

Building Blocks for Direct Sequential Simulation on Unstructured Grids

M. J. Pyrcz (mpyrcz@ualberta.ca) and C. V. Deutsch (cdeutsch@ualberta.ca)
University of Alberta, Edmonton, Alberta, CANADA

Abstract

Direct sequential simulation (DSS) of continuous variables on unstructured grids requires a number of frequently called subroutines. The average covariance between two arbitrary blocks $v_1(\mathbf{u})$ and $v_2(\mathbf{u}')$ must be calculated quickly and efficiently; subroutines to setup this calculation and to perform the required numerical integration are presented. The speed of DSS on unstructured grids is enhanced with use of a non-stationary covariance lookup table of average covariance between blocks; subroutines for construction of this table and its use in DSS are documented. A central requirement of DSS is fast retrieval of the shape of conditional distributions; subroutines for the setup of a shape lookup table and fast lookup by super block search have been developed. These building blocks are essential for many aspects of DSS with regular and unstructured grids. Clean, well documented, and tested code has been prepared for widespread use by CCG researchers and member companies.

It is common when simulating directly to unstructured grids to encounter spurious kriging weights as a result of the screening effect and limitations in precision. Techniques for addressing this problem are also presented.

Introduction

There are two key requirements to work within the framework of DSS on irregular grids: (1) valid local distributions of uncertainty must be rapidly generated for all estimates and estimation variances and (2) mean covariance values must be quickly calculated for any two volumes. These processes must be rapid in comparison with the brute force method of simulating on a fine grid and block averaging a posteriori to an irregular grid.

A framework for the first requirement has been presented by Deutsch et. al (2001). A series of distributions, parameterized by kriging estimate and kriging variance that result in the global distribution being honored are constructed. This procedure utilizes the link between the global direct distribution and the Gaussian distribution. This procedure is reviewed in the program design section. A novel approach to meet the second requirement (mean covariance calculation and retrieval) is presented in this work.

We believe these solutions facilitate the transition of geostatistics to the direct simulation of irregular grid stochastic models. These subroutines are documented and provided as building blocks for future research and development in this area.

CCDF Shape Determination

The kriging system, which is at the heart of most simulation algorithms, provides an estimate and an estimation variance. When working in Gaussian (y) space these two

parameters are sufficient to characterize the local distributions of uncertainty, since the local distributions are Gaussian. The consistent application of Gaussian distributions ensures that the global distribution is reproduced. In the DSS framework, the local distributions are not sufficiently characterized by the kriging estimate and the estimation variance. A method to determine the shape of the local distributions, such that the global is reproduced is required, see Figure 1. The key is the link between direct space (z space) and Gaussian space (y space) through the global distribution.

Given a specific quantile of a y space distribution with mean (m) and standard deviation (σ) a z space quantile may be calculated by the following procedure. The inverse of the specific quantile is calculated as the inverse of the nonstandard Gaussian function, $G_{\{m,\sigma\}}$.

$$Val1 = G_{\{m,\sigma\}}^{-1}(q) \quad (1)$$

Then, the associated probability is calculated from the standard Gaussian distribution, $G_{\{0,1\}}$.

$$Val2 = G_{\{0,1\}}[Val1] \quad (2)$$

$$\text{or } Val2 = G_{\{0,1\}}[G_{\{m,\sigma\}}^{-1}(q)] \quad (3)$$

Now the link between the global Y space and global Z space distributions is utilized to move into Z space. The value in Z space associated to the probability, $Val2$, is now calculated from the Z space global distribution.

$$Val3 = F^{-1}[Val2] \quad (4)$$

$$\text{or } Val3 = F^{-1}[G_{\{0,1\}}[G_{\{m,\sigma\}}^{-1}(q)]] \quad (5)$$

$Val3$ is the z -quantile of a distribution of uncertainty associated with the probability value q .

This procedure is repeated for a series of quantiles to characterize the full distribution. The mean and variance of the z space distribution is not known prior to characterization of the distribution and numerical approximation. This procedure is repeated for a range of y space means and variances. The result is a table of valid z space local distributions. An example CCDF table was generated with a bimodal z space distribution, and the y space discretized into 100 equally spaced entries in both mean and variance, see Figure 2.

This table is constructed by the subroutine PRECDIST.for. The flow chart for this subroutine is shown in Figure 9. Given the specified number of quantiles, means, and variances, and the range of means and variances in Gaussian space, and the global CCDF the CCDF table is constructed. The resulting quantiles are stored in a CCDF list array, and the associated z space mean and variance for each distribution are stored in separate arrays. A super block search (arbitrarily set to 20x20 for now) over the CCDF table is set

up to increase to speed of subsequently finding the nearest CCDF entry to a required z mean and variance.

The arrays required for the generation of the CCDF lookup table are dimensioned as allocatable in the DISTMODULE module and are then allocated in at the beginning of the PRECDIST.for subroutine. It is assumed that the PRECDIST.for subroutine will only be called once within the main program.

The series of transforms to move from y space to z space described above (see also Figure 10), is expanded in the flowchart in Figure 11.

The PRECDIST.for subroutine establishes the framework required for the rapid retrieval of valid local distributions of uncertainty. The GETCDIST.for subroutine retrieves a distribution from this table given a called mean and variance. This subroutine may be called any number of times after the PRECDIST.for subroutine has been executed. The flowchart of this subroutine is shown in Figure 11.

The tasks of the GETCDIST.for subroutine are divided into two main operations. The first is to find the CCDF table entry nearest to the called mean and variance. This is accomplished by finding the super block associated with the called mean and variance and then finding the closest entry within the block. If this super block has no entries then a search over all entries is performed. Although this global search requires more time, this situation will occur infrequently.

The second operation is to correct the retrieved distribution to have exactly the required mean and variance. This is achieved through the affine correction of each quantile. After affine correction each quantile is checked to ensure that it remains within the range of the global distribution. Values outside this range are truncated to the distribution minimum or maximum.

The parameters (m_z, σ_z^2) of the local CCDF's are calculated by block kriging, which requires mean block-to-block covariance values.

Mean Covariance Calculation

The mean covariance between volumes v_1 and v_2 is defined precisely as the average covariance value for a vector where the head describes v_1 and the tail describes v_2 .

$$\bar{C}(v_1, v_2) = \frac{1}{|v_1| |v_2|} \int \int_{v_1 v_2} C(u - u') du du' \quad (6)$$

The common practice for calculating the mean covariance between two volumes (v_1 and v_2), is to discretize the volumes and to average the point covariance over all possible combinations of discretized nodes between v_1 and v_2 .

$$\bar{C}(v_1, v_2) \cong \frac{1}{n \cdot n'} \sum_{i=1}^n \sum_{j=1}^{n'} \gamma(u_i - u_j) \quad (7)$$

It has been found that in 3D an acceptable number of discretizations is 4 in each direction (Journel and Huijbregts, pg.97, 1978). Thus, a single mean covariance between two three dimensional blocks requires $4^6 = 16,384$ calculations. Three discretizations require $3^6 = 729$ calculations for each mean covariance. This great number of calculations will increase CPU time. To keep run times under control either the number of discretizations (n and n') should be limited or the calculation of $C(\mathbf{u}-\mathbf{u}')$ should be sped up.

To speed up the covariance calculation the PREGBAR.for subroutine builds a variogram look up table for each variogram structure and for each variogram. Given sufficient discretizations of the variogram lag, these tabulated values are sufficient estimates of the point covariance. The flow chart for PREGBAR.for is shown in Figure 12.

This subroutine should be executed once prior to mean covariance calculations or geometric distance calculations. There are two components to this subroutine, the first builds the rotation matrices required to calculate geometric distances. These are standardized such that the ranges of all structures are set to 1. Then the standardized covariances are calculated and entered into the variogram table. The ranges of each structure are standardized to a range of 1.

All required global variables, such as the number of lag discretizations, and the variogram table array, are declared in the GAMMAMODULE.for module.

