, self-transitions (between two beds of the same category) are unobservable (particularly from borehole data), so that not only the diagonal transition frequencies but also the marginal frequencies cannot be directly evaluated. Off-diagonal transition *counts* can be estimated accurately, because transitions from one category to a *different* category are observable.

Assuming that self-transitions are unobservable, a more practical problem is posed. Instead, one analyzes observations of off-diagonal transition frequencies η_{jk}, which are defined as the transition counts divided by the sum total of all the off-diagonal transition counts

$$\begin{bmatrix} - & \eta_{12} & \eta_{13} & \eta_{14} \\ \eta_{21} & - & \eta_{23} & \eta_{24} \\ \eta_{31} & \eta_{32} & - & \eta_{34} \\ \eta_{41} & \eta_{42} & \eta_{43} & - \end{bmatrix}$$

Self-transitions are unobservable, so the diagonal entries are left blank. The objective of the independent transition frequencies (Turk, 1979; Turk, 1982; Johnson and Pattie, 1993) is to find the f_i's that satisfy

$$\frac{1}{T}\sum_{j\neq i}^{K} f_i f_j = \sum_{j\neq i}^{K} \eta_{ij} \qquad \forall i = 1,..,K \tag{29}$$

where $T = \sum_{i=1}^{K}\sum_{j\neq i}^{K} f_i f_j$. The non-linear system of equations (29) can be solved by the method of iterative proportion fitting (IPF) (Johnston and Pattie, 1993; Carle, 1997a). The resulting independent model will display the same marginal frequencies of *embedded* occurrences η_i, where

$$\eta_i = \sum_{j\neq i}^{K} \eta_{ij} \qquad \forall i = 1,..,K$$

The off-diagonal transition frequencies will be independent with respect to the estimated f_i's. Continuous lag Markov chain models can be linked to this model of independency by noting that η_i is proportional to proportions p_i divided by mean length $\overline{L}_i$ such that

$$\frac{p_i}{L_i} \propto \eta_i \qquad \forall i = 1,...,K$$

Thus, given a set of proportions and mean lengths, a continuous lag Markov chain with independent transition frequencies can be derived.

So a more specific question arises, "For the given proportions and mean lengths, what is the transition frequency matrix that maximizes entropy as defined in (27) ?" It so happens that the concept of statistical independence yields the same result as maximum entropy. The

resulting independent or maximum entropy transition frequency matrix $\mathbf{F}_z^{(S\max)}$ corresponding to the proportions and mean lengths of the previous examples is

$$\mathbf{F}_z^{(S\max)} = \begin{bmatrix} (0.0841) & 0.0377 & 0.0311 & 0.0152 \\ 0.0377 & (0.3671) & 0.2196 & 0.1075 \\ 0.0311 & 0.2196 & (0.3396) & 0.0888 \\ 0.0152 & 0.1075 & 0.0888 & (0.2115) \end{bmatrix}$$

where the diagonal entries represent the row/column totals or marginal frequencies. This matrix can be used as a basis for interpretation of juxtapositional tendencies relative to a maximally disordered bedding succession for the same proportions and mean lengths. For example, if an observed transition frequency $f_{43,z}$ is greater than $f_{43,z}^{(S\max)}$, then one might conclude that there is a statistical tendency for *levee* to occur above *channel*.

One can develop a transition rate matrix by (a) establishing proportions and mean lengths, (b) computing the maximum entropy transition frequency matrix $\mathbf{F}_z^{(S\max)}$ and corresponding transition rates, and (c) interpreting the off-diagonal transition rates relative to the maximum entropy transition rates. For example, applying (26), a maximum entropy transition rate $r_{jk,z}^{(S\max)}$ can be computed by

$$r_{jk,z}^{(S\max)} = \frac{f_{jk,z}^{(S\max)}}{\overline{L}_{j,z} \sum\limits_{k \neq j} f_{jk,z}^{(S\max)}}$$

The vertical transition rate matrix (21) can be formulated relative to the maximum entropy transition rates $r_{jk,\phi}^{(S\max)}$ as

$$\mathbf{R}_z^{(S\max)} = \begin{bmatrix} \frac{-1}{\overline{L}_{1,z}=1.15\text{ m}} & 1.80r_{12,z}^{(S\max)} & 0.32r_{13,z}^{(S\max)} & 0.41r_{14,z}^{(S\max)} \\ 1.78r_{21,z}^{(S\max)} & \frac{-1}{\overline{L}_{2,z}=2.27\text{ m}} & 0.58r_{23,z}^{(S\max)} & 1.59r_{24,z}^{(S\max)} \\ 0.27r_{31,z}^{(S\max)} & 1.35r_{32,z}^{(S\max)} & \frac{-1}{\overline{L}_{3,z}=0.82\text{ m}} & 0.45r_{34,z}^{(S\max)} \\ 0.56r_{41,z}^{(S\max)} & 0.00r_{42,z}^{(S\max)} & 2.29r_{43,z}^{(S\max)} & \frac{-1}{\overline{L}_{4,z}=1.24\text{ m}} \end{bmatrix} \quad (30)$$

By this approach off-diagonal transition rates are established by multiplying $r_{jk,z}^{(S\max)}$ by a coefficient: greater than unity indicates that the two categories tend to occur next to each other; lesser than unity indicates that the two categories tend not to occur next to each other.

The maximum entropy concept is particularly useful for interpreting juxtapositional tendencies of an existing Markov chain model. For example, one can take the proportions and mean lengths established by the existing model, say (21), then generate a Markov chain model with maximally disordered juxtapositional tendencies for those mean lengths and proportions (which establish the marginal frequencies of embedded occurrences). Comparison of transition probability measurements with the maximum disorder model can then be used to interpret whether the stratigraphy exhibits some degree of order in the juxtapositional tendencies. **MCMOD** can be used to establish the maximum entropy Markov chain model, shown in Figure 13,

for the given proportions and mean lengths in the previous examples by specifying for option 5

$$\begin{bmatrix} \overline{L}_{1,z} = 1.15 & 1.0 & 1.0 & 1.0 \\ 1.0 & \overline{L}_{2,z} = 2.27 & 1.0 & 1.0 \\ 1.0 & 1.0 & \overline{L}_{3,z} = 0.82 & 1.0 \\ 1.0 & 1.0 & 1.0 & \overline{L}_{4,z} = 1.24 \end{bmatrix}$$

where setting the off-diagonal coefficients equal to unity yields the maximum-entropy transition rates. *The mean length must be specified for all categories including the background category.*

The symmetric three-category case. In the special case of three categories with an assumption of symmetry, the maximum entropy system of equations reduces to 3 equations (the row/column summing constraints) and 3 unknowns (the symmetric off-diagonal transition frequencies). Therefore, the juxtapositional tendencies will always appear as "maximum entropy" for the given proportions and mean lengths. This situation illustrates the possible conflict between geological and statistical interpretations of maximum disorder. For example, consider a system with three facies: channel, levee, and a large proportion of floodplain deposits. A geologist would expect that levee deposits tend to occur laterally adjacent to the channel deposits and not haphazardly throughout the floodplain. Conversely, the location of the levee deposits could be viewed as tending to occur laterally adjacent to the floodplain deposits. If the levee deposits did in fact occur haphazardly throughout the floodplain, the effect would be to reduce the mean length of the floodplain deposits, i.e. the floodplain deposits would be less laterally extensive. Thus, in a geologic interpretation of the relative disorder of juxtapositional tendencies, one may also need to make comparisons of entropy allowing for one or more facies (particularly the background category) to vary in mean length.

Parameter File

The parameter file for **MCMOD** consists of four parts: (a) parameters common to the 3-D model, (b) parameters describing the x-direction model, (c) parameters describing the y-direction model, and (d) parameters describing the z-direction model. The formats describing each x, y, z-direction model are identical. Figure 14 shows an example parameter file for a four-category application using options 1 and 2. The parameters are described in further detail in Table 6. Figure 15 shows another example parameter file using options 3 and 4.

Line	Description
1	number of categories
2	proportions for each category (should add up to unity!)
3	background category (highly recommended, but can be set at zero if not used)
4	file name for debugging file
5	file name for 3-D Markov chain model produced in binary grid (.bgr) format
6	file name for determinant file used later in **TSIM** to determine statistical closeness
7	x, y, z lateral extent of 3-D model in terms of determinant
8	x, y, z lag spacing for 3-D model (same as for conditional simulation by **TSIM**)
9	x-direction output file name for 1-D Markov chain model
10	x-direction number of lags, lag spacing
11	x-direction modeling approach (option 1 in this example)
12	row 1 entries for x-direction transition rate matrix
13	row 2 entries for x-direction transition rate matrix
14	row 3 entries for x-direction transition rate matrix
15	row 4 entries for x-direction transition rate matrix
16	y-direction output file name for 1-D Markov chain model
17	y-direction number of lags, lag spacing
18	y-direction modeling approach (option 1 in this example)
19	row 1 entries for y-direction transition rate matrix
20	row 2 entries for y-direction transition rate matrix
21	row 3 entries for y-direction transition rate matrix
22	row 4 entries for y-direction transition rate matrix
23	z-direction output file name for 1-D Markov chain model
24	z-direction number of lags, lag spacing
25	z-direction modeling approach (option 2 in this example)
26	transition probability data file
27	lag number for developing Markov chains from transition probability data

Table 6. Description of parameter file shown in Figure 13.

```
4                                       /# of categories
0.066 0.565 0.190  0.179                /proportions
2                                       /background category
../llnl/tp/mcmod1_2.dbg                 /name of debugging file
../llnl/tp/tpxyz1_2.bgr                 /output file for 3-D model
../llnl/tp/det1_2.bgr                   /output file for determinant
0.05   0.05 0.05                        /determinant extent for 3-D model
3.0 10.0  0.30                          /dx,dy,dz for 3-D model
../llnl/tp/llnltpxm1_2.eas              /X-direction output file
200     1.                              /X-Direction: # lags, spacing
1                                       /option: 1=r,2=d,3=etp,4=etf,5=i
 -0.125  0.    -1.    -1.               /row 1 transition rates
  0.     0.     0.     0.               /row 2 transition rates
  0.0042 0.    -0.167 -1.               /row 3 transition rates
  0.004  0.     0.084 -0.100            /row 4 transition rates
../llnl/tp/llnltpym1_2.eas              /Y-direction output file
200     2.5                             /Y-Direction; # lags, spacing
1                                       /option: 1=r,2=d,3=etp,4=etf,5=i
 -0.042  0.     0.0036 0.0022           /row 1 transition rates
  0.0    0.     0.     0.               /row 2 transition rates
  0.0013 0.    -0.050  0.016            /row 3 transition rates
  0.0008 0.     0.017 -0.020            /row 4 transition rates
../llnl/tp/llnltpzm1_2.eas              /Z-direction output file
200   0.1                               /Z-Direction: # lags, spacing
2                                       /option: 1=r,2=d,3=etp,4=etf,5=i
../llnl/tp/llnl1195tpz.eas              /data file
2                                       /lag#
```

Figure 14. Example parameter file for MCMOD using options 1 and 2.

```
4                                          /# of categories
0.066 0.565 0.190  0.179                   /proportions
2                                          /background category
../llnl/tp/mcmod3_4.dbg                    /name of debugging file
../llnl/tp/tpxyz3_4.bgr                    /output file for 3-D model
../llnl/tp/det3_4.bgr                      /output file for determinant
0.05   0.05 0.05                           /determinant extent for 3-D model
3.0 10.0  0.30                             /dx,dy,dz for 3-D model
../llnl/tp/llnltpxm3_4.eas                 /X-direction output file
200     1.                                 /X-Direction: # lags, spacing
3                                          /option: 1=r,2=d,3=etp,4=etf,5=i
8.0     0.   -1.   -1.                     /row 1 embedded tp's & ml's
0.      0.    0.    0.                     /row 2      ''
0.025   0.   6.0   -1.                     /row 3      ''
0.040   0.   0.84   10.                    /row 4      ''
../llnl/tp/llnltpym3_4.eas                 /Y-direction output file
200     2.5                                /Y-Direction; # lags, spacing
4                                          /option: 1=r,2=d,3=etp,4=etf,5=i
24.     0.   -1.     -1.                   /row 1 embedded tf's & ml's
0.     62.81  0.      0.                   /row 2      ''
0.0095  0.   20.0    -1.                   /row 3      ''
0.0058  0.    0.1208 50.                   /row 4      ''
../llnl/tp/llnltpzm3_4.eas                 /Z-direction output file
200   0.1                                  /Z-Direction: # lags, spacing
3                                          /option: 1=r,2=d,3=etp,4=etf,5=i
1.15  0.    0.12  0.075                    /row 1 embedded tp's & ml's
0.    0.    0.    0.                       /row 2      ''
0.025 0.    0.82  0.10                     /row 3      ''
0.04  0.    0.96  1.24                     /row 4      ''
```

Figure 15. Example parameter file for MCMOD using options 3 and 4.

