MPESIM2: A Multiple-Point Event Simulation Program

Steve Lyster

The MPS-GS algorithm has been developed at the CCG for the last several years. The MPESIM2 program implements this algorithm using FORTRAN code in the style of GSLIB. This paper outlines the workflow used by MPESIM2, lists the parameters with explanations, and shows examples of other files used for simulation.

Introduction

Simulation of complex geologic structures is an area of ongoing research. Multiple-point statistics (MPS) is a term used to describe high-order spatial moments or algorithms that use these high-order moments. In the last few years a new algorithm has been developed at the CCG, called MPS-GS, that uses MPS in a Gibbs sampler framework (Lyster et al, 2006, Lyster, 2007, Lyster and Deutsch, 20008). The use of a Gibbs sampler for facies simulation was first proposed by Srivastava, 1992, and the idea of using MPS was briefly touched upon in that paper.

For research purposes, a program was written to implement MPS-GS; this program is called MPESIM2. The name is a reference to the Multiple-Point Events used to calculate conditional probabilities and the SIMulation purpose of the program. This program can be used to generate facies models with statistics derived from a training image. The program uses GSLIB-style input and output formats as well as a parameter file that is specified in the command prompt.

MPS-GS Algorithm

The MPS-GS algorithm is an iterative method for simulating facies. As put forward elsewhere, the MPS-GS algorithm proceeds as follows:

1. Start with a randomly-populated field on the coarsest grid
2. At a random location:
    a. Calculate the conditional probability density function (cPDF)
    b. Adjust the cPDF to account for secondary information and global facies proportions
    c. Apply noise reduction or cleaning and correct cPDF to sum to 1.0
    d. Draw a new facies value from the corrected cPDF
    e. Move to another unsampled location and repeat Step 2
3. After every location has been visited, check for convergence
    a. If there is no convergence yet, repeat Step 2 over all locations
    b. If convergence has been achieved, populate the next-coarsest grid
4. After the final grid has been simulated write out the results and start the next realization

The statistics used to calculate the cPDF in Step 2a are derived from a training image (TI); it is this part of the algorithm that uses MPS. The MPESIM2 program implements the MPS-GS algorithm using the following equation to estimate the conditional probability of each facies, *k*:

$$P^*(k) = \sum_{i,\alpha} \lambda_{i,\alpha}^k \left[I\left(E_i^\alpha\right) - P\left(E_i^\alpha\right) \right] + P(k) \quad (1)$$

Equation 1 is a linear estimate using non-linear indicators; E_i^a represents a discrete multiple-point event, or MPE. Each event i=1,…,M is defined by N points in space that are offset from the central point to be estimated; within each event there are a=1,…, K^N classes that define all of the possible combination of facies within the event. The optimal weights assigned to each indicator are found by solving the system of equations:

$$\sum_{j,\beta} \lambda_{j,\beta}^{k} \cdot Cov\left\{E_i^{\alpha}, E_j^{\beta}\right\} = Cov\left\{E_i^{\alpha}, k\right\} \tag{2}$$

where all of the covariances between MPEs are derived from the TI. This system of equations is very large and is solved by using the LAPACK (Linear Algebra PACKage) suite of subroutines (Anderson et al, 1999). The solution to the equation may be found as:

$$\begin{bmatrix} \\ \lambda \\ \\ \end{bmatrix} = \begin{bmatrix} \\ LHS \\ \\ \end{bmatrix}^{-1} \begin{bmatrix} \\ RHS \\ \\ \end{bmatrix} \tag{3}$$

where the LHS matrix is the covariances between the MPEs and the RHS is the covariance between the MPEs and the location being estimated. The LHS matrix is singular and very large; the maximum size is MK^N which can easily be in the thousands. Those MPEs with no occurrences in the TI, or very few, are ignored. Even so, with more than two or three facies the size of the MPEs is limited by this system of equations.