The CALCGBAR.for subroutine may be then executed any number of times to calculate the mean covariance between any two volumes. This subroutine requires the shape (extents and orientation) and separation of the two volumes and the variogram table and rotational matrices from the PREGBAR.for subroutine.

There are two options available for the calculation of the mean covariance; the first is the traditional discretization approach, mentioned above. The second is a novel approach of randomly sampling the point covariance between volumes v_1 and v_2 and assigning the sample mean as an approximation of the mean covariance. We know that the sample mean covariance will be unbiased, and for most circumstances a reasonable estimate may be acquired with as few as 30 samples. Thus this random sampling method results in a drastic decrease in the computational effort required to calculate the mean covariance. Preliminary results have shown the kriging matrices to be stable with these approximated mean covariances.

The CALCGBAR.for subroutine flow chart is shown in Figure 13. The procedure loops over discretized combinations or random locations $v_1(\mathbf{u})$ and $v_2(\mathbf{u}')$ and calculates the geometric distance between these locations, and then finds the point covariance between these locations. If the geometric distance is very close to 0 then the maximum covariance minus the nugget effect is assigned as the point covariance. A point covariance of the variance, when a nugget effect is present leads to a bias in the mean covariance called the zero effect (Journel and Huijbregts, pg.96, 1978).

To further reduce redundant calculations a nonstationary mean covariance table is built. The mean covariance between all volumes within the range of correlation of each other is

calculated and stored. During the construction of the kriging matrices the required mean covariances are found from this table.

Nonstationary Covariance Table

To further speed up the process of building kriging matrices based on mean covariances a nonstationary covariance table is introduced. The subroutine that builds this table is called NONSTATCOVTABLE.for. This algorithm screens the irregular grid for all volumes and data combinations within the range of correlation of each other. The mean covariance is calculated and stored in the nonstationary covariance table. During kriging, the mean covariances are called from the table, which avoids redundant calculations of the mean covariance. The flow chart for this subroutine is shown in Figure 14.

This table is constructed prior to simulation. The existing search subroutines for the purpose of build the kriging matrices are replaced by a table look up. A pointer array is assigned based on increasing geometric distance between the nodes or data points in the mean covariance table. By applying this pointer array the search only requires that the table is checked until maximum number of data or previously simulated nodes are found.

Search Considerations

While simulating irregular volumes, screening can cause extreme positive and negative weights. Xie, Y., Deutsch, C.V. and Tran, T. (2001) describe this phenomenon. Figure 3 demonstrates a variety of settings that result in extreme weights.

While these weights are legitimate solutions to the kriging matrix, they may result in negative kriging variances or too large kriging variances:

$$\sigma_k^2 = \bar{C}(v_o, v_o) - \sum_{\alpha=1}^n \lambda_{\alpha} \bar{C}(v_{\alpha}, v_o) \quad (8)$$

Given a positive definite covariance model is and correctly calculated mean covariances the resulting kriging variance are non-negative. The cause of these negative variances is precision limitations in the solution of the kriging matrix being magnified by large weights. Extreme weights are rare when simulating directly to a regular grid. An example is illustrated (see Figure 4) of a tartan simulation to model with equal volumes for each cell. A realization with the same variogram and reference histogram but with an irregular (tartan) grid is shown in Figure 5. This setting resulted in frequent extreme weights.

Two techniques to remediate this problem, template screening and iterative kriging, are now proposed and demonstrated. A third, octant search, is then discussed.

This technique is based on preemptively preventing screening by rejecting any conditioning data that are shadowed by a closer data. This is accomplished by creating a Boolean matrix representing the nodes within the range of correlation of the location being estimated. When a datum is selected to condition this unknown location (at the center of the Boolean matrix) a shadow template is positioned with the apex on the

datum. All locations within this template are set to 'false' and may not be subsequently selected as conditioning data for this estimate. Figure 6 shows an example of this matrix template.

This template technique was applied to the same setting as shown in Figure 5. These results were based on a shadow with a subtended angle of 60 degrees. There may be a need to further refine the shadow template, since some extreme weights remain. This technique is computationally expensive since many trigonometric operations are required to build each shadow. In addition, this method leads to a significant increase in the level of complexity of the local data search algorithm.

In the iterative kriging method the kriging weights are checked for anomalous values. If an absolute weight larger than a threshold (set to 2.0 in the following example) results, then the least correlated conditioning datum is removed and the kriging matrix is solved again. This process is iterated until all weights are within the acceptable interval. The results from this technique are shown in Figure 8.

This technique does not add significantly to the complexity of the current algorithm and is computationally cheap if anomalous weights are infrequent. Also, this technique is a self-healing method, since the correction is only applied when required, the level of correction is tailored to the need and there is no need for parameter tweaking.

There are some remaining negative variances that occur at locations with relatively high weights. These events do not seem to significantly impact the realization and they may be removed by further suppressing magnitude of the weights.

Another possible technique would be to apply the octant search method. This method is commonly applied in simulation (i.e. SGSIM from GSLIB) to prevent all conditioning data from originating from a single well or drill hole. In this setting, screening could be avoided by only accepting the nearest conditioning data from each octant. The rationale is similar to template screening. This technique may too greatly limit the available conditioning.

Conclusion

Key building blocks for the direct simulation on irregular grids have been included and documented in this work. These subroutines (1) calculate valid Z space distributions of uncertainty for all possible kriging estimates and estimation variances and (2) calculate mean covariances for irregular grids based on rectangular parallelepipeds.

Search issues which result in anomalous weights, out of place local realizations, and negative variances have been discussed and demonstrated. Three techniques, template screening of conditioning data, iteration of the kriging solution and octant search have been introduced. Future work will be carried out to quantify the computational cost of each technique and their impact on the resulting realizations.

References

- Deutsch, C.V., Tran, T.T., and Xie, Y-L. (2001). *An Approach to Ensure Histogram Reproduction in Direct Sequential Simulation*, Centre for Computational Geostatistics 3rd Annual Report, University of Alberta.
- Deutsch, C.V. and Journel A.G. (1998). *GSLIB: Geostatistical Software Library: and User's Guide*, Oxford University Press, New York, 2nd Ed.
- Journel, A.G. and Huijbregts, Ch.J. (1978). *Mining Geostatistics*, Academic Press, New York.
- Xie, Y., Deutsch, C.V. and Tran, T. (2001). *Preliminary Research Toward Direct Geostatistical Simulation of Unstructured Grids*, Centre for Computational Geostatistics 3rd Annual Report, University of Alberta.