Implementation Notes

- **Line 3** – If the background category is not used (set at zero), a 3-D model will not be produced.
- **Line 6** – The determinant is the product of the eigenvalues. For a Markov chain, $\det[\mathbf{T}(0)] = 1$ and $\det[\mathbf{T}(\infty)] = 0$. In general, try values ranging from 0.01 to 0.1 – the smaller the number, the greater the lateral extent; the greater the lateral extent, the larger the array size for the 3-D transition probability model, which, as stated earlier, is a *five*-dimensional array. The actual array size will depend on the Markov chain model parameters (primarily mean length) and lag spacing (line 7).
- **Line 7** – Anticipate that **TSIM** will assume the lag spacing of the 3-D Markov chain model is the same as the grid spacing for generating the realizations. **MCMOD** runs fast, so re-running **MCMOD** with revised lag spacing is easily performed by editing line 7 as needed, without changing any other parameters. One should consider that reducing grid spacing in the conditional simulations will necessitate increased array size for the 3-D Markov chain model. The determinant limit specification takes care of this adjustment automatically.
- **1-D files** – The 1-D x, y and z-direction model files are produced independently of the 3-D Markov chain model. Thus, very small lag spacings can be prescribed for the 1-D models, which are useful for graphing the models as continuous functions.
- **Option 2** – If an option other than '2' is chosen in line 25 as given in Table 6, the input format would replace lines 26-27 with four (K) lines describing the parameters of each row of the transition rate matrix.
- **Options 3 through 5** – These options permit establishment of the transition rates by more interpretable methods than options 1 and 2. In each of these options, the *diagonal* transition rates will be established indirectly by prescribing values of *mean length*.

- **Background Category** – The background category concept eliminates the need to specify entries in the background category row and column, except for the diagonal (mean length) entry in options 4 and 5.
- **Symmetry** – One can enforce *symmetry* in opposing pairs of off-diagonal transition rates by setting one of the opposing entries equal to negative unity (-1.0). Thus, if all juxtapositional tendencies are assumed symmetric, one can focus on establishing transition rates for either the lower or upper off-diagonal entries; the opposing off-diagonal entries can all be set to -1.0.

Include File

The file **mcmod.inc** is used to set the dimensions of arrays used in **MCMOD**. If not familiar with the dimension settings, the user should check **mcmod.inc**, reset dimensions (if necessary), and recompile **mcmod.f(or)**.

Output

The output files of **MCMOD** are a 3-D Markov chain model (line 5), which consists of a 5-D array stored in the binary grid format, and a 3-D grid of determinant values (line 6). These two grid files are used by **TSIM** for conditional simulation.

The one-dimensional x, y, and z-direction Markov chain model output files for (lines 9, 16, and 23) are generated independently of the 3-D model to enable interpretation and comparison with transition probability measurements along principal directions. It is recommended to generate the one-dimensional models at small lag spacings (usually smaller than the grid spacing of the 3-D model) in order to adequately display the continuity of the Markov chain model.

Debugging File

After running **MCMOD**, one should check the debugging file because it provides useful diagnostic information on the principal-direction and 3-D Markov chain models as follows:

- options used
- actual transition rates
- transition rates in terms of the more interpretive frameworks of options 3 through 5.
- eigenvalue and spectral component matrices
- 3-D model size and dimensions

This information is particularly useful for interpretation of juxtapositional tendencies and for making adjustments to the model.

Model Adjustments

One might develop an initial model using options 1 or 2, then refine the model using the

debugging output to help initialize the application of the other more interpretive options.

Negative Off-Diagonal Transition Rates

It is important to check for *negative* off-diagonal transition rates, for which the debugging file will include a warning message. Negative transition rates are likely to occur in a background row or column as a result of a strongly positive juxtapositional tendency at another entry. Although negative transition rates will surely lead to negative off-diagonal transition probabilities at very small lags, it is possible that the off-diagonal transition probabilities at the discretization of the 3-D model (and realizations) will remain positive. Thus, negative transition rates may be acceptable in some situations; however, this presents a problem for the 3-D Markov chain model interpolation scheme unless, the same entries are negative for all principal directions or the negative rates occur only in the background row or column.

7 TSIM

Once the 3-D transition probability model and determinant grid files are established by **MCMOD**, it is a relatively simple to generate a conditional simulation using **TSIM**.

Data in the GEOEAS format, such as the data file originally used to calculate transition probabilities in **GAMEAS**, may "condition" the simulations. Variations in azimuthal and dip directions of anisotropy can be incorporated by providing *a priori* grids of azimuth and dip angles.

The primary advantage of **TSIM** over other geostatistical algorithms is the incorporation of *all* of the bivariate statistics (cross-correlations) in the simulation process, which enables reproduction of juxtapositional relationships including asymmetric patterns such as fining-upward tendencies (Carle and Fogg, 1998).

Before running **TSIM**, the user must

1. Generate 3-D transition probability model and measures of closeness (e.g., determinant), as produced by **MCMOD**.
2. Set up a parameter file.
3. Check array dimension settings in **tsim.inc**.

Theory

TSIM generates conditional (or unconditional) simulations through a two-step procedure of:

1. Generating an "initial configuration" using a cokriging-based version of the sequential indicator simulation (SIS) algorithm (Deutsch and Journel, 1992).
2. Iteratively improving the conditional simulation in terms of matching simulated and modeled transition probabilities by applying the simulated quenching (zero-temperature annealing) algorithm.

The two steps are mutually dependent because the SIS step alone will not yield stochastic simulations that adequately honor the model of spatial variability, and the quenching step will not succeed without a rudimentary initial configuration. Both the SIS and quenching steps may rely on the same Markov chain model of spatial variability and, thus, are conducive to implementation in succession.

Sequential Indicator Simulation

The initialization step utilizes the sequential indicator simulation (SIS) algorithm described by Deutsch and Journel (1992, p. 123-125, p. 148), except that a transition probability-based indicator cokriging estimate is used to approximate local conditional probabilities by

$$\Pr\{k \text{ occurs at } \mathbf{x}_0 \mid i_j(\mathbf{x}_\alpha);\, \alpha = 1, ..., N;\, j = 1, ..., K\} \approx \sum_{\alpha=1}^{N} \sum_{j=1}^{K} i_j(\mathbf{x}_\alpha) w_{jk,\alpha}$$

where N is the number of data, K is the number of categories, $w_{jk,\alpha}$ represent a weighting coefficient, and $i_j(\mathbf{x}_\alpha)$ represents the value of an indicator variable

$$i_j(\mathbf{x}_\alpha) = \begin{cases} 1, & \text{if category } j \text{ occurs at } \mathbf{x}_\alpha \\ 0, & \text{otherwise} \end{cases} \qquad j = 1, ..., K$$

The transition probability-based cokriging system of equations (Carle, 1996; Carle and Fogg, 1996) for computing the weighting coefficients is

$$\begin{bmatrix} \mathbf{T}(\mathbf{x}_1 - \mathbf{x}_1) & \cdots & \mathbf{T}(\mathbf{x}_N - \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \mathbf{T}(\mathbf{x}_1 - \mathbf{x}_N) & \cdots & \mathbf{T}(\mathbf{x}_N - \mathbf{x}_N) \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_N \end{bmatrix} = \begin{bmatrix} \mathbf{T}(\mathbf{x}_0 - \mathbf{x}_1) \\ \vdots \\ \mathbf{T}(\mathbf{x}_0 - \mathbf{x}_N) \end{bmatrix}$$

where

$$\mathbf{W}_\alpha = \begin{bmatrix} w_{11,\alpha} & \cdots & w_{1K,\alpha} \\ \vdots & \ddots & \vdots \\ w_{K1,\alpha} & \cdots & w_{KK,\alpha} \end{bmatrix}$$

Use of cokriging instead of the traditional indicator kriging approach improves consideration of spatial cross-correlations.

Simulated Quenching

Starting from a SIS-generated initial configuration, the simulated quenching step is implemented to improve agreement between measured and modeled transition probabilities. The quenching step attempts to solve the optimization problem of

$$\min \left\{ \mathbf{O} = \sum_{l=1}^{M} \sum_{j=1}^{K} \sum_{k=1}^{K} \left[t_{jk}(\mathbf{h}_l)_{SIM} - t_{jk}(\mathbf{h}_l)_{MOD} \right]^2 \right\}$$

where "$\mathbf{O}$" denotes an objective function, the $\mathbf{h}_l$ denote $l = 1, ..., M$ specified lag vectors, and "MEAS" and "MOD" distinguish measured and simulated (measured from the realization) transition probabilities, respectively (Aarts and Korst, 1989; Deutsch and Journel, 1992, p. 159-160; Deutsch and Cockerham, 1994; Carle, 1997b). The simulated quenching algorithm is implemented by repeatedly cycling through each nodal location of the conditional simulation and inquiring whether a change to another category will reduce $\mathbf{O}$; if so, the change is accepted. This iterative improvement procedure continues until $\mathbf{O}$ is minimized, or a limit on the number of iterations is reached. Conditioning is maintained by not allowing changes of categories at

conditioning locations. "Artifact discontinuities" (Deutsch and Cockerham, 1994) are avoided by generation of the initial configuration and including consideration for anisotropy and limiting the number of lags in formulation of the objective function (Carle, 1997b).

Application

The conditional simulation procedure is somewhat *ad hoc* because it employs approximations and localized optimization schemes, not exact mathematical procedures. In other words, the simulation procedure remains somewhat of an "art." Nonetheless, the art involved permits considerable flexibility and potential for generating useful simulation results.

In running **TSIM** two important parameters affect both execution time and the nature of the simulation result:

- number of data used in the cokriging equations (for the SIS step)
- determinant limit, which controls the number of quenching lags.

Execution time in the SIS step is roughly exponentially proportional to the number of data used in the cokriging equations. More cokriging data tend to produce more intricate heterogeneity patterns. Experience has shown that four to twelve data usually produce desirable results, with less data favored to reduce execution time for applications with very large grids and large numbers of categories. An even number of cokriging data is recommended because odd numbers (say three or five) may produce artifactual features as a result of a systematic asymmetry in the data configuration.

For the quenching step, execution time is roughly linearly proportional to the number of quenching lags. More quenching lags does not necessarily yield better results. Artifactual results, particularly near conditioning and edge locations, can be caused by over-emphasis on fitting the simulated spatial variability to the model at large lags (Carle, 1997b). In general, it is recommended to choose a determinant limit that ensures coverage by quenching lags in all principal directions [e.g. lags with grid spacings of (1,0,0), (0,1,0), and (0,0,1)] and a reasonable number of non-major directions [e.g., (1,1,0), (1,0,1), etc.], without an excess of redundancy [e.g., (1,0,0),(2,0,0),...,(10,0,0)]. Exact specifications for determining a suitable determinant limit cannot be given because this depends on the particular transition probability model and grid spacing. In practice, a reasonable determinant limit can be chosen by mapping the determinant grid. Otherwise, **TSIM** contains an internal check. If the determinant limit is too large to encompass the smallest lag in a principal direction, say lag (1,0,0) for the x direction, **TSIM** will automatically lower the determinant. Thus, if too high a determinant limit is entered, say 0.99, **TSIM** will determine the largest acceptable determinant limit – a default value. In practice, one might start quenching with the default value, then compare quenching results for successively lower determinant limits.

Parameter File

An example parameter file for **TSIM** is shown in Figure 16, with parameters described in Table 7.

```
4                                       /number of categories
0.066 0.565 0.190 0.179                 /proportions
../llnl/sim/simxyz.bgr                  /output file
1                                       /output format: 1=binary, 2=ascii
1                                       /debugging level
tpsim.dbg                               /debugging file
4175                                    /seed
1                                       /number of simulations
1966.3      -60      3.                 /xcenter, nx+, xsiz
3023.5      -30     10.0                /ycenter, ny+, ysiz
142.07      -60      0.3                /zcenter, nz+, zsiz
1  4                                    /ndmin, ndmax
1                                       /ibasis:0=cov,1=tp
0.001                                   /wratio
../llnl/tp/tpxyz.bgr                    /trans. prob. model file
../llnl/tp/det.bgr                      /determinant file
../llnl/data/llnl1195.eas               /input data file
-50. -50.                               /azimuths: coord, true
1.5 1.5                                 /dip:  coord, true
junkaz.bgr                              /azimuth int*1 file
junkdip.bgr                             /dip int*1 file
 4    0.00001  -1                       /maxit; tol; -1=weight,1=lag1
0.4                                     /quenching determinant limit
```

Figure 16. Example parameter file for TPSIM.

Line	Description
1	number of categories
2	proportions
3	output file for grid
4	output format: 1=binary, 2=ASCII
5	debugging level: higher yields more information
6	debugging file name
7	seed for random number generator
8	number of simulations
9	x minimum; number of nodes in x direction; x node size
10	y minimum; number of nodes in y direction; y node size
11	z minimum; number of nodes in z direction; z node size
12	minimum and maximum number of data points used for cokriging estimates
13	basis function for cokriging estimates: 0=covariance; 1=transition probability
14	value for defining singularities in singular value decomposition (0.001 works fine)
15	transition probability model file name (generated by **MCMOD**)
16	determinant file name (generated by **MCMOD**)
17	input data file name (GEOEAS format)
18	fixed azimuths: coordinate system; stratigraphic
19	fixed dips: coordinate system; stratigraphic
20	specified azimuth direction file
21	specified dip direction file
22	quenching parameters: max# of iterations; tolerance; 1= closest lags only, -1=weight
23	determinant value prescribing spatial limit of quenching lags

Table 7. Description of parameters for TPSIM.