Solving the singular system does not lead to a unique solution. The Moore-Penrose pseudoinverse (Anderson et al, 1999) is used, as this is the solution that leads to the minimum norm and therefore avoids any extreme weights. The pseudoinverse is found by decomposing the LHS matrix into its eigenvalues and eigenvectors:

$$\begin{bmatrix} \\ LHS \\ \\ \end{bmatrix} = \begin{bmatrix} \\ V \\ \\ \end{bmatrix} \begin{bmatrix} \\ \omega \\ \\ \end{bmatrix} \begin{bmatrix} \\ V \\ \\ \end{bmatrix}^{T} \tag{4}$$

where V is the orthogonal matrix of eigenvectors and the diagonal of ω is the corresponding eigenvectors (several of which are zero due to the singularity of the matrix). The pseudoinverse is found by taking the reciprocal of the eigenvalues, leaving those that are zero as zero. The optimal weights are then:

$$\begin{bmatrix} \\ \lambda \\ \\ \end{bmatrix} = \begin{bmatrix} \\ V \\ \\ \end{bmatrix} \begin{bmatrix} \\ 1/\omega \\ \\ \end{bmatrix} \begin{bmatrix} \\ V \\ \\ \end{bmatrix}^{T} \begin{bmatrix} \\ RHS \\ \\ \end{bmatrix} \tag{5}$$

Because the optimal weights do not depend on the MPE indicator values, the optimal weights only need to be found once and then may be stored and used as necessary. The estimator in Equation 1 may be broken into two parts, beginning with the "base" estimates:

$$P^{+}\left(k\right) = -\sum_{i,\alpha} \lambda_{i,\alpha}^{k} \cdot P\left(E_i^{\alpha}\right) + P\left(k\right) \tag{6}$$

Equation 6 is the estimates for each facies assuming that all of the MPE indicators are zero (which will not be true, as one indicator per event must equal one). The "base" estimates are independent of the indicators and can be calculated for a given MPS template and TI without requiring any data input. Equation 6 can then be easily updated to account for local information:

$$P^{*}\left(k\right)=P^{+}\left(k\right)+\sum_{i,\alpha}\lambda_{i,\alpha}^{k}\cdot I\left(E_{i}^{\alpha}\right) \tag{7}$$

There are M indicators with a value of one and the rest are zero as per the exhaustive and exclusive properties of indicators. Equation 7 then only requires the addition of the base estimates and M weights. This addition is very fast; much faster than solving a system of equations or looking up values in a table or search tree.

The MPS template that defines the spatial offsets of the MPEs is calculated automatically by the `MPESIM2` program. The "optimal" template for a given TI is that which uses the lowest-entropy MPEs. The spatial entropy between an N-point event and the central point of the template is calculated as:

$$H_{i}=-\sum_{\alpha=1}^{K^{N}}\sum_{k=1}^{K}P\left(E_{i}^{\alpha}\cap k\right)\cdot\ln\left[P\left(E_{i}^{\alpha}\cap k\right)\right] \tag{8}$$

Minimizing this entropy should lead to the MPEs that convey the most information about the central location that contains the facies of interest, k.

Calculation of the optimal template and estimation weights is done automatically by the `MPESIM2` program. However, the program stops between each step to allow the user to inspect the output; template offsets should be looked at and the optimal weights checked for anomalous results before proceeding with simulation. The template and weights are stored in files that will be discussed later; once these files have been created the TI is no longer needed. Figure 1 shows a flowchart of this part of the workflow as implemented in `MPESIM2`.

Figure 2 shows a flowchart of the `MPESIM2` program for the data input/preparation steps; this is after the statistics have been calculated but before simulation commences. Both hard data (wells, drillholes, etc) and soft data (probability cube) may be used. Hard data are frozen in place and not visited in the random path; soft data are honoured by using a multiple servosystem approach as shown in Equation 9:

$$P'\left(k\right)=P^{*}\left(k\right)+\mu\cdot\left[P^{LOC}\left(k\right)-P^{SIM}\left(k,B_{k}\right)\right] \tag{9}$$

A servosystem is a typical approach to use for matching facies proportions in simulation (Liu, 2006). The value of μ is a control parameter and is usually set to a default value of 1.0. Using multiple servosystems, the global proportions of facies within the local bins, B_k, are matched. This reproduces the local proportions without giving them undue weight. The local bins are calculated as:

$$B_{k}=\text{int}\left(\frac{P^{LOC}\left(k\right)-P^{MIN}\left(k\right)}{P^{MAX}\left(k\right)-P^{MIN}\left(k\right)+\varepsilon}\cdot N_{B}\right)+1 \tag{10}$$

After the servosystem update in Equation 9 has been applied, the updated probabilities of the facies are modified to reduce the overall noise in the realization by adding a connectivity correction:

$$P''\left(k\right)=P'\left(k\right)+\eta\cdot\left[P\left(C_{k}\right)-E\left\{P\left(C_{k}\right)\,|\,P\left(C_{k}\right)>0\right\}\right] \tag{11}$$

In Equation 11 C_k is the connectivity of facies k; this is the number of adjacent cells that have the value k. $P(C_k)$ is the probability of the connectivity values and is calculated from the TI. This correction causes the short-range connectivity to be matched to that of the TI; very continuous facies have higher probabilities of large C_k values. The possible values of C_k range from 0 (no connectivity) to 7 (all six adjacent cells in 3D are facies k). The η parameter is a controlling value, usually set at about 0.1.