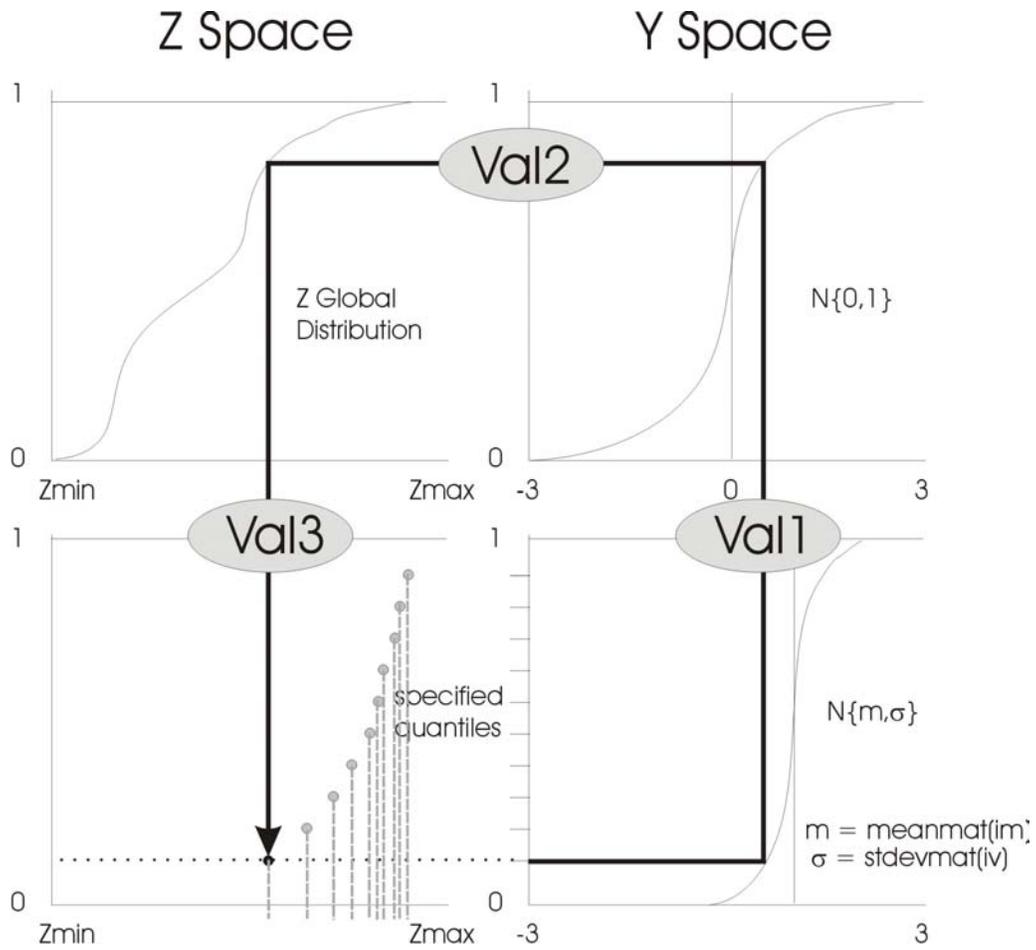


Figure 1: The graphical representation of the transformations applied to calculate the local distributions of uncertainty with a shape such that the global distribution is reproduced. The illustrated transformation is repeated for a sufficient number of quantiles to describe the local distribution.

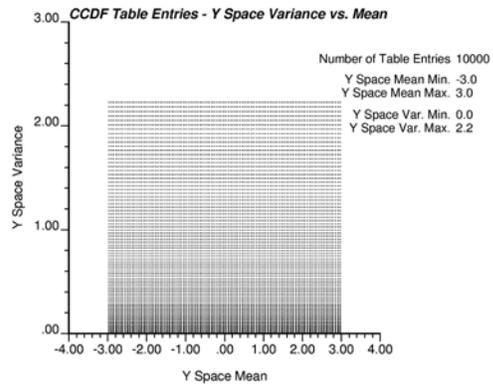
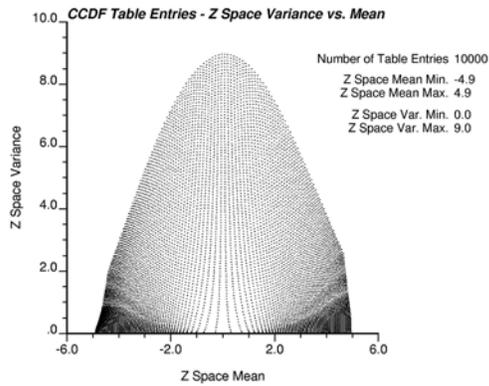
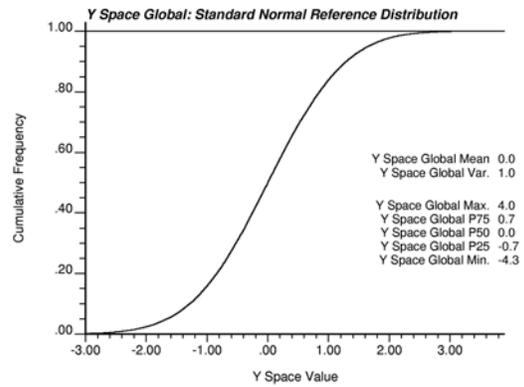
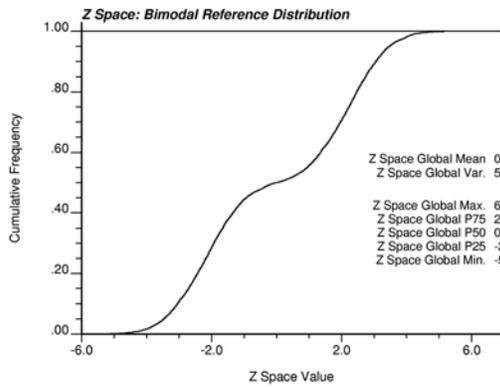


Figure 2: The global distributions applied to build a CCDF table. The Z space reference distribution is set as a bimodal distribution while the Y space is the required standard normal distribution. The regular series of means and variances for which local distributions were calculated are shown (lower right) along with the resulting means and variances of the Z space entries.

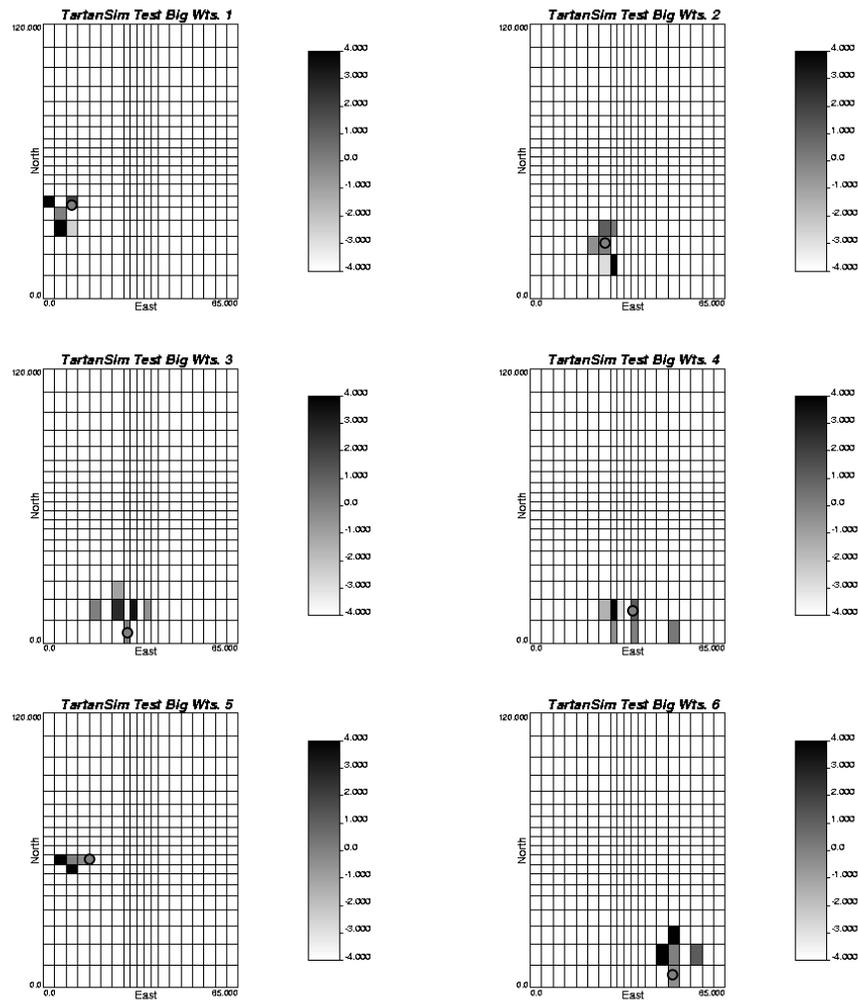


Figure 3: A variety of estimation settings, which resulted in extreme weights. The estimation location coded with a weight of 0.0 and a grey dot and the conditioning data (previously simulated nodes) are coloured according to their weights.