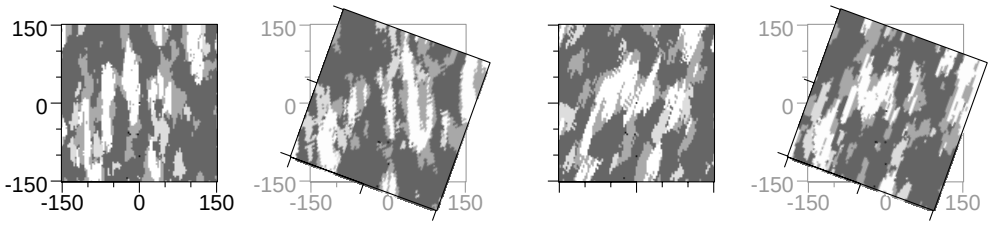
Implementation Notes

- **Lines 3 and 8**. Line 3 constitutes the root of the output file name. If multiple realizations are requested by setting line 8 greater than one, then an integer signifying the realization number will be appended to root of the output file name. For example, if line 3 = "simulation.bgr" and line 8 = "3", then the output realizations will be named **simulation.bgr1**, **simulation.bgr2**, and **simulation.bgr3**. A maximum of 100 realizations can be requested.
- **Lines 9-11**. The coordinate system can also be specified relative to the *center* of the simulated volume, rather than relative to the coordinate minima. The centered approach facilitates model development in many applications because the center of a model is usually easier to establish than the corners, particularly if the model coordinate system is rotated relative to the data coordinate system. The centered approach can be implemented by specifying *negative* numbers for the number of nodes relative to the center.

Line	Description
9	x center; *negative* number of nodes in $\pm x$ direction; x node size
10	y center; *negative* number of nodes in $\pm y$ direction; y node sizeentry
11	z center; *negative* number of nodes in $\pm z$ direction; z node size

Thus, if "-20" is entered for the number of nodes in $\pm x$ direction, the number of nodes in the x-direction of the simulation will be $2 \times 20 + 1 = 41$, with x node 21 centered at the "x center."

- **Line 13**. The cokriging equations are actually solved using a basis function approach (Carle, 1996). Selecting the covariance option implements simple cokriging, which will de-emphasize trends in proportions (nonstationarity of the mean) in the simulation results.
- **Line 14**. For more details about solving linear systems of equations with singular value decomposition, see Press and others (1992, p. 51-63).
- **Lines 18-19**. Set fixed values for coordinate system and stratigraphic azimuth and dip directions, for example: (a) coordinate azimuth $= 0°$, stratigraphic azimuth $= 0°$; (b) coordinate azimuth $= 20°$, stratigraphic azimuth $= 0°$; (c) coordinate azimuth $= 0°$, stratigraphic azimuth $= 20°$; (d) coordinate azimuth $= 20°$, stratigraphic azimuth $= 20°$.



(a) (b) (c) (d)

- **Lines 20-21**. Optional binary grid files of integer values containing local *azimuth* directions rounded to nearest degree and local *dip* directions rounded to nearest *tenth* of a degree. If not using these options, insert bogus file names. These grids must have identical dimen-

sions as the simulation. This option enables generation of realizations with variable dip and strike, as would result from structural features such as folds or geomorphic features such as meandering or a radial pattern of deposition.

- **Line 22**. These are quenching parameters. No more than five quenching iterations are usually necessary – too many iterations may produce unrealistic artifacts. If the number of iterations is set at zero, then simulated quenching will not be performed. If the number of iterations is *negative*, then simulated quenching will be performed on existing realizations as named by line 3. This feature is useful for examining different SIS and quenching schemes. The tolerance limit sets a criteria for terminating quenching based on the value of the objective function as normalized relative to its initial value. The last parameter enables the objective function to consider quenching lag vectors with components of no more than one nodal spacing (set $= 1$) or to weight quenching lags by the determinant value (set $= -1$).
- **Redundant data.** If more than one datum fall within a grid block, the simulation honors the first datum encountered in the data file.

Include File

The file **TSIM.inc** is used to dimension arrays of **TSIM**. If not familiar with the dimension settings, the user should check **TSIM.inc**, reset dimensions (if necessary), and recompile **TSIM.f(or)**.

IMPORTANT**!** The three main 3-D arrays of **TSIM** consist of integer values prescribing (a) the category at each node, (b) azimuthal direction angles rounded to nearest degree, and (c) dip direction angles rounded to nearest tenth of a degree. To conserve memory usage, disk space, and input/output time, these arrays have been declared in **TSIM.inc** as 1-byte variables instead of the default 4-byte type. The one-byte integers may range from -128 to 127. However, the `integer*1` variable type is not an ANSI FORTRAN standard, although some compilers (e.g., DEC and SGI) support it. In other cases (e.g., SUN), a `logical*1` variable type can be used to store 1-byte integer values. Therefore, the user may need to customize the variable declaration statement for the arrays `sim`, `iaz`, and `idip` in **TSIM.inc**. One can still use the standard `integer*4` or, alternatively, an `integer*2` declaration.

Also, because 1-byte integer or logical variables types are not standard FORTRAN, the binary (***.bgr**) files may not transfer across different computer systems.

Output

The output from **TSIM** can be produced in either ASCII or a compact binary formats. **TSIM** assigns negative values to any grid block with a category determined directly by conditioning data, which facilitates highlighting and understanding of the impact of conditioning data.

8 CHUNK

CHUNK displays conditional simulations produced by **TSIM** in color or grayscale 3-D perspective using the PostScript graphical language. **CHUNK** expects a 3-D array of integer values in the binary grid format, as produced by **TSIM**. Display of the simulation may be broken apart (exploded) into several sub-volumes (chunks), any number of which may be removed to reveal internal architecture. Grid blocks that occur on the edge of the simulation, which are prone to artifactual results, may be stripped away to reveal more representative surfaces.

Before running **CHUNK**, the user must

1. Create a 3-D array of integer values in the binary grid format, such as a realization produced by **TSIM**.
2. Set up a parameter file.
3. Check array dimension settings in **chunk.inc**.

Parameter File

Figure 17 shows an example parameter file for **CHUNK**, with parameters described in Table 8 and resulting graphical output shown in Figure 18.

Implementation Notes

- **Line 4** – The option for 2, 4 or 8-bit graphics permits different numbers of possible colors or grayshades. The possible number of grayshades is 2^{nbit} where $nbit$ is the number of bits selected. Therefore, 2-bit graphics permits 4 grayshades, 4-bit graphics permits 16 grayshades, and 8-bit graphics permits 256 grayshades. The possible number of colors is $2^{(3\times nbit)}$, which yields 64 for 2-bit, 4096 for 4-bit, and 16,777,216 for 8-bit. In general, 4-bit graphics is more than adequate. Grayshade maps are more likely to need 8-bit graphics. If the number of nodes is very large, 2-bit graphics might be useful for speeding up I/O, cutting down file size, or dealing with an architectural limit in PostScript mentioned below.
- **Line 5** – Color **rgb** (red, green, blue) values for background, axes, and title range from 0.0 to 1.0. A value of 1.0 denotes full intensity. For example, **rgb** values are for red=(1., 0., 0.),

```
0                                       /1=landscape
0.35 0.40                               /x,y plot translation (inches)
2.25 1.90                               /x,y legend translation (inches)
4         1                             /nbit:2,4 or 8; color?(=1)
1. 1. 1.    0. 0. 0.   0. 0. 0.        /bkgr, title, axes rgb
../../llnl/sim/simq.bgr                 /input binary grid file
2 2 2                                   /# x,y,z chunks
0. 150. 0.                              /x,y,z spacing betw. chunks
1                                       /# of chunks to crop
8                                       /chunk#'s to crop
../../manual/simq.ps                    /output .ps file
8                                       /# categories: cut, clr/gray
-4.0  13 13 8
-3.0  13 7 16
-2.0  13 5  1
-1.0  1  1  1
1.0   5   5  5
2.0   16  8  1
3.0   16 10 16
4.0   16 16 11
2  -2  -3                               /shading: XY,XZ,YZ ; +=light
-181.5 -305.   -18.15                   /Xmin,Ymin,Zmin
3. 10. 0.3                              /dX, dY, dZ
5 5 5 5 5 5                             /cropping: xl,xr,yl,yr,zl,zr
150. 150. 30.                           /X,Y,Z scales (units/inch)
150. 150. 10.                           /X,Y,Z label increments
0 0 0                                   /X,Y,Z decimal places
10 10 10                                /X,Y,Z tics per label
Strike (m)                              /X title
Dip  (m)                                /Y title
Vertical (m)                            /Z title
1.00                                    /Data scale factor
                                        / title, line 1
LLNL Aquifer System                     / title, line 2
0.95 -0.16  0.8 0.35                    /concat parameters
1                                       /legend? (1=yes)
4 4 1                                   /# of cats; nrow,ncol
1.6 2.0                                 /height & width
5                                       /category #
debris flow                             /label
6                                       /category #
floodplain                              /label
7                                       /category #
levee                                   /label
8                                       /category #
channel                                 /label
```

Figure 17. Example parameter file for CHUNK.

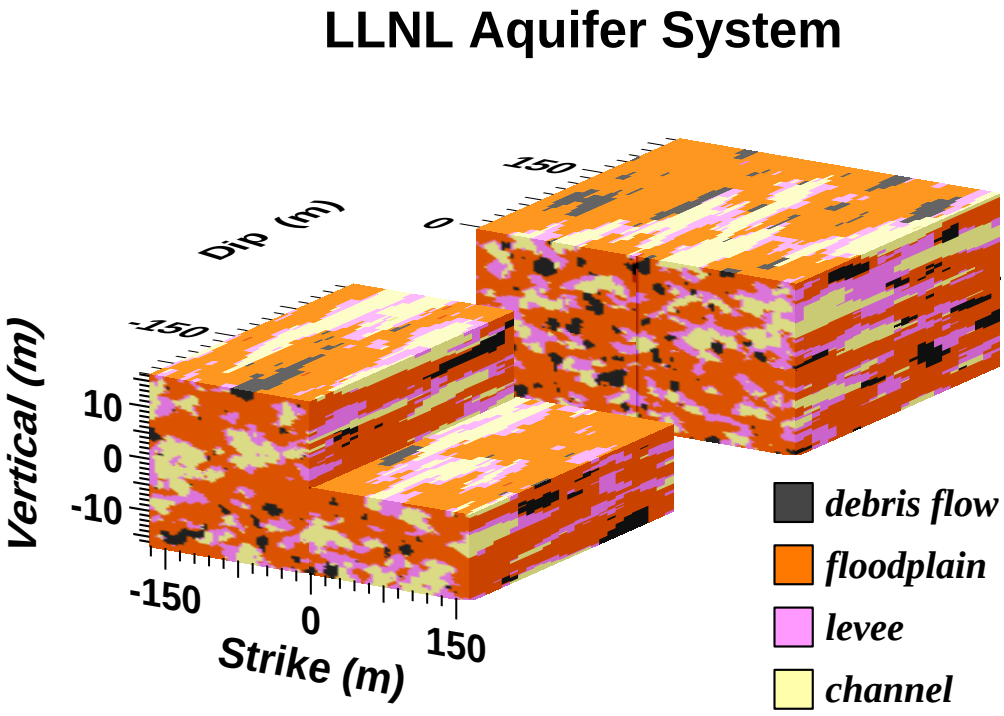


Figure 18. Example PostScript graphical output from CHUNK.

Line	Description
1	page orientation: 1=landscape, portrait otherwise
2	x, y plot translation (inches)
3	x, y legend translation (inches)
4	bits per pixel (2, 4, or 8); color=1, grayscale=0
5	red-greed-blue (rgb) color (0-1) for: background; title; and axes
6	binary grid input file name
7	# of chunks in x, y, and z direction
8	x, y, z spacing between chunks
9	# of chunks to crop (zero if none)
10	chunk #'s to crop (put dummy value if line nine is zero)
11	postscript output file name
13	# of categories (in ascending numerical order)=ncut
14 to 13+ncut	category and corresponding rgb values
14+ncut	shading: xy, xz, and yz plane (+ to lighten, - to darken)
15+ncut	x, y, and z minimum values for grid
16+ncut	x, y, and z grid spacing
17+ncut	# layers to crop in $x$min, $x$max, $y$min, $y$max, $z$min, $z$max directions
18+ncut	x, y, and z scales (units/inch)
19+ncut	x, y, and z label increment
20+ncut	x, y, and z # of decimal places in label
21+ncut	x, y, and z tics per label
22+ncut	x title
23+ncut	y title
24+ncut	z title
25+ncut	data scale factor
26+ncut	Title line #1 (leave blank line if not used)
27+ncut	Title line #2 (leave blank line if not used)
28+ncut	transformation parameters a, b, c, and d: $x' = \mathrm{a}x + \mathrm{c}y$; $y' = \mathrm{b}x + \mathrm{d}y$
29+ncut	legend? 0=no, 1=yes
30+ncut	# of categories (ncat) for legend; # rows, columns
31+ncut	height and width of legend
32+ncut	category #
33+ncut	category label
34+ncut to end	(repeat format of two previous lines)

Table 8. Description of parameters for CHUNK.

green=(0.,1.,0.), blue=(0.,0.,1.), yellow=(1.,1.,0.), cyan=(0.,1.,1.), and magenta=(1.,0.,1.). Grayshades can be produced by setting all **rgb** intensities equal with (0.,0.,0.)=black and (1.,1.,1.)=white.

- **Line 10** - Numbering of chunks is based on order in which they are drawn, from lower back left to upper front right, cycling by $+x$, $-y$, and $+z$.
- **Lines 14 to 13+ncut** – Grayshade and **rgb** intensities are specified by integer values of (1,2,3,4) for 2-bit graphics, (1,2,3,...,16) for 4-bit graphics, and (1,2,3,...,256) for 8-bit graphics. These values are then scaled to intensities ranging from 0.0 and 1.0. If a grayshade image is requested (second parameter in line 4 set to zero), only one value per line is needed (versus three per line for color).
- **Line 17+ncut** - Use these settings to crop off layers on the outer edges of a realization. Edge effects may mask more representative surfaces underneath.
- **Line 28+ncut** – The a, b, c, and d transformation parameters used to create the quasi 3-D perspective should maintain $\mathrm{a}^2 + \mathrm{c}^2 \leq 1$ and $\mathrm{b}^2 + \mathrm{d}^2 \leq 1$ to preserve x, y, z scaling relationships.
- **Architectural Limit** – Note that Postscript contains an architectural limit of 65535 elements in any array or string. What this means for **CHUNK** is that garbled results will occur on any face of a chunk that utilizes more than 262140 bits. One can get around this limitation by dividing the image into more chunks and/or using less bits per pixel.