Figure 3 shows a flowchart of the steps followed by the `MPESIM2` program during simulation. When moving from one grid to the next, all cells in the current grid must be populated. The `MPESIM2` program does this by assigning the facies values randomly (for the first grid only) or by assigning the nearest value from the previous grid (for all finer multiple grids).

Parameter File

Several text files are used by the MPSEIM2 program. The first is a GSLIB-style parameter file (Deutsch and Journel, 1998). If no parameter file is found on the program execution then a default parameter file is created; Figure 4 shows a default parameter file. Line 1 is the name of the file containing the MPS; if this file exists it will be read in, if not it will be created (see Figure 1). Lines 2 through 4 define the MPS to use in simulation; the number of grids to use (*G*), the number of MPEs for the template (*M*), and the points per event (*N*). The *M* and *N* values must agree with the MPS file, and *G* must be no greater than that contained in the file (the simulation cannot use more grids than the MPS have been calculated for). Line 5 defines the minimum frequency of a MPE class to consider it in the system of equations (Equation 2). A higher value on Line 5 will reduce the size of the system to solve but will also ignore more MPEs.

Lines 6 to 8 define the TI file. Line 6 is the filename; Line 7 is the column containing the facies information; and Line 8 is the size of the TI field (X, Y, and Z). No TI file is needed if the MPS file (Line 1) already exists.

Lines 9 through 11 defines the template to calculate the MPS within, if necessary. Line 9 is the name of the file containing the template offsets; if this file exists it will be read, otherwise it will be created. Line 10 is the columns containing the X, Y, and Z offsets for each point in the template. Line 11 is the maximum offsets in the X, Y, and Z directions if a template is to be created. These lines are irrelevant is the MPS file already exists, but still must exist in the parameter file. If the simulation is 2D, the Z offset column may be set to 0.

Lines 12 through 14 define the indicators for the facies. Line 12 specifies the number of facies, *K*; Line 13 provides the integer codes defining the facies and must have *K* values; and Line 14 defines the global target proportions of the facies and must also have *K* values. If there are fewer than *K* parameters on Lines 13 or 14 the program may crash.

Lines 15 and 16 specify the hard conditioning data file. Line 15 is the file name, and Line 16 identifies the columns for the X, Y, Z, and facies information. If the data set is 2D the Z column may be set to 0. If the conditioning data file is not found then the realizations will be unconditional.

Lines 17 through 23 define the output. Line 17 is the name of the file to write the results to; this file will be in standard GSLIB format (Deutsch and Journel, 1998) with one column. Line 18 is the number of realizations to generate. Lines 19 to 21 are the grid definitions in the standard GSLIB format. Line 22 specifies the stopping criteria for the realizations, with a stopping number (typically left as 1 except for research); a minimum proportion of cell values to change in each loop (if less than, say, 0.1% of all values change the image is assumed to have converged); and the maximum number of loops to allow before stopping. Line 23 is the debugging output file; this file will provide a record of the output to the screen during simulation and will contain the parameters used, the number of changes in each loop, and the univariate proportions in each loop.

Lines 24 to 26 define the local proportions (or probability cube) information. Line 24 is the file name (if the file does not exist no local proportions will be used); Line 25 provides the *K* columns for the local data; and Line 26 specifies the discretization of the local PDFs or number of bins B_k in Equations 9 and 10. If local PDF data exists then the global proportions are not used.

Line 27 specifies the servosystem and connectivity correction control parameters. Line 28 is the random number seed value.

Template and MPS Files

Using the TI and the defined parameters, the MPESIM2 program will calculate the optimal MPS template and weights used in Equation 1. These two calculations are stored in files for easy inspection by the user and for use later without having to be recalculated.