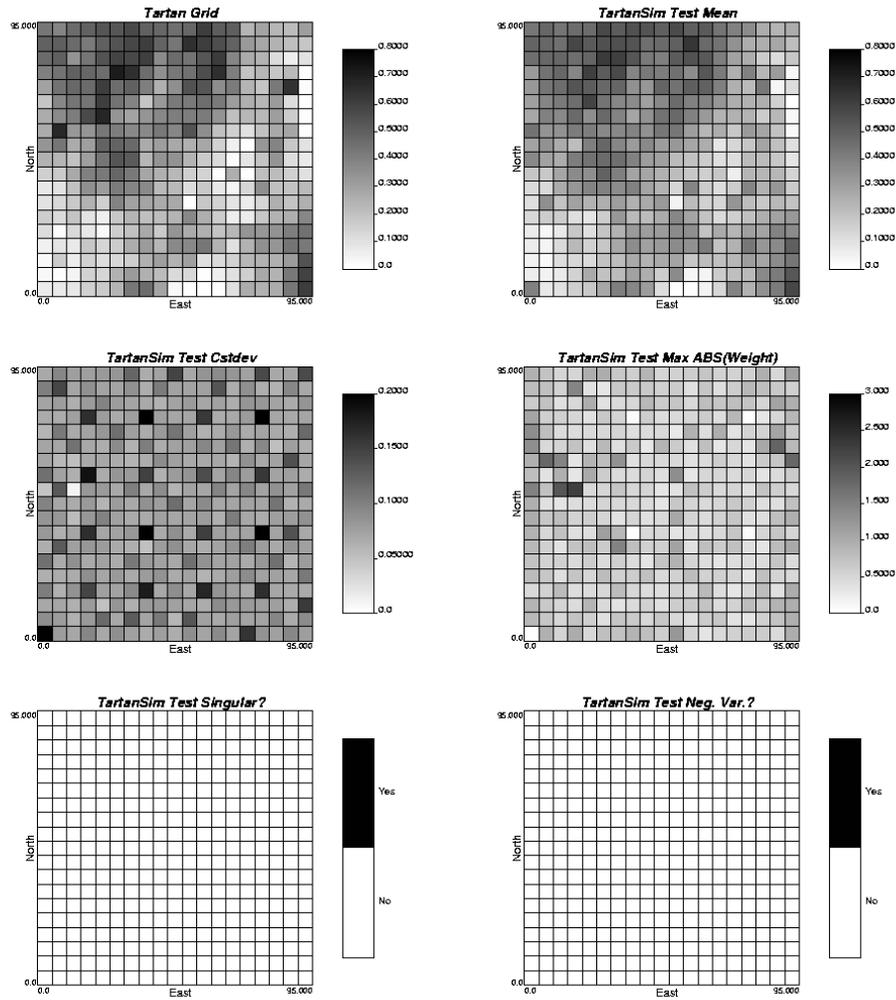


Figure 4: Tartansim on a regular grid with an isotropic variogram (spherical, range = 50) and a reference distribution. The simulated realization (top left), kriging estimates (top right), kriging standard deviation (middle left), maximum absolute kriging weight (middle right), map of singular matrices (bottom left), and map of negative variances (bottom right) are shown. The realization is well behaved (no out of place local realizations), and there are only a couple large weights present.

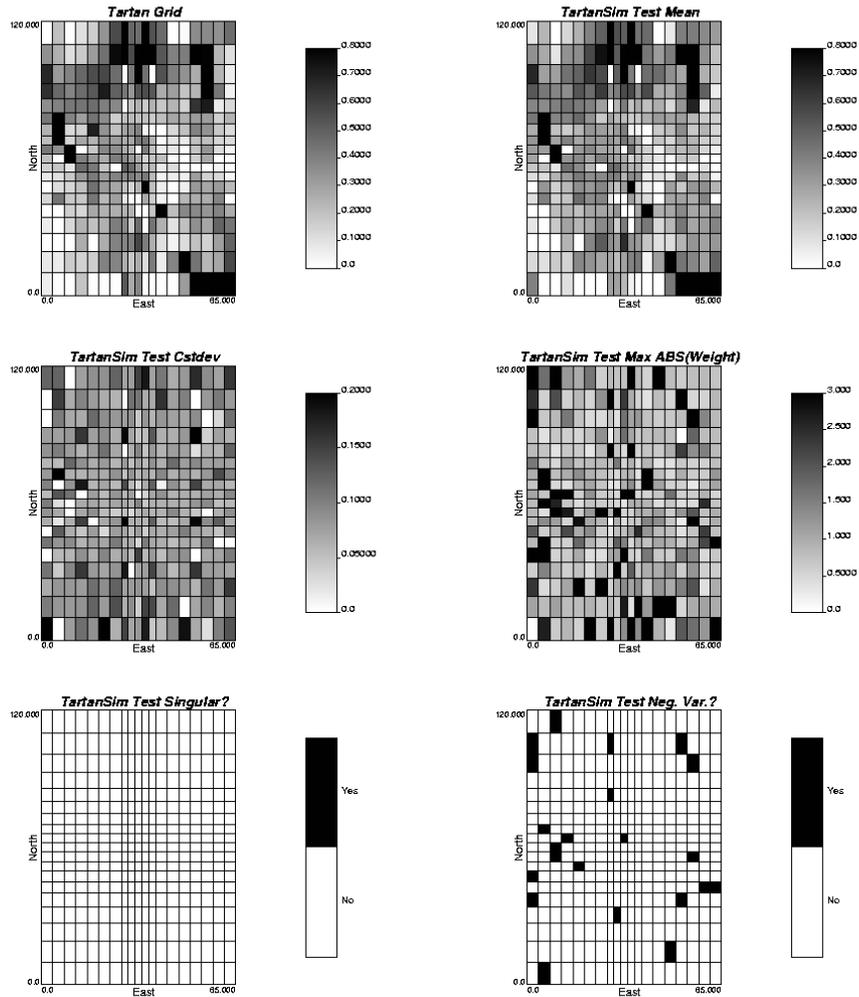


Figure 5: Tartansim on an irregular grid with the same variogram, random number seed and reference distribution applied in Figure 2. The simulated realization (top left), kriging estimates (top right), kriging standard deviation (middle left), maximum absolute kriging weight (middle right), map of singular matrices (bottom left), and map of negative variances (bottom right) are shown. All out of place local realizations and negative variances occur at locations with anomalous weights.

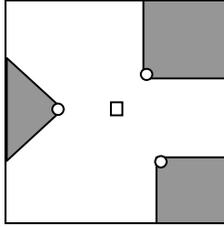


Figure 6: The Boolean matrix which filters screened data. This is a 2D example where three data have already been chosen as conditioning (circles). The shadow templates (grey areas) are located with their apexes on the chosen data and radiate away from the unknown location at the center square. These templates are like exaggerated shadows cast by the chosen data if the unknown location was a light source.