Include File

The file **chunk.inc** is used to dimension arrays of **CHUNK**. The main array setting controls the size of the grid that can be visualized. If not familiar with the dimension settings, the user should check **chunk.inc**, reset dimensions (if necessary), and recompile **chunk.f(or)**.

IMPORTANT! **CHUNK** reads in the 3-D simulation array produced by **TSIM**. Recall that the variable type for the simulation array was set in **TSIM.inc**, so the corresponding variable type for the 3-D array specified in **chunk.inc** and read into **CHUNK** must be the same! Therefore, whether your system accepts `integer*1`, `logical*1`, or whatever, just make sure that both the `sim` array in **TSIM.inc** and the `ival1` array in **chunk.inc** are declared as the same variable type.

9 PostScript Basics

Regular PostScript (*.ps) Files

The output files from **GRAFXX** and **CHUNK** are produced in the PostScript (PS) graphical language. Inevitably, the user will need or want to modify the graphical output. In many cases, modifications can be made quite simply by editing the ASCII format PS file (***.ps**). Some basic properties which are useful for editing PS files are:

- All text strings are surrounded by parentheses. One can modify the text string by searching for (finding) it, then modifying the text string within the parentheses.
- On a line previous to a text string, a command such as `234.00 489.00 m` designates the X,Y page coordinates of the text string in 72nds of an inch. One can modify the coordinates to move the location of the text string.
- A command such as `72 144 translate` will translate subsequent graphics in the file by the X,Y page units specified.
- A command such as `2.0 2.0 scale` will scale subsequent graphics by factors of 2.0 in the X and Y page directions.
- The commands `findfont`, `scalefont`, and `setfont` *find*, *scale*, and *set* the current font.

Further details can be found in many reference texts such as the PostScript Language Reference Manual (Adobe Systems Incorporated, 1990).

Encapsulated PostScript (*.eps) Files

Although regular PS files can be directly interpreted by printers and on-screen viewers, they usually are not readily incorporated into word processing or slide presentation programs. The Encapsulated PostScript (EPS) format, however, is quite portable. The regular PS files (***.ps**) produced by **GRAFXX** and **CHUNK** can be converted to EPS files (***.eps**) by adding necessary header and footer information. This information can be added using the program **ps2eps.f(or).**

BoundingBox

The most important of the added EPS information is a “BoundingBox,” which specifies the *lower left* and *upper right* X,Y corner coordinates of the plot in units of 1/72 of an inch. The BoundingBox provides an opportunity to crop unneeded blank space surrounding the plot.

```
simq.ps                                  /postscript input file
simq.eps                                 /encapsulated postscript file
77 251 548 583                           /bounding box (in inches)
Steve Carle                              /creator
9/9/97                                   /date
3:00PM                                   /time
```

Figure 19. Example parameter file for PS2EPS.

An obvious way to define a suitable BoundingBox is to print out the PS file, then measure the displacements of the desired lower left and upper right corners relative to the lower left corner of the page.

An easier way to define a BoundingBox is to employ the Ghostview previewer, which displays PS or EPS files on screen. In the upper left corner, Ghostview tracks plot coordinates in units of 1/72 of an inch whenever the Mouse-driven crosshairs are located on the screen view of the plot. Thus, one can find suitable BoundingBox coordinates by moving the crosshairs to the desired lower left and upper right corners.

PS2EPS

The **ps2eps.f(or)** program is implemented in the same manner as all other T-PROGS programs. An example parameter file is shown in Figure 19, with parameters described in Table 9.

Line	Description
1	input PostScript file name
2	output Encapsulated PostScript file name
3	BoundingBox in 72nds of an inch: lower left X,Y; upper right X,Y
4	name of plot creator
5	date
6	time

Table 9. Description of parameters for PS2EPS.

10 Examples

The following examples are given to help reinforce procedures for implementing T-PROGS. For some, such working examples will provide the quickest route toward understanding how to obtain results with T-PROGS.

GSLIB's true.dat

The **true.dat** data set in GSLIB provides a widely accessible "exhaustive" data set that is useful for testing geostatistical techniques. In applying an indicator cross-variogram-based geostatistical approach, Goovaerts (1996) divided **true.dat** into four categories by cutoff values as follows:

category 1 = second highest 40%

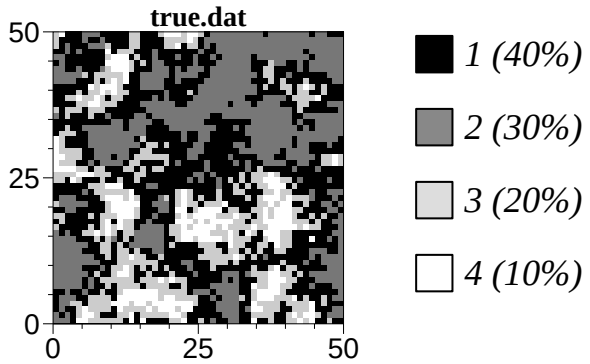
category 2 = highest 30%

category 3 = second lowest 20%

category 4 = lowest 10%

As such, the categories were treated as facies defining an intermediate scale of permeability heterogeneity.

The goal of this example is to generate a 3-D realization having an isotropic pattern of spatial variability similar to the 2-D **true.dat** data set.



Step 1 – Put data into GEOEAS format

```
 true.dat categorized by 0.15 0.44 and 2.12 cutoffs
 7
 x
 y
 z
 1=2nd highest 40%
 2=highest 30%
 3=2nd lowest 20%
 4=lowest 10%
     1.00000    1.00000  0. 0 1 0 0
     2.00000    1.00000  0. 0 1 0 0
     3.00000    1.00000  0. 0 1 0 0
     4.00000    1.00000  0. 1 0 0 0
     5.00000    1.00000  0. 0 0 0 1
     6.00000    1.00000  0. 0 0 0 1
     7.00000    1.00000  0. 0 0 1 0
     8.00000    1.00000  0. 0 0 1 0
     9.00000    1.00000  0. 0 0 1 0
    10.00000    1.00000  0. 0 0 1 0
     11.0000    1.00000  0. 0 0 1 0
     12.0000    1.00000  0. 0 0 1 0
.
.
.
etc.
```

Step 2 – Calculate isotropic transition probabilities using GAMEAS

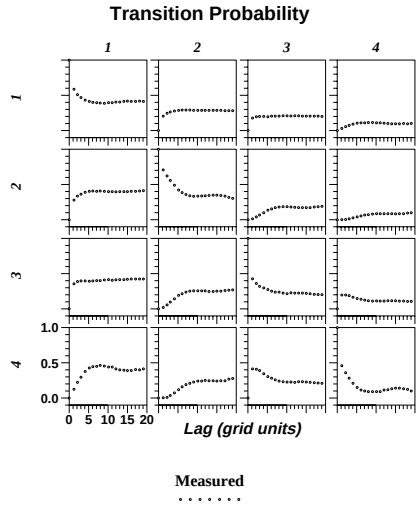
It is assumed that the pattern of spatial variability is isotropic and, thus, does not depend on direction. The omni-directional statistics are computed by employing a large azimuth bandwidth and an azimuth angle tolerance of slightly greater than 90°.

```
START OF PARAMETERS
../true/truecat.eas                     /input file
1 2 3                                   /x,y,z columns
4 4 5 6 7                               /nvar, var1,2,3,... columns
-1. 2.                                  /vmin, vmax
../true/tp.eas                          /output file
25                                      /# lags
1.                                      /lag spacing
0.5                                     /lag tolerance
1                                       /ndir
0.2 90.1 10000000.  0.0 22.50 0.25      /az,daz,azbw;dip,ddip,dipbw
16                                      /# of bivariate statistics
1 1  11                                 /j,k, 11=tp
1 2  11
1 3  11
1 4  11
2 1  11
2 2  11
2 3  11
2 4  11
3 1  11
3 2  11
3 3  11
3 4  11
4 1  11
4 2  11
4 3  11
4 4  11
```

Step 3 – Plot the transition probability data matrix using GRAFXX

Plot the transition probability data before proceeding to modeling spatial variability. The quality of the data will influence the choice of modeling approach.

```
0                                       /symmetry check (1=sym)
0.20 0.20                               /dx dy between plots (inches)
0                                       /line at zero? 1=yes
1                                       /number of input files
../../true/tp.eas                       /input file 1
-10 0.40  0 1.00                        /file 1: marker, lw, dash, gray
../../true/tp.ps                        /output file
4                                       /number of categories
0. 20. -0.1 1.0                         /Xmin,Xmax,Ymin,Ymax
0 1                                     /# of X,Y decimal places
15.   0.825                             /X,Y scales (units/inch)
1.0                                     /Data scale factor
0.0                                     /axes gray level
5.  0.5                                 /X,Y label increments
5    5                                  /X,Y tics per label
Lag (grid units)                        /X title
                                        /Y title
0                                       /1= titles for each plot
1                                       /X title variable 1
1                                       /Y title variable 1
2                                       /X title variable 2
2                                       /Y title variable 2
3                                       /X title variable 3
3                                       /Y title variable 3
4                                       /X title variable 4
4                                       /Y title variable 4
                                        /title, line 1
Transition Probability                  /title, line 2
1                                       /1=plot legend
6.5                                     /width of legend (inches)
Measured                                /name of variable 1
```



Step 4 – Model spatial variability using MCMOD

In this example, option 2 is initially employed using transition probability data for the second lag ($\Delta h = 2.2$) to model spatial variability in all principal directions.

```
4                                       /# of categories
0.4016 0.3004 0.2008 0.0972             /proportions
2                                       /background category
../true/mcmod.dbg                       /name of debugging file
../true/tpxy2.bgr                       /output file for 3-D model
../true/det2.bgr                        /output file for determinant
0.002   0.002 0.002                     /determinant extent for 3-D model
1. 1. 1.                                /dhx,dhy,dhz for 3-D model
../true/tpxm2.eas                       /X-direction output file
201     0.10                            /X-Direction: # lags, spacing
2                                       /1=r,2=d,3=etp,4=etf,5=i,6=p,7=f
../true/tp.eas                          /data file
2                                       /lag
../true/tpym2.eas                       /Y-direction output file
100     0.25                            /Y-Direction; # lags, spacing
2                                       /1=r,2=d,3=etp,4=etf,5=i,6=p,7=f
../true/tp.eas                          /data file
2                                       /lag
../true/tpzm2.eas                       /Z-direction output file
100   0.25                              /Z-Direction: # lags, spacing
2                                       /1=r,2=d,3=etp,4=etf,5=i,6=p,7=f
../true/tp.eas                          /data file
2                                       /lag
```

Step 5 – Examine debugging output from MCMOD

Warnings are given that several off-diagonal entries have negative transition rates (Don't panic!). Clearly, this categorization of **true.dat** data set yields very strong juxtapositional tendencies of $2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 2$. The negative transition rates are caused by very high transition probabilities for $1 \rightleftarrows 2$ and $3 \rightleftarrows 4$ at the second lag ($\Delta h = 2.2$).

The debugging output contains useful interpretive information. Juxtapositional tendencies appear fairly symmetric, because opposing off-diagonal entries of embedded transition *frequencies* are similar in magnitude. The symmetry is also evident in the coefficients of the transition rates with respect to independent transition frequencies, which also indicate the strong juxtapositional tendencies between $1 \rightleftarrows 2$ and $3 \rightleftarrows 4$ by coefficients much greater than unity. Transition rates of zero or less indicate that the categories are rarely, if ever, juxtaposed next to each other.