Figure 5 shows an example of a template file. For this template, a simple 2D TI was used. Line 1 defines the parameters for the template: *G*, *M*, and *N*. Line 1 is also the GSLIB-style header line for the file. Line 2 is the number of columns in the file; if the template is created by MPESIM2 this is always 3. The Lines numbered 3 in Figure 5 are the header lines and there is one for each column in the file; in this case, one

each for the X, Y, and Z offsets. The Lines numbered as 4 defined the offsets in the template from the central location of (0,0,0). There are *GMN* lines in the file, with *N* cycling the fastest, then *M*, then *G*, so the first *N* lines define the first MPE in the first grid and the *(MN+1)* to *(MN+N)* lines define the first MPE in the second grid. In the example template file two grids were used with four, four-point MPEs in each grid for a template size of 16 points and 32 lines in the file.

Figure 6 shows an example of an MPS file. The dots "…" represent lines that have been removed to shorten the file to a single page for this example. Line 1 is the header line; note the MPS file is no longer GSLIB-standard, but is unique to the MPESIM2 program. Line 2 contains the parameters for the MPS in the file, with four numbers: *G*, *M*, *N*, and *K* (these letters are shown in order on Line 1). These parameters must agree with the parameter file used, though grids may be dropped from MPS calculation to simulation.

The Lines labeled as 3 are the headers for each grid and there are *G* of these lines. After the grid header there is a Line 4 (again one for each grid), which defines the global proportions for this grid, P_k, k=1,…,K. Note that these numbers vary slightly from grid to grid due to the range of the search for each grid; these very small fluctuations can have a significant impact on the multiple-point covariance matrix. After the Line 4 there is a group of Lines, labeled 5 here, for each grid. There are seven lines and they define the connectivity values $P(C_k)$ k=1,…,K; each line defines one connectivity value from C_k=0 to C_k=6. These are the connectivity values as used in Equation 11.

After the grid header lines, there is the header for the first MPE on that grid. Line 6 is the header for each MPE, and there are *M* of these lines for each grid. After Line 6 is the group of Lines 7 that specify the point offsets for the MPE. There are *N* of these lines for each event. The set of Lines 8 are the statistics calculated for the indicators in the MPE. Each line specifies the class α, the global proportion of that class, $P(E_i^a)$, and weight assigned to the indicator of that class for each facies, $\lambda^k_{i,a}$, k=1,…,K. There are a maximum of K^N of these lines for each event, one for every possible class. Classes that are below the cutoff proportion in the parameter file are not included.

The Lines 6 through 8 are repeated *M* times for each grid. At the end of each grid are the weights assigned to univariate indicators. MPESIM2 incorporates single-point indicators into the estimation; this is shown in Equation 12, which is Equation 1 with an additional summation.

$$P^*(k) = \sum_{i,\alpha} \lambda^k_{i,\alpha} \cdot \left[I\left(E_i^\alpha\right) - P\left(E_i^\alpha\right) \right] + \sum_{u,k'} \chi^k_{u,k'} \cdot \left[I\left(k';u\right) - P\left(k'\right) \right] + P(k) \qquad (12)$$

These univariate indicators are redundant with the MPEs and do not change the estimates if all of the MPEs are informed; however, for large events or 4+ facies many MPEs are not found in the TI and using univariate indicators speeds up convergence of the algorithm. Line 9 is a header for the univariate indicators; each of the Lines labeled 10 contain the X, Y, and Z offset of the indicator, the facies value at that offset, and the *K* weights associated with that indicator. There are *MNK* lines in this part of the MPS file.

After the univariate indicators portion of the file, Lines 3 through 10 repeat again for each of the remaining grids. Line 11 defines the end of the MPS file.

FORTRAN Files

There are 81 files of FORTRAN code needed to compile the MPESIM2 program. The majority of these are from the LAPACK suite of subroutines (Anderson et al, 1999). A full list of these files is shown in Figure 7. There is the main file, `mpesim2.f`; a group of GSLIB and similarly-styled files; and a variety of LAPACK and BLAS (Basic Linear Algebra Subprograms) subroutines.

Comments

The results of the MPESIM2 program are robust with respect to the parameters; typically good results can be obtained by using 4-5 grids, and 8-12 four-point MPEs or 4-8 eight-point MPEs (Lines 2-4). The minimum MP frequency (Line 5) should be adjusted based on the size of the TI and number of points and facies to be used in the simulation; typically a frequency of 10^{-5} (one in 100,000) is suitable for large TIs,

though this may be adjusted to reduce the size of the matrix to invert. A servosystem parameter of 1.0 and connectivity factor of 0.1 normally gives acceptable results (Line 27). A stopping number of 1, threshold of 0.01 and maximum number of loops of 30 is also a good starting point for simulation (Line 22).