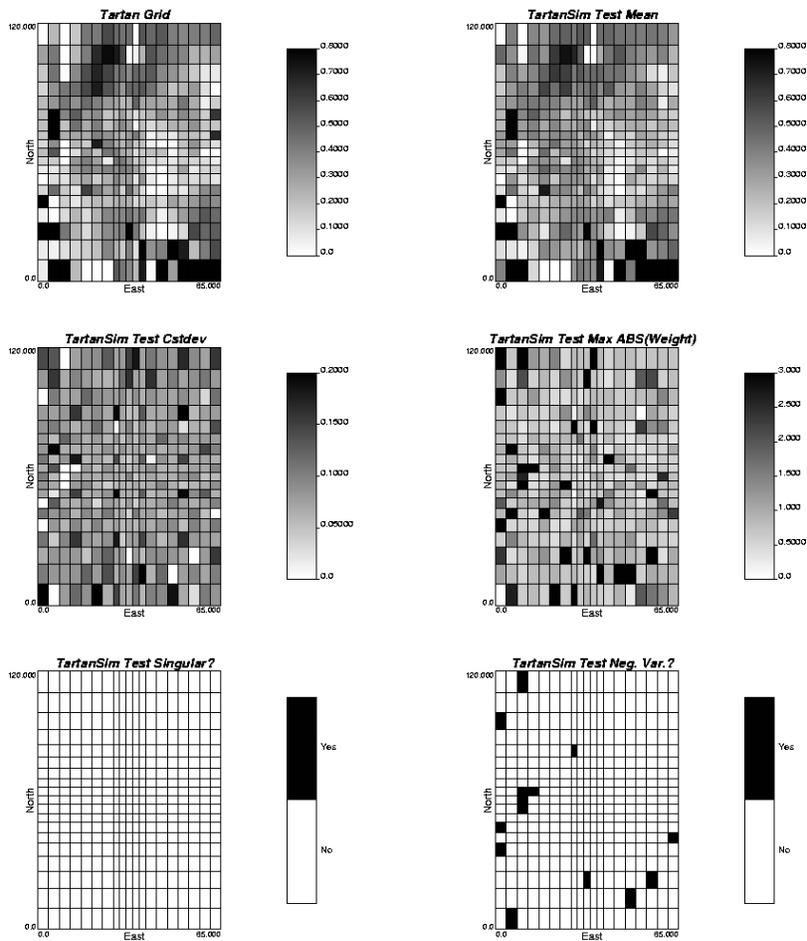


Figure 7: Tartansim on an irregular grid with the same variogram, reference distribution, random number seed and grid as applied in Figure 3 and with the conditioning template applied. The simulated realization (top left), kriging estimates (top right), kriging standard deviation (middle left), maximum absolute kriging weight (middle right), map of singular matrices (bottom left), and map of negative variances (bottom right) are shown. There is a reduction in the occurrence of anomalous weights, out of place local realizations, and negative variances.

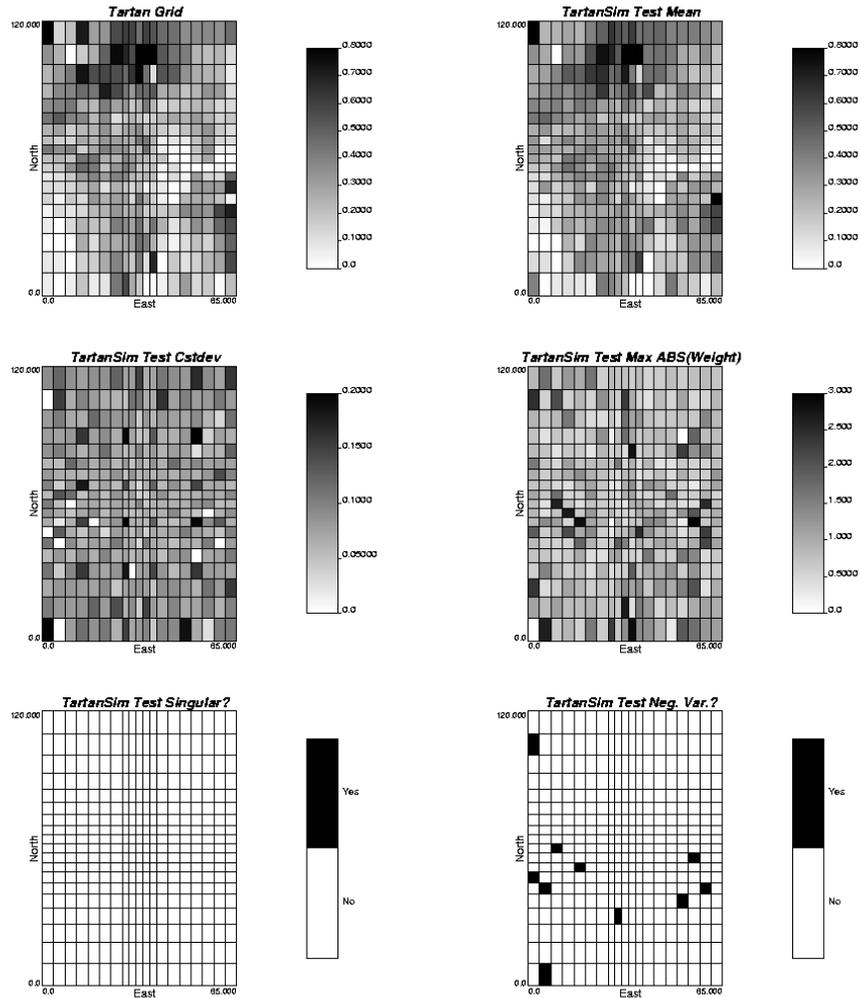
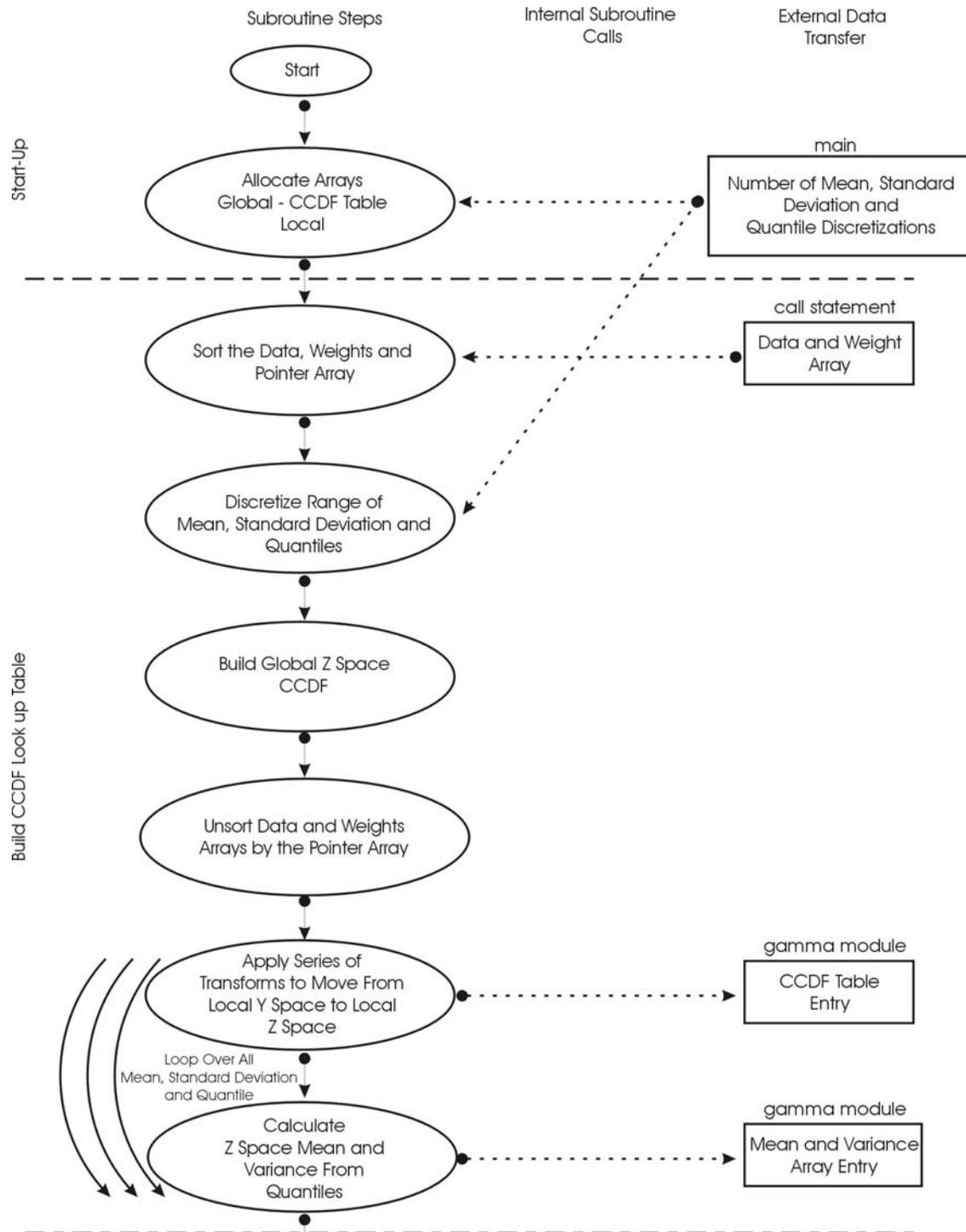


Figure 8: Tartansim on an irregular grid with the same variogram, reference distribution and grid as applied in Figure 3, and with the iterative method applied. The simulated realization (top left), kriging estimates (top right), kriging standard deviation (middle left), maximum absolute kriging weight (middle right), map of singular matrices (bottom left), and map of negative variances (bottom right) are shown. There is an elimination of anomalous weights, and out of place local realizations, and a reduction in negative variance occurrences.