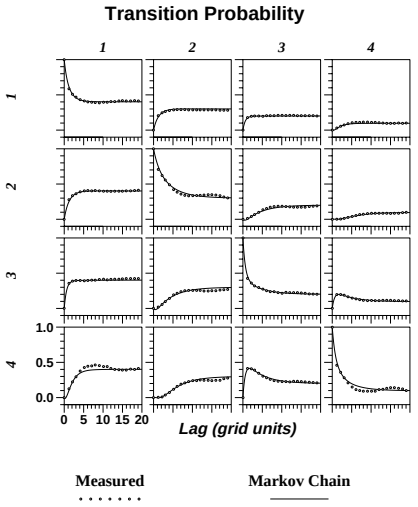
At the bottom of the debugging file, the lateral extent of the 3-D model is given in increments of nodal spacing. Make sure the lateral extent is large (or small) enough to encompass ranges of correlation for most or all of the categories. Recall that reducing the determinant limit increases the lateral extent of the 3-D model.

```
 MCMOD debugging file

Parameter file: mcmod.par
Number of categories: 4
Proportions:  0.4016  0.3004  0.2008  0.0972
Background category: 2

------- X-DIRECTION: -------
Method - option 2: trans. prob. at specified lag
1-D model output file: ../goofball/tpxm2.eas
WARNING: Off-diagonal Transition Rate   1  2        is too large for column
WARNING: Off-diagonal Transition Rate   1  4        is negative.
WARNING: Off-diagonal Transition Rate   2  1        is too large for row  2
WARNING: Off-diagonal Transition Rate   2  3        is negative.
WARNING: Off-diagonal Transition Rate   3  2        is negative.
WARNING: Off-diagonal Transition Rate   3  4        is too large for column
WARNING: Off-diagonal Transition Rate   4  1        is negative.
WARNING: Off-diagonal Transition Rate   4  3        is too large for row  4

Rate Matrix for X-Direction:
 -0.552953  0.251718  0.330448 -0.029212
  0.339745 -0.285394 -0.063882  0.009531
  0.653765 -0.086431 -0.972821  0.405488
 -0.115940  0.020553  0.841820 -0.746433

embedded transition probabilities:
  1.000000  0.455224  0.597606 -0.052830
  1.190444  1.000000 -0.223839  0.033395
  0.672030 -0.088846  1.000000  0.416816
 -0.155326  0.027535  1.127791  1.000000

embedded transition frequencies:
  0.385736  0.175596  0.230518 -0.020378
  0.177281  0.148920 -0.033334  0.004973
  0.228031 -0.030147  0.339317  0.141433
 -0.019575  0.003470  0.142133  0.126027
 entropy=   -1032.46

w.r.t. independent transition freqs:
(    1.8085)     2.2039      0.9605     -0.3085
     2.2250 (    3.5039)    -0.6139      0.3327
     0.9501     -0.5552 (    1.0279)     3.1416
    -0.2963      0.2322      3.1571 (    1.3397)

w.r.t. volumetric proportions:
(    1.8085)     0.9068      1.7809     -0.3252
     2.6738 (    3.5039)    -0.7799      0.2404
     1.3374     -0.2364 (    1.0279)     3.4272
    -0.3492      0.0828      5.0706 (    1.3397)

w.r.t. # of embedded occurrences
(    1.8085)     1.8777      1.0818     -0.2575
     2.6266 (    3.5039)    -0.5614      0.2255
     1.1510     -0.3942 (    1.0279)     2.1851
    -0.3519      0.1616      2.9048 (    1.3397)
.
.
.
 Constructing 3-D transition probability model
 # of lags in +x,+y,+z direction =  7  8  8
 total # of lags =  4335
```

Step 6 – Compare measured and modeled transition probabilities

The comparison is accomplished by slight modifications to the **GRAFXX** parameter file used in step 3.

```
0                                       /symmetry check (1=sym)
0.20 0.20                               /dx dy between plots (inches)
0                                       /line at zero? 1=yes
2                                       /number of input files
../../true/tp.eas                       /input file 1
-10 0.40  0 1.00                        /var 1: marker, lw, dash, gray
../../true/tpxm2.eas                    /input file 2
0 1.0  0 0.00                          /var 2: marker, lw, dash, gray
../../true/tpm.ps                       /output file
4                                       /number of categories
0. 20. -0.1 1.0                         /Xmin,Xmax,Ymin,Ymax
0 1                                     /# of X,Y decimal places
15.   0.825                             /X,Y scales (units/inch)
1.0                                     /Data scale factor
0.0                                     /axes gray level
5.  0.5                                 /X,Y label increments
5    5                                  /X,Y tics per label
Lag (grid units)                        /X title
                                        /Y title
0                                       /1= titles for each plot
1                                       /X title variable 1
1                                       /Y title variable 1
2                                       /X title variable 2
2                                       /Y title variable 2
3                                       /X title variable 3
3                                       /Y title variable 3
4                                       /X title variable 4
4                                       /Y title variable 4
                                        /title, line 2
Transition Probability                  /title, line 2
1                                       /1=plot legend
6.5                                     /width of legend (inches)
Measured                                /name of variable 1
Markov Chain                            /name of variable 2
```



Step 7 - Generate a 3-D realization using TSIM

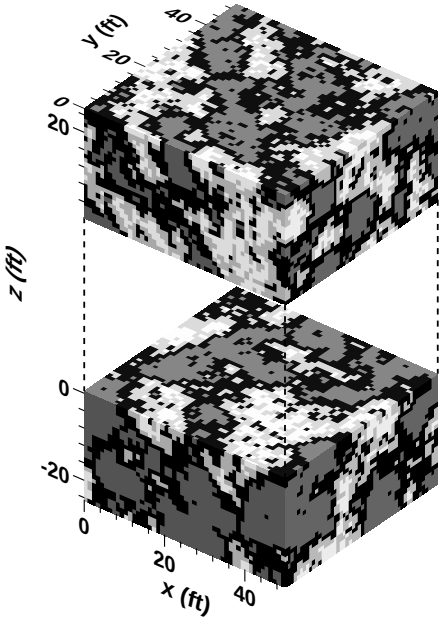
A $50 \times 50 \times 50$ realization is generated assuming isotropy.

```
4                                       /number of categories
0.4 0.3 0.2 0.1                         /proportions
../true/sim3d.bgr                       /output file
1                                       /output format: 1=binary, 2=ascii
1                                       /debugging level
tpsim.dbg                               /debugging file
3251                                    /seed
1                                       /number of simulations
1.0     50     1.0                      /xmin, nx+, xsiz
1.0     50     1.0                      /ymin, ny+, ysiz
-24.    50     1.0                      /zmin, nz+, zsiz
1  4                                    /ndmin, ndmax
1                                       /ibasis:0=cov,1=tp
0.001                                   /wratio
../true/tp3d.bgr                        /trans. prob. model file
../true/det3d.bgr                       /determinant file
../true/datcat.eas                      /input data file
0. 0.                                   /azimuths: coord, true
0. 0.                                   /dip:  coord, true
junkaz.bgr                              /azimuth int*1 file
junkdip.bgr                             /dip int*1 file
-1    0.00001  0                        /maxit; tol; -1=weight,1=lag1
0.05                                    /quenching determinant limit
```

Step 8 - View realization using CHUNK

```
0     0                                 /1=landscape; 1=11x17
-0.5 0.                                 /x,y plot translation (inches)
3.50 2.50                               /x,y legend translation (inches)
4         0                             /nbit 2,4,8; color(1) or gray(0)
1. 1. 1.    0. 0. 0.   0. 0. 0.         /bkgr, title, axes rgb
../../true/sim3d.bgr                    /input binary grid file
1 1 2                                   /# x,y,z chunks
0.  0. 40.                              /X,Y,Z spacing betw. chunks
0                                       /# of chunks to crop
1                                       /chunk#'s to crop
../../true/true3d.ps                    /output .ps file
-128. 127.                              /gmin gmax
8                                       /# cutoffs: cut, clr/gray
-4.0  16
-3.0  13
-2.0   8
-1.0   1
 1.0   1
 2.0   8
 3.0  13
 4.0  16
1  -2  -1                               /shading: XY,XZ,YZ ; +=light
0.  0. -24.5                            /Xmin,Ymin,Zmin
1. 1. 1.                                /dX, dY, dZ
0 0 0 0 0 0                             /cropping: xl,xr,yl,yr,zl,zr
20. 20. 20.                             /X,Y,Z scales (units/inch)
20. 20. 20.                             /X,Y,Z label increments
0 0 0                                   /X,Y,Z decimal places
5 5  5                                  /X,Y,Z tics per label
x (ft)                                  /X title
y (ft)                                  /Y title
z (ft)                                  /Z title
1.00                                    /Data scale factor
3-D conditional simulation              / title, line 1
based on "true.dat" data set            / title, line 2
0.92  -0.4 0.7 0.48                     /concat parameters
2  0                                    /legend? (0=no), vert?
4 4 1                                   /# of cats; nrow,ncol
2.0 2.0                                 /height & width
5                                       /category 1
1 (40%)                                 /label
6                                       /category 2
2 (30%)                                 /label
7                                       /category 3
3 (20%)                                 /label
8                                       /category 4
4 (10%)                                 /label
```

3-D conditional simulation
based on "true.dat" data set



LLNL Data Set

The Lawrence Livermore National Laboratory (LLNL) data set consists of about 5,500 m of semi-continuous vertical profiles (logs) of core descriptions of alluvial sediments obtained from 125 boreholes drilled for hydrogeologic characterization of the shallow aquifer system underlying LLNL (Qualheim, 1988). Lithofacies were categorized as *debris flow* (poorly sorted clay, silt, sand, gravel), *floodplain* (clay and silt), *levee* (silty fine sand), and *channel* (moderately to well sorted sand and gravel) deposits. This categorization was based on geologic interpretation of the depositional system with consideration for contrasts in hydrogeologic properties, to serve as a geologically sound basis for defining “hydrofacies” in detailed 3-D models of the groundwater flow system (Noyes, 1990).

Step 1 – Put data into GEOEAS format

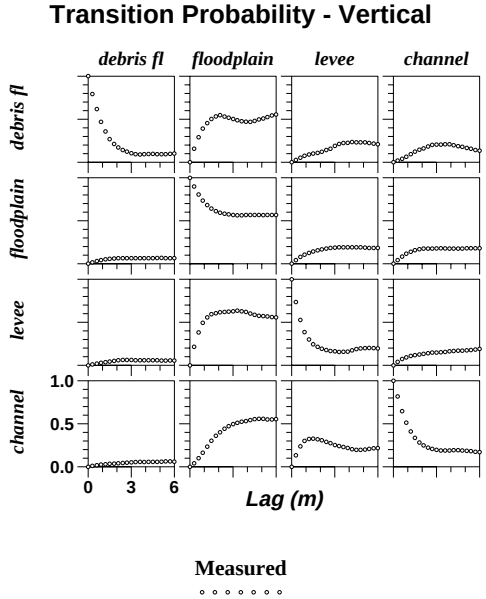
```
 Data
 7
  x    = easting
  y    = northing
  z    = elevation above mean sea level
 1 = debris flow
 2 = floodplain
 3 = levee
 4 = channel
    2132.8    2487.4    137.07   0   1   0   0
    2132.8    2487.4    136.77   0   1   0   0
    2132.8    2487.4    136.47   0   1   0   0
    2132.8    2487.4    136.17   0   1   0   0
    2132.8    2487.4    135.87   1   0   0   0
    2132.8    2487.4    135.57   1   0   0   0
    2132.8    2487.4    132.27   0   1   0   0
    2132.8    2487.4    131.97   0   1   0   0
    2576.2    2695.5    186.48   0   1   0   0
    2576.2    2695.5    182.28   0   0   0   1
    2576.2    2695.5    181.98   0   0   0   1
    2576.2    2695.5    181.68   0   0   0   1
    2576.2    2695.5    181.38   0   0   0   1
    2576.2    2695.5    181.08   0   1   0   0
    2576.2    2695.5    175.98   1   0   0   0
    2576.2    2695.5    175.68   0   1   0   0
    2576.2    2695.5    175.38   0   1   0   0
    2576.2    2695.5    112.98   0   1   0   0
.
.
.
```

Step 2 – Calculate vertical transition probabilities using GAMEAS

```
START OF PARAMETERS
data.eas                                  /input file
1 2 3                                     /x,y,z columns
4 4 5 6 7                                 /nvar, var1,2,3,... columns
-1. 2.                                    /vmin, vmax
datatpz.eas                               /output file
41                                        /# lags
0.3000                                    /lag spacing
0.1500                                    /lag tolerance
1                                         /ndir
0.0 90. 0.25 -90.0 22.50 0.25             /az,daz,azbw;dip,..,..
16                                        /# of bivariate statistics
1 1  11                                   /j,k, 11=tp
1 2  11
1 3  11
1 4  11
2 1  11
2 2  11
2 3  11
2 4  11
3 1  11
3 2  11
3 3  11
3 4  11
4 1  11
4 2  11
4 3  11
4 4  11
```

Step 3 – Plot vertical transition probabilities using GRAFXX

```
0                                       /symmetry check (1=sym)
0.20 0.20                               /dx dy between plots (inches)
0                                       /line at zero? 1=yes
1                                       /number of input files
../../llnl/tp/llnl1195tpz.eas           /input file 1
-10  0.55 0 1.                          /file 1: marker, lw, dash, gray
../../manual/llnl/llnltpz.ps            /output file
4                                       /number of categories
0. 6. 0.0 1.0                           /Xmin,Xmax,Ymin,Ymax
0 1                                     /# of x,y decimal places
5.4   0.9                               /X,Y scales (units/inch)
1.0                                     /Data scale factor
0. 0. 0.                                /axes color
3.  0.5                                 /X,Y label increments
3    5                                  /X,Y tics per label
Lag (m)                                 /X title
                                        /Y title
0                                       /1= titles for each plot
debris fl                               /X title variable 1
debris fl                               /Y title variable 1
floodplain                              /X title variable 2
floodplain                              /Y title variable 2
levee                                   /X title variable 3
levee                                   /Y title variable 3
channel                                 /X title variable 4
channel                                 /Y title variable 4
                                        /title, line 1
Transition Probability - Vertical       /title, line 2
1                                       /1=plot legend
7.5                                     /width of legend (inches)
Measured                                /label for file 1 data
```



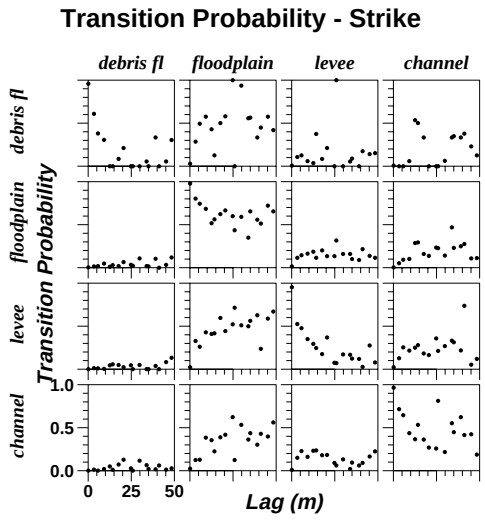
Step 4 – Calculate lateral transition probabilities using GAMEAS