Currently the maximum size of the MPEs is limited by the memory requirements, as the entire LHS matrix of Equation 2 is initialized even for the classes that are not used. This simplifies the programming but drastically increases the RAM needed to run the program for more than two facies. The program runs best using a TI that is several times the size of the largest structure to be reproduced. For long-range channel connectivity this means having a TI that is several times the size of the simulation grid. Using TIs with more than 3-4 facies does not always give good results; in these cases grouping facies and simulating hierarchically would be necessary.

The `MPESIM2` program was created primarily as research code for implementing and testing the MPS-GS algorithm. As such, there are a number of improvements that could be made in the code. The program is fast enough to be used for large geomodels but could be optimized further. Solving the system of equations uses a set of very efficient subroutines, but the calculation of covariances from the TI could be improved.

References

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. (1999) *LAPACK User's Guide, 3rd Ed.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 429 p.

Deutsch, C.V. and Journel, A.G. (1998) *GSLIB: Geostatistical Software Library and User's Guide, 2nd Ed*. Oxford University Press, New York, 369 p.

Liu, Y. (2006) Using the Snesim Program for Multiple-Point Statistical Simulation. *Computers & Geosciences*, Vol. 32, No. 10, Dec. 2006, pp 1544-1563.

Lyster, S. (2007) A Second Generation Gibbs Sampler for Simulation with Multiple-Point Statistics. *Centre for Computational Geostatistics*, No. 9, 26 p.

Lyster, S., and Deutsch, C.V. (2006) Short Note: An Entropy-Based Approach to Establish MPS Templates. *Centre for Computational Geostatistics*, No. 8, 10 p.

Lyster, S. and Deutsch, C.V. (2008) MPS Simulation in a Gibbs Sampler Algorithm. *8th International Geostatistics Congress*, 10 p.

Lyster, S., Deutsch, C.V., and Dose, T. (2006) A New MPS Simulation Algorithm Based on Gibbs Sampling. *Centre for Computational Geostatistics*, No. 8, 16 p.

Srivastava, M. (1992) Iterative Methods for Spatial Simulation. *Stanford Center for Reservoir Forecasting*, No. 5, 24 p.