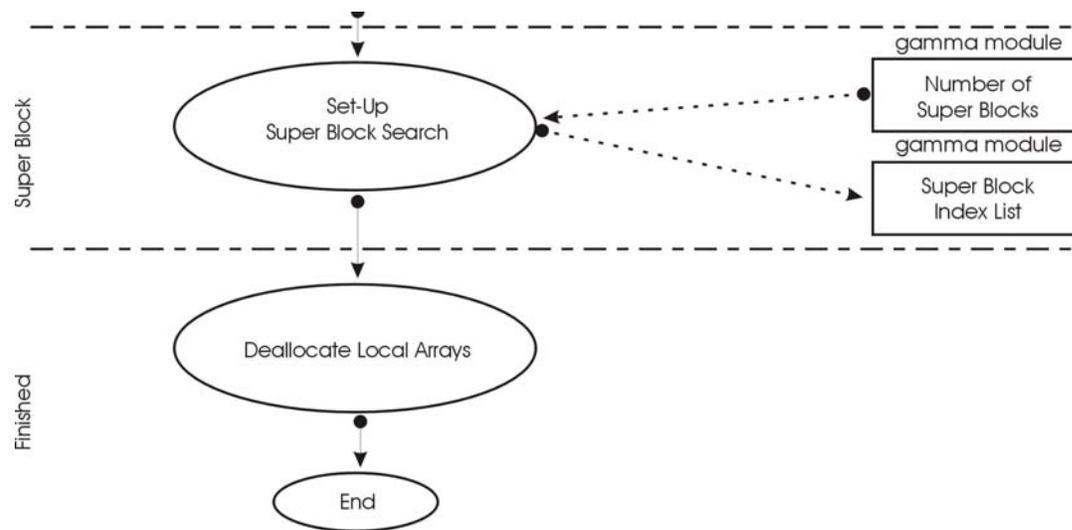


Figure 9: The main PRECDIST.for subroutine flow chart. This subroutine builds a table of local CCDF and then sets up a super block search to facilitate the locating of the nearest CCDF entry to a required mean and variance.

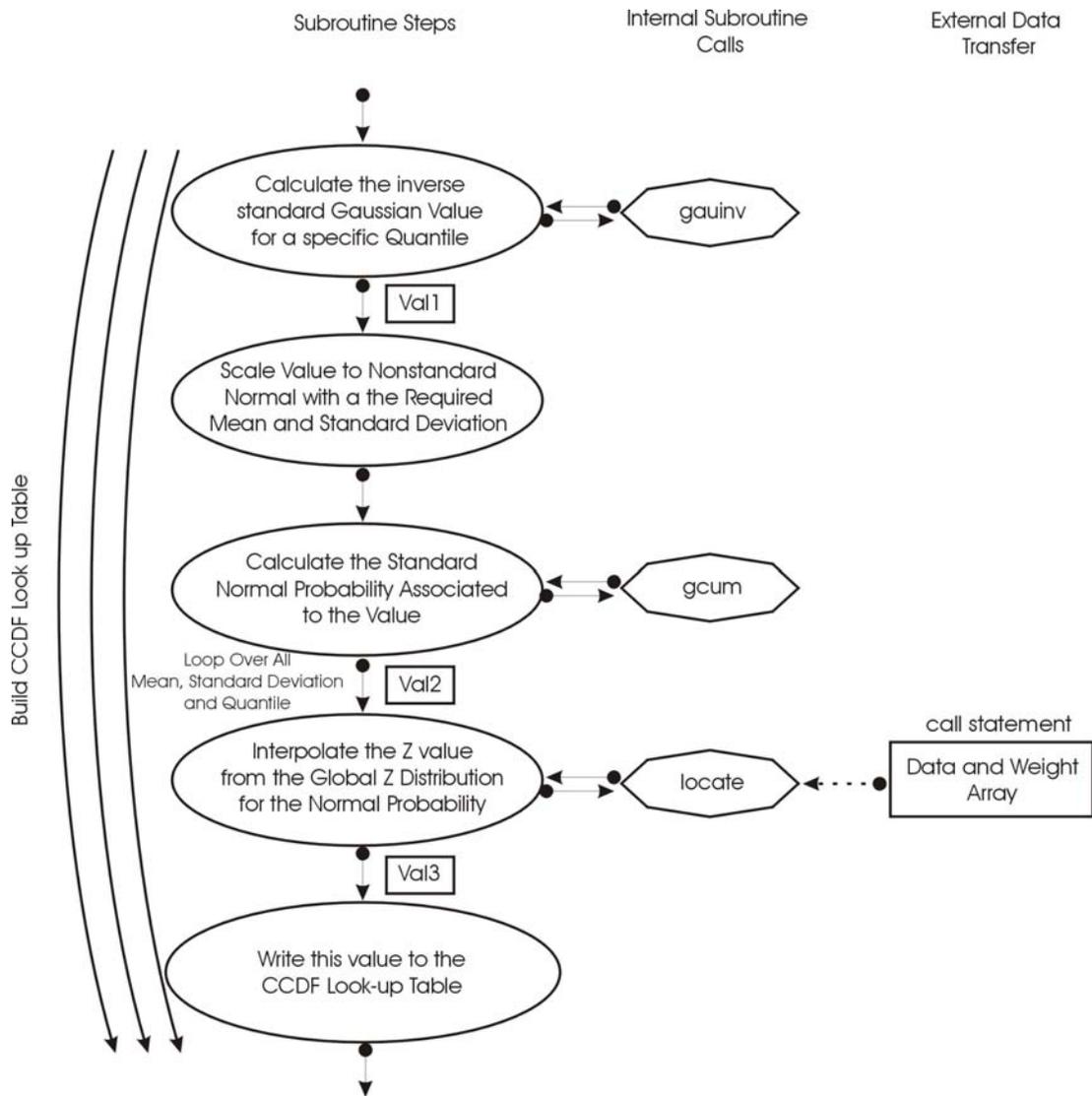


Figure 10: The transforms to move from Y space to Z space in the PRECDIST.for subroutine. These steps are illustrated graphically in Figure 1. Variables denoted in boxes are the same variables labeled in Figure 1.