Although the alluvial fan facies architecture is expected to be highly anisotropic, the lateral transition probabilities were calculated as if lateral isotropy were assumed. The anisotropy directions in the LLNL alluvial system substantially vary because of radial fan morphology, variation in fan source location, fan commingling, fluvial meandering, and deformation. The isotropic calculation was made under the assumption that the resulting transition probabilities would primarily reflect an upper limit to the strike-direction spatial continuity. It would also prescribe a lower limit to the dip-direction spatial continuity.

```
START OF PARAMETERS
../llnl/data/llnl1195.eas                  /input file
1 2 3                                      /x,y,z columns
4 4 5 6 7                                  /nvar, var1,2,3,... columns
-1. 2.                                     /vmin, vmax
../llnl/tp/llnltpx.eas                     /output file
25                                         /# lags
3.00                                       /lag spacing
1.5000                                     /lag tolerance
1                                          /ndir
0.0 90. 10000000.  0.0 22.50 0.30          /az,daz,azbw; dip,ddip,dipbw
16                                         /# of bivariate statistics
1 1  11                                    /j,k, 11=tp
1 2  11
1 3  11
1 4  11
2 1  11
2 2  11
2 3  11
2 4  11
3 1  11
3 2  11
3 3  11
3 4  11
4 1  11
4 2  11
4 3  11
4 4  11
```

Step 5 – Plot lateral transition probabilities using GRAFXX

```
0                                          /symmetry check (1=sym)
0.20 0.20                                  /dx dy between plots (inches)
0                                          /line at zero? 1=yes
1                                          /number of input files
../../llnl/tp/llnltpx.eas                  /input file 1
-10  0.5 0 0.                             /marker, lw, dash, gray
~/geostats/manual/llnl/llnltpx.ps          /output file
4                                          /number of categories
0. 50. 0.0 1.0                             /Xmin,Xmax,Ymin,Ymax
0 1                                        /# of X, Y decimal places
45.   0.9                                  /X,Y scales (units/inch)
1.0                                        /Data scale factor
0. 0. 0.                                   /axes color
25.  0.5                                   /X,Y label increments
5    5                                     /X,Y tics per label
Lag (m)                                    /X title
Transition Probability                     /Y title
0                                          /1= titles for each plot
debris fl                                  /X title variable 1
debris fl                                  /Y title variable 1
floodplain                                 /X title variable 2
floodplain                                 /Y title variable 2
levee                                      /X title variable 3
levee                                      /Y title variable 3
channel                                    /X title variable 4
channel                                    /Y title variable 4
                                           /title, line 1
Transition Probability - Strike            /title, line 2
1                                          /1=plot legend
8.5                                        /width of legend (inches)
Data
```



Step 6 – Develop 1- and 3-D Markov Chain models using MCMOD

The following procedure, used to develop the LLNL principal direction models, may guide application of **MCMOD** to typical data sets derived from boreholes:

1. Initially develop the vertical-direction model, using either option 1, 2, or 3. For the strike and dip directions, arrive at plausible mean lengths consistent with the transition probability data (if available) to define the diagonal entries. For the off-diagonal entries, assume symmetry and use option 3 to estimate embedded transition probabilities. By assuming symmetry and a background category, only three of the twelve off-diagonal transition rates need to be prescribed for a four-category system (only one of six for a three-category system).

2. Look at the debugging output file from **MCMOD**, which includes other interpretations yielding an equivalent Markov chain model. Option 3, the embedded transition probability framework, is usually the most interpretable. If necessary, refine the vertical-direction model.

3. For the strike and dip-direction models, initiate the off-diagonal embedded transition probabilities by considering Walther's Law concepts, that vertical facies successions reflect lateral facies successions. For the off-diagonal terms needed to complete the strike and dip direction models, assign the embedded transition probabilities obtained from the vertical model. Refine the strike and dip direction models as necessary to maintain geologic plausibility, adherence to probability law, and consistency with transition probability data.

```
4                                       /# of categories
0.066 0.565 0.190  0.179                /proportions
2                                       /background category
../llnl/tp/mcmod.dbg                    /name of debugging file
../llnl/tp/tpxyz.bgr                    /output file for 3-D model
../llnl/tp/det.bgr                      /output file for determinant
0.05  0.05  0.05                        /determinant limits for 3-D model
3.0  10.0  0.30                         /dhx,dhy,dhz for 3-D model
../llnl/tp/llnltpxm.eas                 /X-direction output file
200     1.                              /X-Direction: # lags, spacing
3                                       /option: 1=r,2=d,3=etp,4=etf,5=i
8.0     0.   -1.   -1.
0.      0.    0.    0.
0.027   0.   6.0   -1.
0.041   0.   0.839  10.
../llnl/tp/llnltpym.eas                 /Y-direction output file
200     2.5                             /Y-Direction; # lags, spacing
3                                       /1=r,2=d,3=etp,4=etf,5=i,6=p,7=f
24.     0.   -1.     -1.
 0.     0.    0.      0.
 0.027  0.   20.0    -1.
 0.041  0.    0.839  50.
../llnl/tp/llnltpzm3.eas                /Z-direction output file
200   0.1                               /Z-Direction: # lags, spacing
2                                       /option: 1=r,2=d,3=etp,4=etf,5=i
../llnl/tp/llnl1195tpz.eas              /data file
3                                       /lag #
```

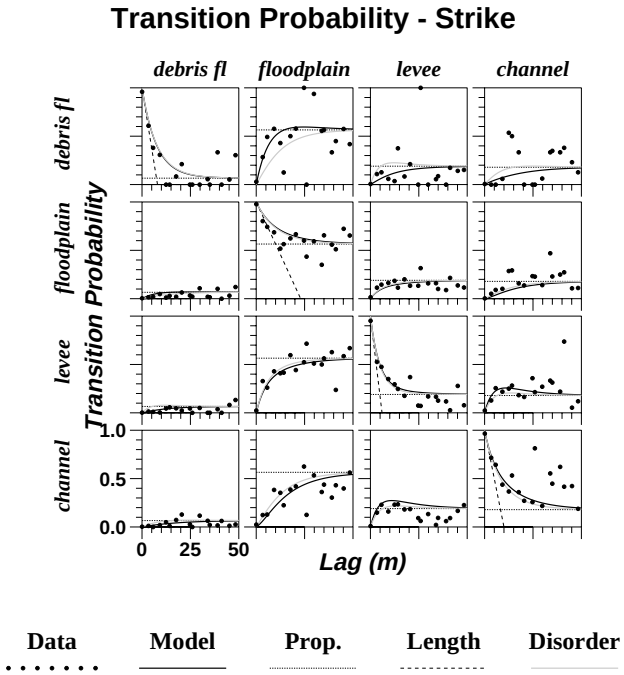
```
 MCMOD debugging file

Parameter file: mcmodllnl3.par
Number of categories: 4
Proportions:  0.0660  0.5650  0.1900  0.1790
Background category: 2


------- X-DIRECTION: -------
.
.
.
------- Z-DIRECTION: -------
Method - option 2: trans. prob. at specified lag
1-D model output file: ../llnl/tp/llnltpzm3.eas
WARNING: Off-diagonal Transition Rate   4  2          is negative.

Rate Matrix for Z-Direction:
 -0.871510  0.702561  0.103207  0.065742
  0.079481 -0.435999  0.150644  0.205875
  0.029521  1.073549 -1.227927  0.124857
  0.039128 -0.022367  0.789836 -0.806597

embedded transition probabilities:
  1.000000  0.806142  0.118423  0.075434
  0.182296  1.000000  0.345514  0.472190
  0.024042  0.874278  1.000000  0.101681
  0.048510 -0.027731  0.979221  1.000000

embedded transition frequencies:
  0.084396  0.068035  0.009994  0.006366
  0.065889  0.361442  0.124883  0.170669
  0.008230  0.299282  0.342319  0.034807
  0.010276 -0.005875  0.207441  0.211843
 entropy=   -56.8526

w.r.t. independent transition freqs:
(    1.1474)     1.8393      0.3121      0.4138
     1.7813 (    2.2936)     0.5699      1.6207
     0.2570      1.3657 (    0.8144)     0.3819
     0.6679     -0.0558      2.2758 (    1.2398)
.
.
.
 Constructing 3-D transition probability model
 # of lags in +x,+y,+z direction =  7  7  9
 total # of lags =  4275
```

Step 7 – Calculate independent or maximum entropy (disorder) model

```
4                                       /# of categories
0.066 0.565 0.190  0.179                /proportions
2                                       /background category
../llnl/tp/mcmodtpze.dbg                /name of debugging file
../llnl/tp/tpxyze.bgr                   /output file for 3-D model
../llnl/tp/dete.bgr                     /output file for determinant
1.000   1.000  1.000                     /determinant limit for 3-D model
3.0 10.0  0.30                          /dhx,dhy,dhz for 3-D model
../llnl/tp/llnltpxme.eas                /X-direction output file
200     1.                              /X-Direction: # lags, spacing
5                                       /option: 1=r,2=d,3=etp,4=etf,5=i
8.0     0.     1.     1.
0.     22.965  0.     0.
1.0     0.     6.0    1.
1.0     0.     1.0    10.
../llnl/tp/llnltpyme.eas                /Y-direction output file
200     2.5                             /Y-Direction; # lags, spacing
5                                       /option: 1=r,2=d,3=etp,4=etf,5=i
24.     0.     1.     1.
 0.    62.665  0.     0.
 1.0    0.    20.0    1.
 1.0    1.     1.    50.
../llnl/tp/llnltpzme3.eas               /Z-direction output file
200   0.1                               /Z-Direction: # lags, spacing
5                                       /option: 1=r,2=d,3=etp,4=etf,5=i
1.1474  0.     1.     1.
0.      2.2936 0.     0.
1.0     0.     0.8144 1.
1.0     0.     1.     1.2398
```

Step 8 – Compare measured and modeled transition probabilities

```
0                                       /symmetry check (1=sym)
0.20 0.20                               /dx dy between plots (inches)
0                                       /line at zero? 1=yes
5                                       /number of input files
../../llnl/tp/llnltpx.eas               /input file 1
-10  0.5 0 0.                          /marker, lw, dash, gray
../../llnl/tp/llnltpxm.eas              /input file 5
0 1.0 0 0.                              /marker, lw, dash, gray
../../llnl/tp/proptpx.eas               /input file 2
0 0.5 1  0.                             /marker, lw, dash, gray
../../llnl/tp/mltpx.eas                 /input file 3
0 0.5 3  0.                             /marker, lw, dash, gray
../../llnl/tp/llnltpxme.eas             /input file 4
0 1.0  0  0.8                           /marker, lw, dash, gray
~/geostats/manual/llnl/llnltpxm.ps       /output file
4                                       /number of categories
0. 50. 0.0 1.0                          /Xmin,Xmax,Ymin,Ymax
0 1                                     /# of X, Y decimal places
45.   0.9                               /X,Y scales (units/inch)
1.0                                     /Data scale factor
0. 0. 0.                                /axes color
25.  0.5                                /X,Y label increments
5    5                                  /X,Y tics per label
Lag (m)                                 /X title
Transition Probability                  /Y title
0                                       /1= titles for each plot
debris fl                               /X title variable 1
debris fl                               /Y title variable 1
floodplain                              /X title variable 2
floodplain                              /Y title variable 2
levee                                   /X title variable 3
levee                                   /Y title variable 3
channel                                 /X title variable 4
channel                                 /Y title variable 4
                                        /title, line 1
Transition Probability - Strike         /title, line 2
1                                       /1=plot legend
7.5                                     /width of legend (inches)
Data
Model
Prop.
Length
Disorder
```

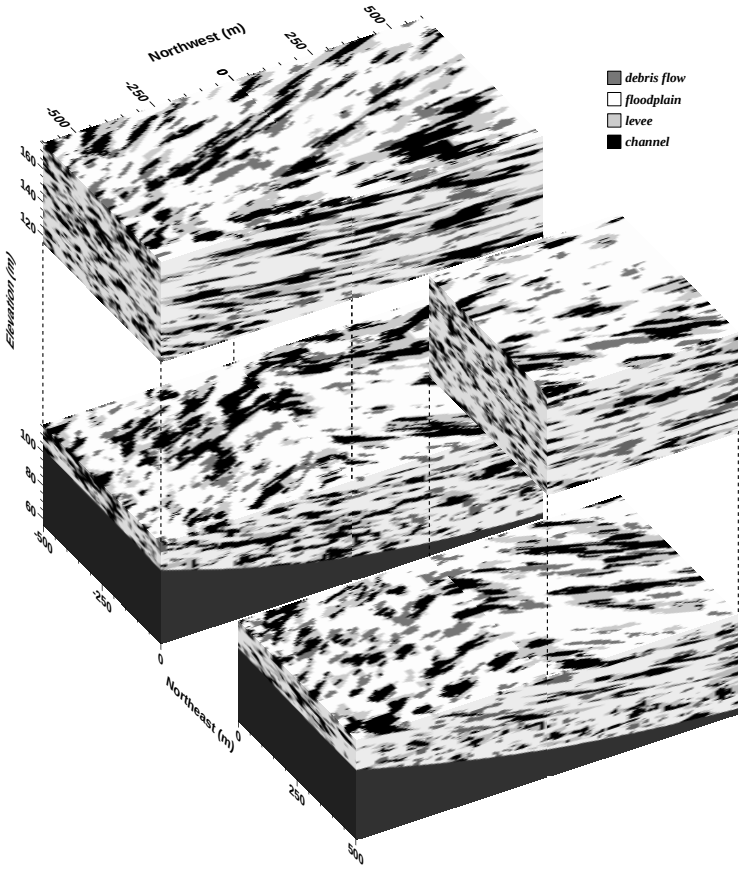


Step 9 – Generate a 3-D realization using TSIM

```
4                                       /number of categories
0.07 0.56 0.19 0.18                     /proportions
../examples/llnlsim.bgr                 /output file
1                                       /debugging level
tpsim.dbg                               /debugging file
2311                                    /seed
1                                       /number of simulations
1955.292    -200      3.0               /xcenter, nx, xsiz
2692.908    -100     10.0               /ycenter, ny, ysiz
 127.50     -241      0.3               /zcenter, nz, zsiz
1  4                                    /ndmin, ndmax
1                                       /ibasis:0=cov,1=tp
0.001                                   /wratio
../tp/llnltpxyz.bgr                     /trans. prob. model file
../tp/llnldetxyz.bgr                    /spectral radius file
../data/llnl1195.eas                    /input data file
0. 0.                                   /azimuths: coord, true
0. 0.                                   /dip:  coord, true
../examples/llnlaz.bgr                  /azimuth int*1 file
../examples/llnldip.bgr                 /dip int*1 file
4      0.00001  0                       /maXit; tol; -1=no dcl,1=lag1
0.5                                     /quenching determinant limit
```

Step 10 – View realization using CHUNK

```
1                                       /landscape(1) or portrait(0)
3.0 3.5                                 /x y plot translation
3.5 10.0                                /x y legend translation
4        0                              /nbit 2,4,8; color(1)or  gray(0)
1. 1. 1.    0. 0. 0.   0. 0. 0.         /rgb: bkgr, title, axes
../examples/llnlsim.bgr                 /input binary grid file
2 2 2                                   /# x,y,z chunks
330. 0.  110.                           /X,Y,Z spacing betw. chunks
1                                       /# of chunks to crop
8                                       /chunk #'s to crop
../examples/llnlsim.ps                  /output .ps file
10                                      /# of categories
-128.  5                                /value, gray
-4.0  1
-3.0  13
-2.0  16
-1.0  8
1.0   8
2.0   16
3.0   13
4.0   1
127.  14
0   -2    -1                            /shading: XY,XZ,YZ(+light -dark)
-601.5 -1005. 55.05                     /Xmin,Ymin,Zmin
3. 10. 0.3                              /dX, dY, dZ
33 33 40 40 5  83                       /cropping: xl,xu,yl,yu,zl,zu
160. 160. 32.                           /X,Y,Z scales (units/inch)
250. 250. 20.                           /X,Y,Z label increments
0 0 0                                   /X,Y,Z decimal places
5  5 4                                  /X,Y,Z tics per label
Northeast (m)                           /X title
Northwest (m)                           /Y title
Elevation (m)                           /Z title
1.00                                    /Data scale factor
                                        / title, line 2
                                        / title, line 2
.707 -.707   0.95 0.312                 /concat paramters
2  0 0                                  /legend, legend orientation?
4 4 1                                   /# of categories; #rows,cols
1.6 1.0                                 /legend height, width
6                                       /category #
debris flow                             /label
7                                       /category #
floodplain                              /label
8                                       /category #
levee                                   /label
9                                       /category #
channel                                 /label
```



LAAPMO4C Data Set

The LAAPMO4C data set consists of continuous vertical profiles (logs) of core descriptions of fluvial sediments categorized as SM (silty sand), ML (silt), CL/ML (clayey silt or silty clay), CL/CH (clay or “fat clay”).

The goal of this example is to generate a 3-D realization that is consistent with the data and spatial variability. This example represents a typical hydrogeologic application, where a detailed and realistic hydrogeologic model is needed, but the data, although intensively sampled, are not adequate to exactly determine the true hydrogeologic structure.

Step 1 – Put data into GEOEAS format

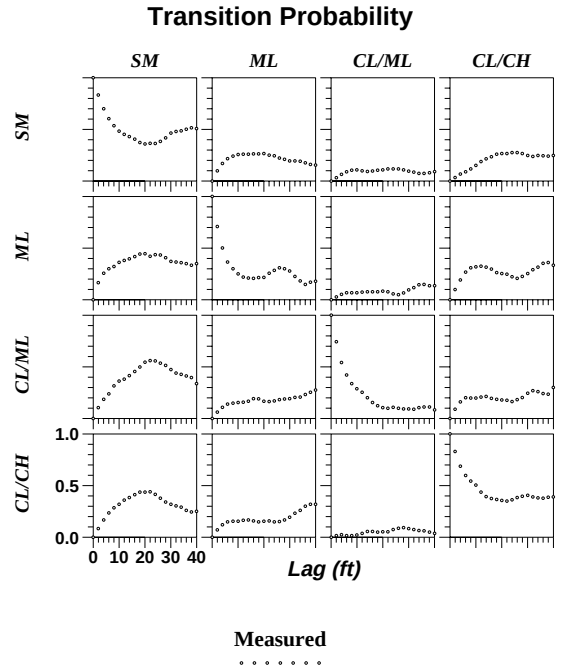
```
Data
7
 x    = easting
 y    = northing
 z    = elevation above mean sea level
1
2
3
4
   3836.1    1425.4    215.12   0   0   0   1
   3836.1    1425.4    213.12   0   0   0   1
   3836.1    1425.4    211.12   0   0   0   1
   3836.1    1425.4    209.12   0   0   0   1
   3836.1    1425.4    207.12   0   0   0   1
   3836.1    1425.4    205.12   0   0   0   1
   3836.1    1425.4    203.12   0   0   0   1
   3836.1    1425.4    201.12   1   0   0   0
   3836.1    1425.4    199.12   1   0   0   0
   3836.1    1425.4    197.12   1   0   0   0
   3836.1    1425.4    195.12   1   0   0   0
   3836.1    1425.4    193.12   1   0   0   0
   3836.1    1425.4    191.12   1   0   0   0
   3836.1    1425.4    189.12   1   0   0   0
   3836.1    1425.4    187.12   1   0   0   0
   3836.1    1425.4    185.12   0   0   0   1
   5242.6    1494.6    218.71   0   0   0   1
   5242.6    1494.6    216.71   0   0   0   1
   5242.6    1494.6    214.71   0   0   0   1
   5242.6    1494.6    212.71   0   0   0   1
   5242.6    1494.6    210.71   0   0   0   1
   5242.6    1494.6    208.71   0   0   0   1
   5242.6    1494.6    206.71   0   0   0   1
.
.
.
```

Step 2 – Calculate vertical transition probabilities using GAMEAS

```
START OF PARAMETERS
../examples/laapmo4c.eas                  /input file
1 2 3                                     /x,y,z columns
4 4 5 6 7                                 /nvar, var1,2,3,... columns
-1. 2.                                    /vmin, vmax
../examples/laapmo4ctpz.eas               /output file
30                                        /# lags
2.                                        /lag spacing
1.                                        /lag tolerance
1                                         /ndir
0. 90.0 10.  90.0 22.50 1.0               /az,daz,azbw;dip,ddip,dipbw
16                                        /# of bivariate statistics
1 1  11                                   /j,k, 11=tp
1 2  11
1 3  11
1 4  11
2 1  11
2 2  11
2 3  11
2 4  11
3 1  11
3 2  11
3 3  11
3 4  11
4 1  11
4 2  11
4 3  11
4 4  11
```

Step 3 – Plot vertical transition probabilities using GRAFXX

```
0                                       /symmetry check (1=sym)
0.20 0.20                               /dx dy between plots (inches)
0                                       /line at zero? 1=yes
1                                       /number of input files
../examples/laapmo4ctpz.eas             /input file 1
-10 0.40  0 1.00                        /file 1: marker, lw, dash, gray
../examples/tpz.ps                      /output file
4                                       /number of categories
0. 40.  0.0 1.0                         /Xmin,Xmax,Ymin,Ymax
0 1                                     /# of x,y decimal places
30.   0.75                             /X,Y scales (units/inch)
1.0                                     /Data scale factor
0.0                                     /axes gray level
10.  0.5                                /X,Y label increments
5     5                                 /X,Y tics per label
Lag (ft)                                /X title
                                        /Y title
0                                       /1= titles for each plot
SM                                      /X title variable 1
SM                                      /Y title variable 1
ML                                      /X title variable 2
ML                                      /Y title variable 2
CL/ML                                   /X title variable 3
CL/ML                                   /Y title variable 3
CL/CH                                   /X title variable 4
CL/CH                                   /Y title variable 4
                                        /title, line 1
Transition Probability                  /title, line 2
1                                       /1=plot legend
6.5                                     /width of legend (inches)
Measured
```



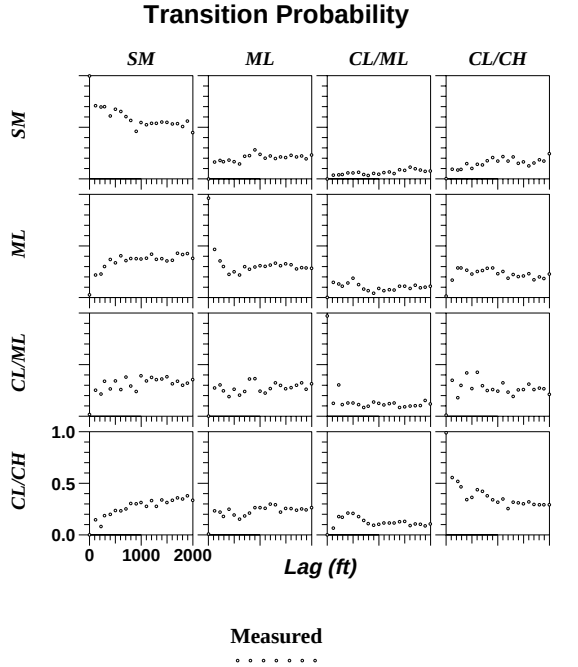
Step 4 – Calculate lateral transition probabilities using GAMEAS

In this application, the lateral spatial variability is assumed isotropic, that is, the dip- and strike-direction patterns of heterogeneity are assumed to be similar. Alternatively, soft information could have been used to create an anisotropic model.

```
START OF PARAMETERS
../examples/laapmo4c.eas                    /input file
1 2 3                                       /x,y,z columns
4 4 5 6 7                                   /nvar, var1,2,3,... columns
-1. 2.                                      /vmin, vmax
../examples/laapmo4ctpxy.eas                /output file
25                                          /# lags
100.                                        /lag spacing
50.                                         /lag tolerance
1                                           /ndir
0.2 90.1 10000000.  0.0 22.50 1.0           /az,daz,azbw;dip,ddip,dipbw
16                                          /# of bivariate statistics
1 1  11                                     /j,k, 11=tp
1 2  11
1 3  11
1 4  11
2 1  11
2 2  11
2 3  11
2 4  11
3 1  11
3 2  11
3 3  11
3 4  11
4 1  11
4 2  11
4 3  11
4 4  11
```

Step 5 – Plot lateral transition probabilities using GRAFXX

```
0                                           /symmetry check (1=sym)
0.20 0.20                                   /dx dy between plots (inches)
0                                           /line at zero? 1=yes
1                                           /number of input files
../examples/laapmo4ctpxy.eas                /input file 1
-10 0.40  0 1.00                            /file 1: marker, lw, dash, gray
../examples/tpxy.ps                         /output file
4                                           /number of categories
0. 2000.  0.0 1.0                           /Xmin,Xmax,Ymin,Ymax
0 1                                         /# of x,y decimal places
1500.   0.75                                /X,Y scales (units/inch)
1.0                                         /Data scale factor
0.0                                         /axes gray level
1000.  0.5                                  /X,Y label increments
10     5                                    /X,Y tics per label
Lag (ft)                                    /X title
                                            /Y title
0                                           /1= titles for each plot
SM                                          /X title variable 1
SM                                          /Y title variable 1
ML                                          /X title variable 2
ML                                          /Y title variable 2
CL/ML                                       /X title variable 3
CL/ML                                       /Y title variable 3
CL/CH                                       /X title variable 4
CL/CH                                       /Y title variable 4
                                            /title, line 1
Transition Probability                      /title, line 2
1                                           /1=plot legend
6.5                                         /width of legend (inches)
Measured
```



Step 6 – Develop 1- and 3-D Markov Chain models using MCMOD

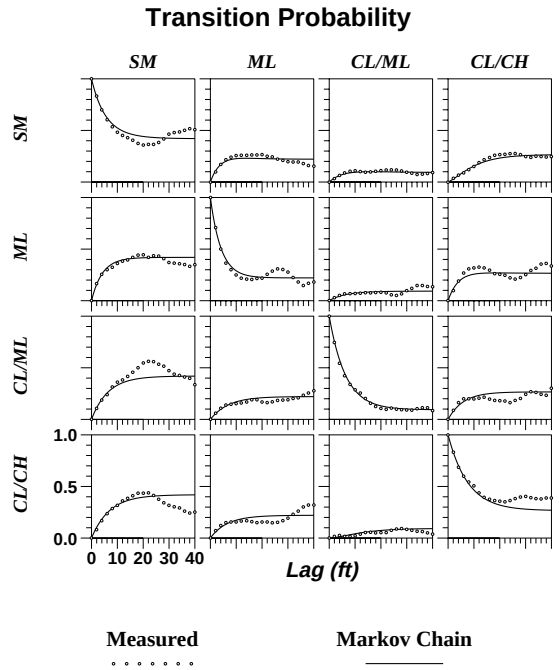
```
4                                       /# of categories
0.420 0.221 0.093 0.266                 /proportions
4                                       /background category
../examples/mcmodtp.dbg                 /name of debugging file
../examples/tp3d.bgr                    /output file for 3-D model
../examples/det3d.bgr                   /output file for determinant
0.020   0.020 0.020                     /spectral extent for 3-D model
50. 50. 2.                              /dhx,dhy,dhz for 3-D model
../examples/tpxm.eas                    /X-direction output file
201     10.                             /X-Direction: # lags, spacing
3                                       /option: 1=r,2=d,3=etp,4=etf,5=i
550.   -1.   -1.    0.
0.30  150.   -1.    0.
0.15  0.5    90.    0.
0.     0.     0.    0.
../examples/tpym.eas                    /Y-direction output file
201     10.                             /Y-Direction; # lags, spacing
3                                       /option: 1=r,2=d,3=etp,4=etf,5=i
550.   -1.   -1.    0.
0.30  150.   -1.    0.
0.15  0.5    90.    0.
0.    0.      0.    0.
../examples/tpzm2.eas                   /Z-direction output file
  201   0.25                            /Z-Direction: # lags, spacing
3                                       /option: 1=r,2=d,3=etp,4=etf,5=i
  9.50   0.67   0.22  0.
  0.53   5.30   0.08  0.
  0.46   0.22   7.00  0.
  0.     0.     0.    0.
```