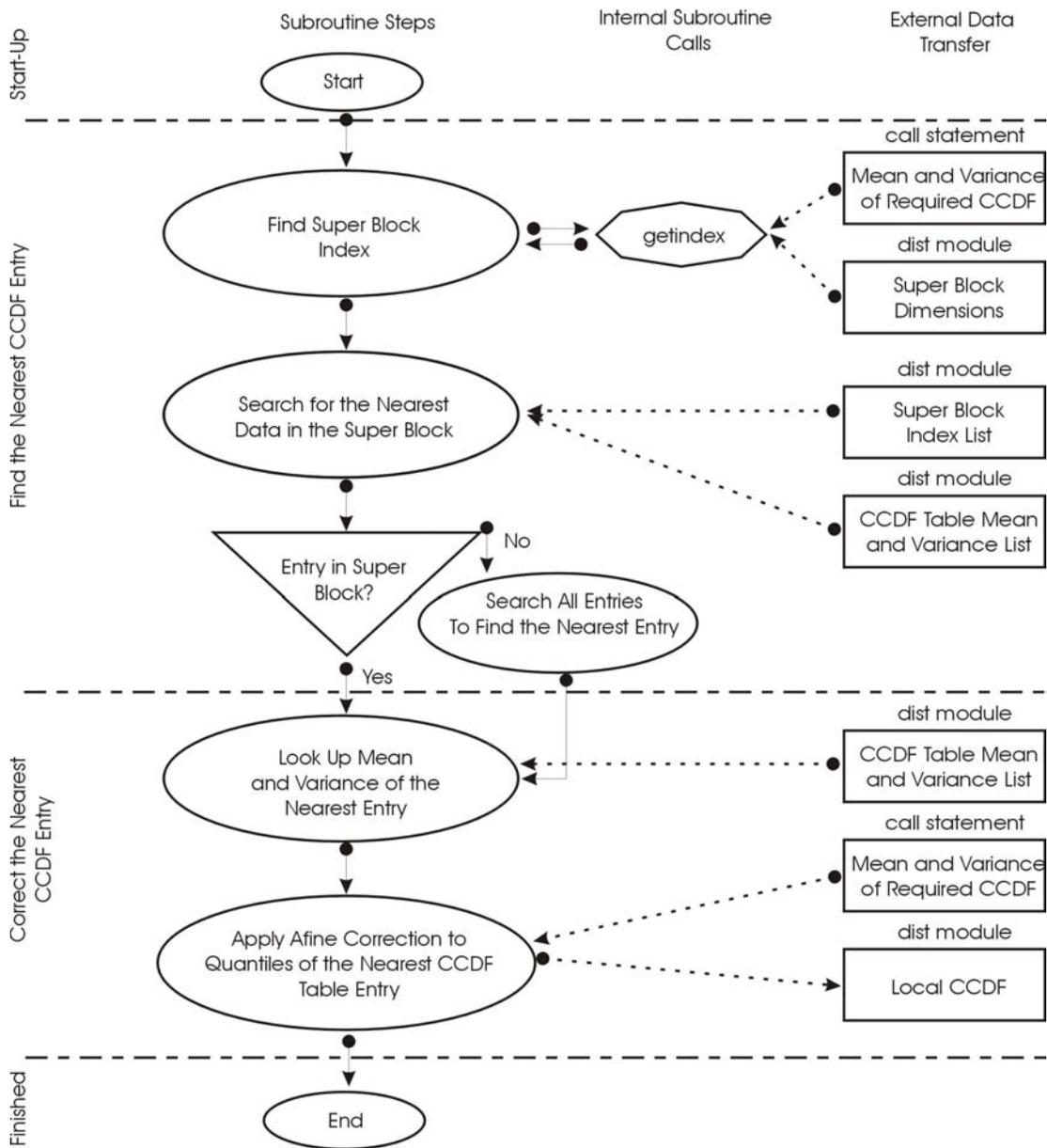


Figure 11: The Getcdist.for subroutine flow chart. Given a required mean and variance a super block search is used to find the nearest CCDF table entry. Affine correction is then applied to forced the returned CCDF to have the exact mean and variance required.

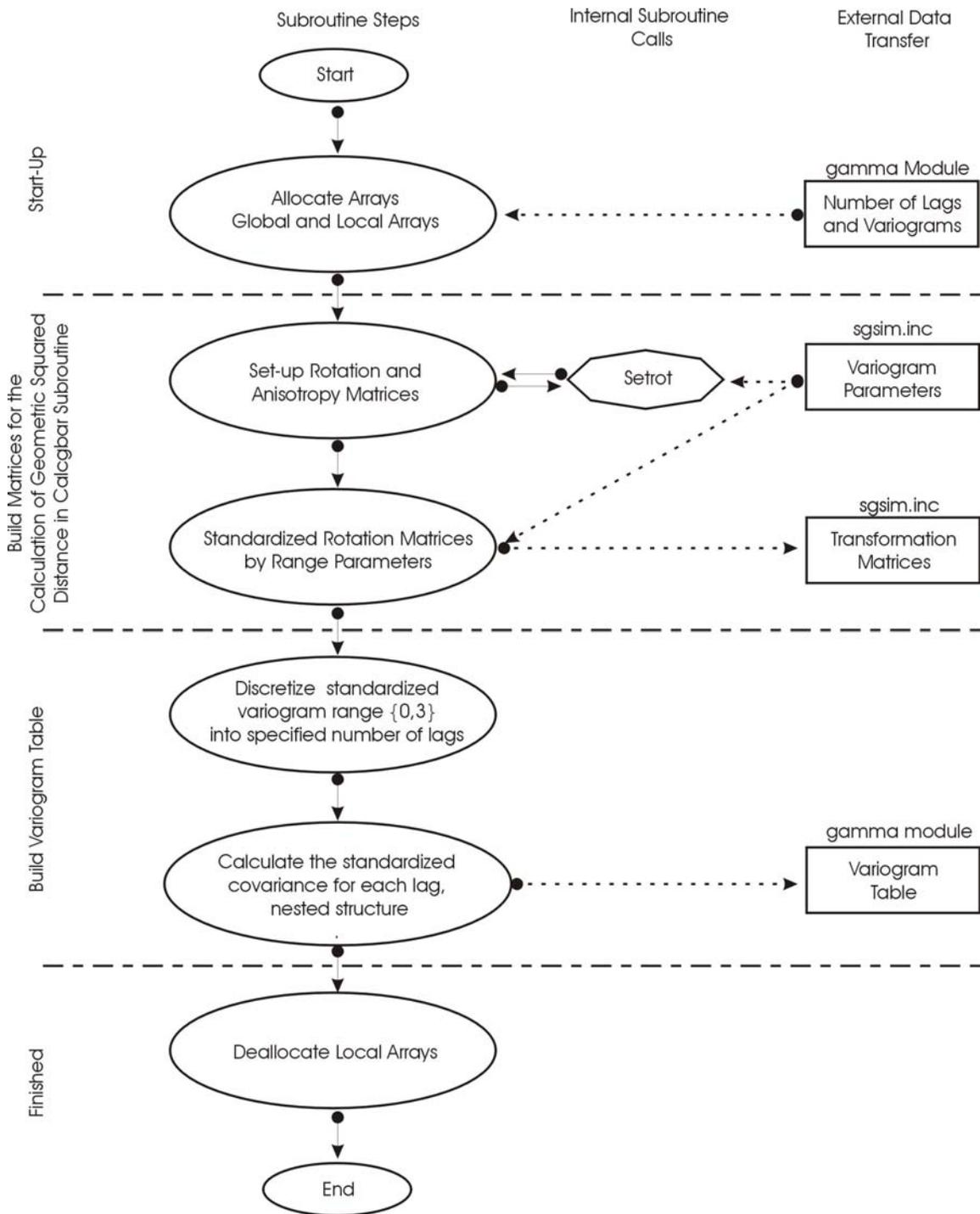
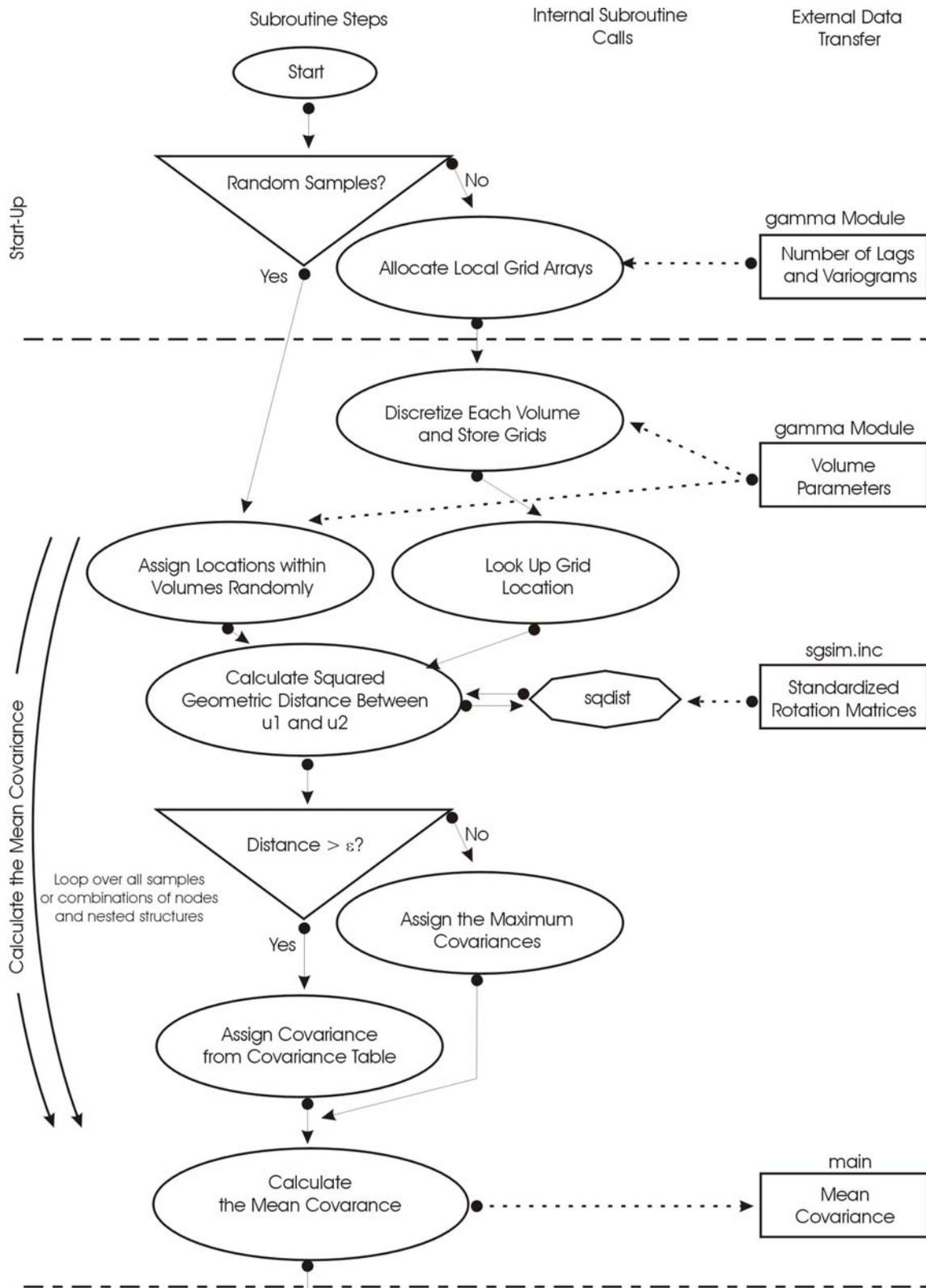


Figure 12: The Pregbar.for subroutine flow chart. This subroutine uses the variogram parameters to build standardized rotational matrices and then to build a variogram lookup table.



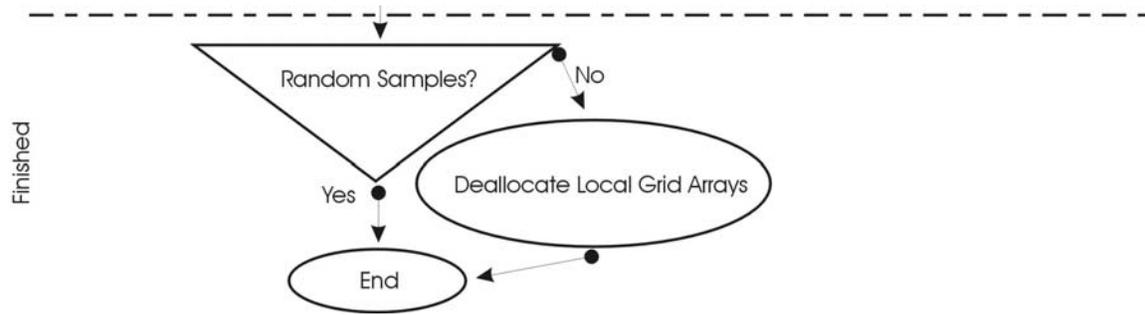


Figure 13: The calgbar subroutine flowchart. The rotational matrices and variogram lookup table calculated in PREGBAR.for are applied to calculate the mean covariance between two rectangular volumes. This is accomplished by averaging point covariances over discretized node combinations or random locations.

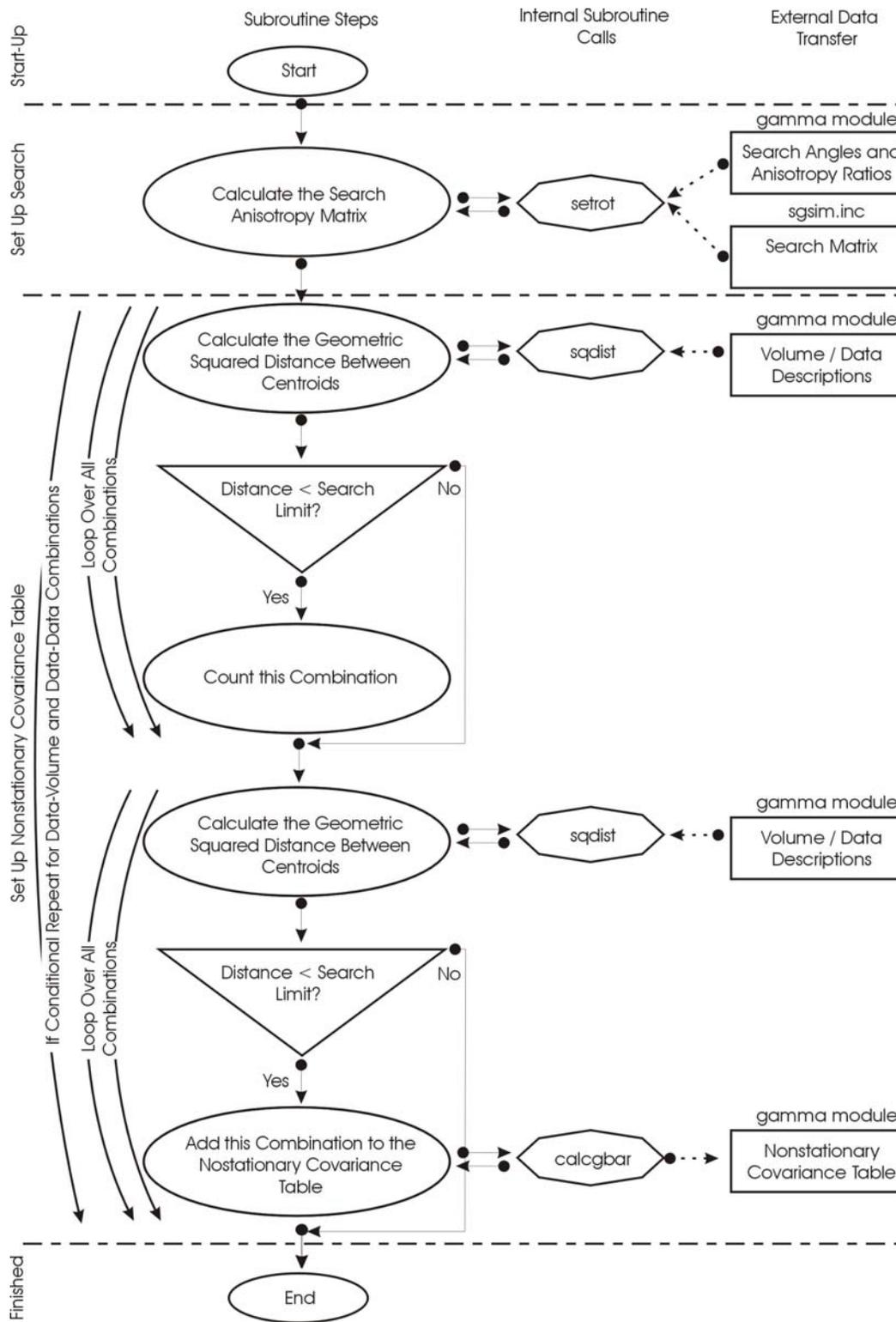


Figure 14: The nonstatcovtable subroutine flowchart. Mean covariances are stored for all combinations of data and volumes, which are correlated with each other.