Step 7 – Compare measured and modeled transition probabilities

In practice, steps 6 and 7 were implemented repeatedly until a satisfactory fit is obtained. The following procedure can be useful for developing the principal direction models:

1. First apply option 2 in **MCMOD**, which fits a Markov chain directly to transition probabilities at a specific lag.
2. Plot the measured and modeled transition probabilities using **GRAFXX**.
3. Look at the debugging file output from **MCMOD**, which includes other interpretations yielding an equivalent Markov chain model. The embedded transition probability framework (option 3 in **MCMOD**) is usually the most interpretable.
4. Apply option 3 in **MCMOD**. Starting from equivalent parameters obtained in first model, adjust the embedded transition probabilities to raise or the lower transition rates. Recall that background column entries are dictated by the other entries in the same row, and that background row entries are dictated by the other entries in the same column.

```
0                                       /symmetry check (1=sym)
0.20 0.20                               /dx dy between plots (inches)
0                                       /line at zero? 1=yes
1                                       /number of input files
../examples/laapmo4ctpz.eas             /input file 1
-10 0.40  0 1.00                        /file 1: marker, lw, dash, gray
../examples/tpz.ps                      /output file
4                                       /number of categories
0. 40.  0.0 1.0                         /Xmin,Xmax,Ymin,Ymax
0 1                                     /# of x,y decimal places
30.   0.75                             /X,Y scales (units/inch)
1.0                                     /Data scale factor
0.0                                     /axes gray level
10.  0.5                                /X,Y label increments
5     5                                 /X,Y tics per label
Lag (ft)                                /X title
                                        /Y title
0                                       /1= titles for each plot
SM                                      /X title variable 1
SM                                      /Y title variable 1
ML                                      /X title variable 2
ML                                      /Y title variable 2
CL/ML                                   /X title variable 3
CL/ML                                   /Y title variable 3
CL/CH                                   /X title variable 4
CL/CH                                   /Y title variable 4
                                        /title, line 1
Transition Probability                  /title, line 2
1                                       /1=plot legend
6.5                                     /width of legend (inches)
Measured
```

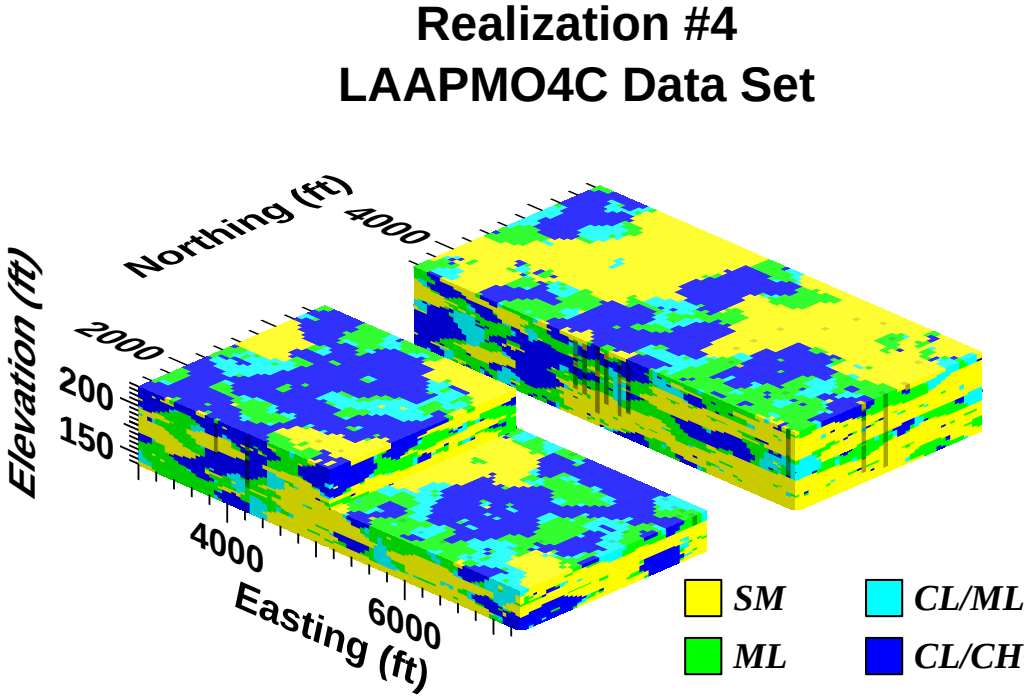


Step 8 – Generate a 3-D realization using TSIM

```
4                                       /number of categories
0.420 0.221 0.093 0.266                 /proportions
../examples/sim3d.bgr                   /output file
1                                       /output format: 1=binary, 2=ascii
1                                       /debugging level
tpsim.dbg                               /debugging file
3251                                    /seed
5                                       /number of simulations
3000.     90     50.0                   /xmin, nx, xsiz
1400.     90     50.0                   /ymin, ny, ysiz
140.       50     2.0                   /zmin, nz, zsiz
1  6                                    /ndmin, ndmax
1                                       /ibasis:0=cov,1=tp
0.001                                   /wratio
../examples/tp3d.bgr                    /trans. prob. model file
../examples/det3d.bgr                   /determinant file
../examples/laapmo4c.eas                /input data file
0. 0.                                   /azimuths: coord, true
0. 0.                                   /dip:  coord, true
junkaz.bgr                              /azimuth int*1 file
junkdip.bgr                             /dip int*1 file
 3     0.00001  0                       /maxit; tol; -1=weight,1=lag1
0.30                                    /quenching determinant limit
```

Step 9 – View realization using CHUNK

```
0                                       /landscape(1) or portrait(0)
0.35 0.40                               /x,y plot translation (inches)
2.0 0.7                                 /x,y legend translation (inches)
4          1                            /nbit 2,4,8; color(1) or gray(0)
1. 1. 1.    0. 0. 0.   0. 0. 0.         /rgb: bkgr, title, axes
../examples/laapmo4c.bgr4               /input binary grid file
2 2 2                                   /# x,y,z chunks
0. 1000. 0.                             /X,Y,Z spacing betw. chunks
1                                       /# of chunks to crop
8                                       /chunk#'s to crop
../examples/laapmo4c4.ps                /output .ps file
8                                       /# of categories
-4    1  1 11
-3    1 11 11
-2    1 11  1
-1   11 11  1
 1   16 16  1
 2    1 16  1
 3    1 16 16
 4    1  1 16
3  -3  0                                /shading: XY,XZ,YZ (+light -dark)
3000. 1400. 140.                        /Xmin,Ymin,Zmin
50. 50.  2.                             /dX, dY, dZ
0 4 1 5 0  9                            /cropping: xl,xu,yl,yu,zl,zu
1500. 1500. 150.                        /X,Y,Z scales (units/inch)
2000. 2000. 50.                         /X,Y,Z label increments
0 0 0                                   /X,Y,Z decimal places
10 10 10                                /X,Y,Z tics per label
Easting (ft)                            /X title
Northing (ft)                           /Y title
Elevation (ft)                          /Z title
1.00                                    /Data scale factor
Realization #4                          / title, line 1
LAAPMO4C Data Set                       / title, line 2
0.92  -0.4 0.92 0.4                     /concat parameters
1                                       /legend? (1=yes)
4 2 2                                   /# of cats; nrow,ncol
0.8 2.5                                 /height & width
5                                       /category #
SM                                      /label
6                                       /category #
ML                                      /label
7                                       /category #
CL/ML                                   /label
8                                       /category #
CL/CH                                   /label
```



References

Aarts, E., and Korst, J., 1989, Simulated annealing and Boltzmann machines: John Wiley & Sons, New York, 272 p.

Adobe Systems Incorporated, 1990, PostScript language reference manual, 2nd edition: Adison-Wesley, New York, 765 p.

Agterberg, F. P., 1974, Geomathematics: Elsevier, Scientific Publishing Co., Amsterdam, New York, 596 p.

Almeida, A. S., Journel, A. G., 1994, Joint simulation of multiple variables with a Markov-type coregionalization model: Math. Geology, v. 26, no. 5, p. 565-588.

Carle, S. F., 1996, A transition probability-based approach to geostatistical characterization of hydrostratigraphic architecture: Ph. D. dissertation, University of California, Davis, 248 p.

Carle, S. F., 1997a, Integration of geologic interpretation into geostatistical simulation: IAMG'97 Proceedings of The Third Annual Conference of the International Association for Mathematical Geology, V. Pawlowsky Glahn (Ed.), International Center for Numerical Methods, Barcelona, Spain.

Carle, S. F., 1997b, Implementation schemes for avoiding artifact discontinuities in simulated annealing: Mathematical Geology, v. 29, no. 2, p. 231-244.

Carle, S. F., and Fogg, G. E., 1996, Transition probability-based indicator geostatistics: Mathematical Geology, v. 28, no. 4, p. 453-477.

Carle, S. F., and Fogg, G. E., 1997, Modeling spatial variability with one- and multi-dimensional Markov chains: Mathematical Geology, v. 28, no. 7.

Carle, S. F., Labolle, E. M., Weissmann, G. S., Van Brocklin, D., and Fogg, G. E., 1998, Conditional simulation of hydrofacies architecture: a transition probability/Markov approach, In Fraser, G. S., and Davis, J. M., Concepts in Hydrogeology and Environmental Geology No. 1, SEPM (Society for Sedimentary Geology) Special Publication, p. 147-170.

Carr, D. D., Horowitz, A., Hrarbar, S. V., Ridge, K. F., Rooney, R., Straw, W. T., Webb, W., and Potter, P. E., 1966, Stratigraphic sections, bedding sequences and random processes: Science, v. 28, no. 3753, p. 89-110.

Deutsch, C. V., and Cockerham, P. W., 1994, Practical considerations in the application of simulated annealing in stochastic simulation: Math. Geology, v. 26, no. 1, p. 67-82.

Deutsch, C. V., and Journel, A. J., 1992, Geostatistical software library and user's guide: Oxford University Press, New York, 340 p.

Doveton, J. H., 1971, An application of Markov chain analysis to the Ayrshire Coal Measures succession: Scott J. Geol., v. 7, no. 1, p. 11-27.

Ethier, V. G., 1975, Application of Markov analysis to the Banff Formation (Mississippian), Alberta: Math. Geology, v. 7, no. 1, p. 47-61.

Goodman, L. A., 1968, The analysis of cross-classified data – independence, quasi-independence, and interactions in contingency table with and without missing entries: American Statistical Association Journal, v. 63, p. 1091-1131.

Goovaerts, P., 1996, Stochastic simulation of categorical variables using a classification algo-

rithm and simulated annealing: Math. Geology, v. 28, no. 7, p. 909-921.

Hattori, I., 1976, Entropy in Markov chains and discrimination of cyclic patterns in lithologic successions: Math. Geology, v. 8, no. 4, p. 477-497.

Johnston, R. J., and Pattie, C. J., 1993, Entropy-maximizing and the iterative proportional fitting procedure: Professional Geographer, v. 45, p. 317-322.

Journel, A. G., 1983, Nonparametric estimation of spatial distributions: Math. Geology, v. 1, no. 1, p. 79-96.

Krumbein, W. C., 1968, Fortran IV computer programs for simulation of transgression and regression with continuous-time Markov models: Computer Contribution 26, Kansas State Geological Survey, p. 1-18.

Krumbein, W. C., and Dacey, M. F., 1969, Markov chains and embedded Markov chains in geology: Math. Geology, v. 1, no. 1, p. 79-96.

Lin, C., and Harbaugh, J. W., 1984, Graphic display of two- and three- dimensional Markov computer models in geology: Van Nostrand Reinhold, New York, 180 p.

Miall, A. D., 1973, Markov chain analysis applied to an ancient alluvial plain succession: Sedimentology, v. 20, no. 3, p. 347-364.

Noyes, C. D., 1990, Hydrostratigraphic analysis of the Pilot Remediation Test Area, LLNL, Livermore, California: unpubl. Masters Thesis, University of California, Davis.

Politis, D. N., 1994, Markov chains in many dimensions: Adv. Appl. Prob., v. 26, no. 3, p. 756-774.

Qualheim, B. J., 1988, Well log report for the LLNL ground water project 1984-1987: Lawrence Livermore National Laboratory, UCID-21342 (updated 1989 and 1993).

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1992, Numerical recipes in FORTRAN: Cambridge Univ. Press, New York, 963 p.

Schwarzacher, W., 1969, The use of Markov chains in the study of sedimentary cycles: Math. Geology, v. 12, no. 3, p. 213-234.