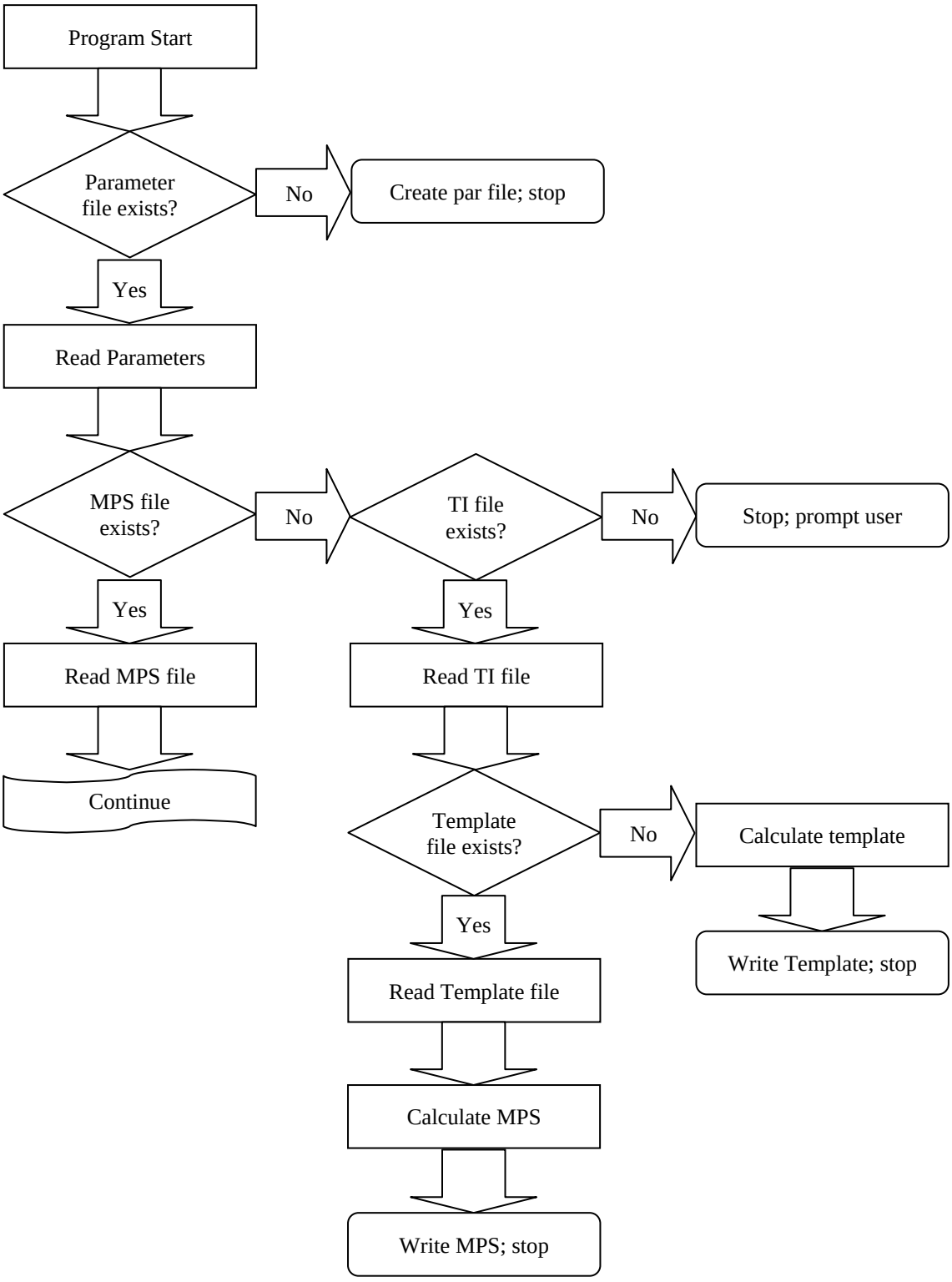

Figure 1: Workflow of the MPESIM2 program for statistical calculation.

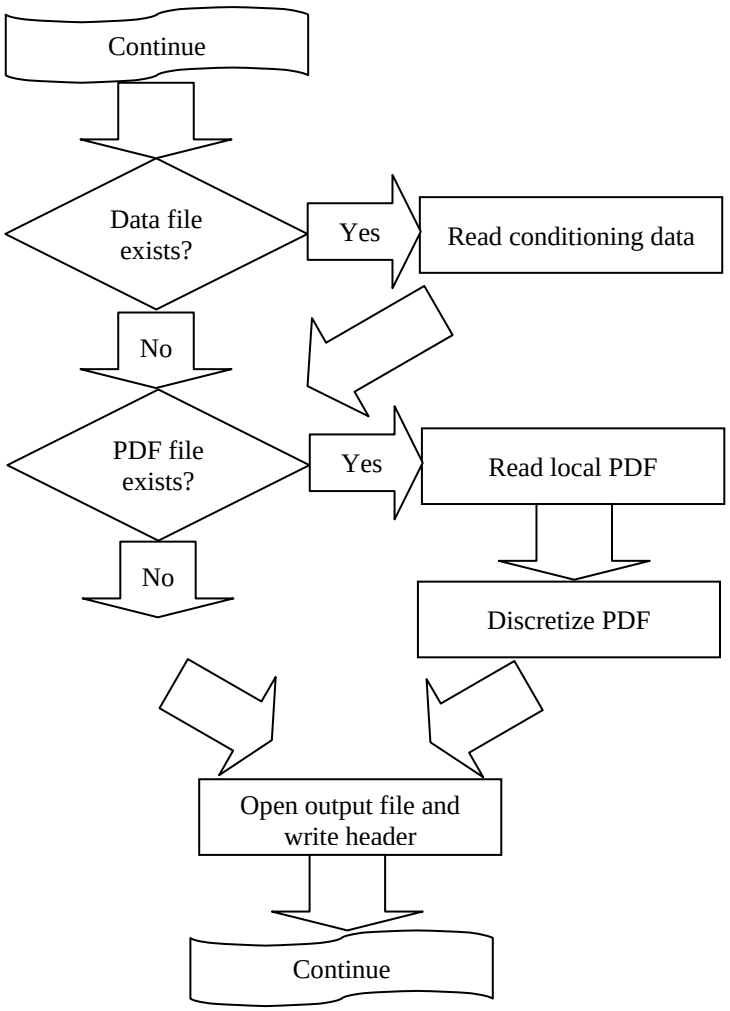

Figure 2: Workflow of the MPESIM2 program for data input and preparation to simulate.

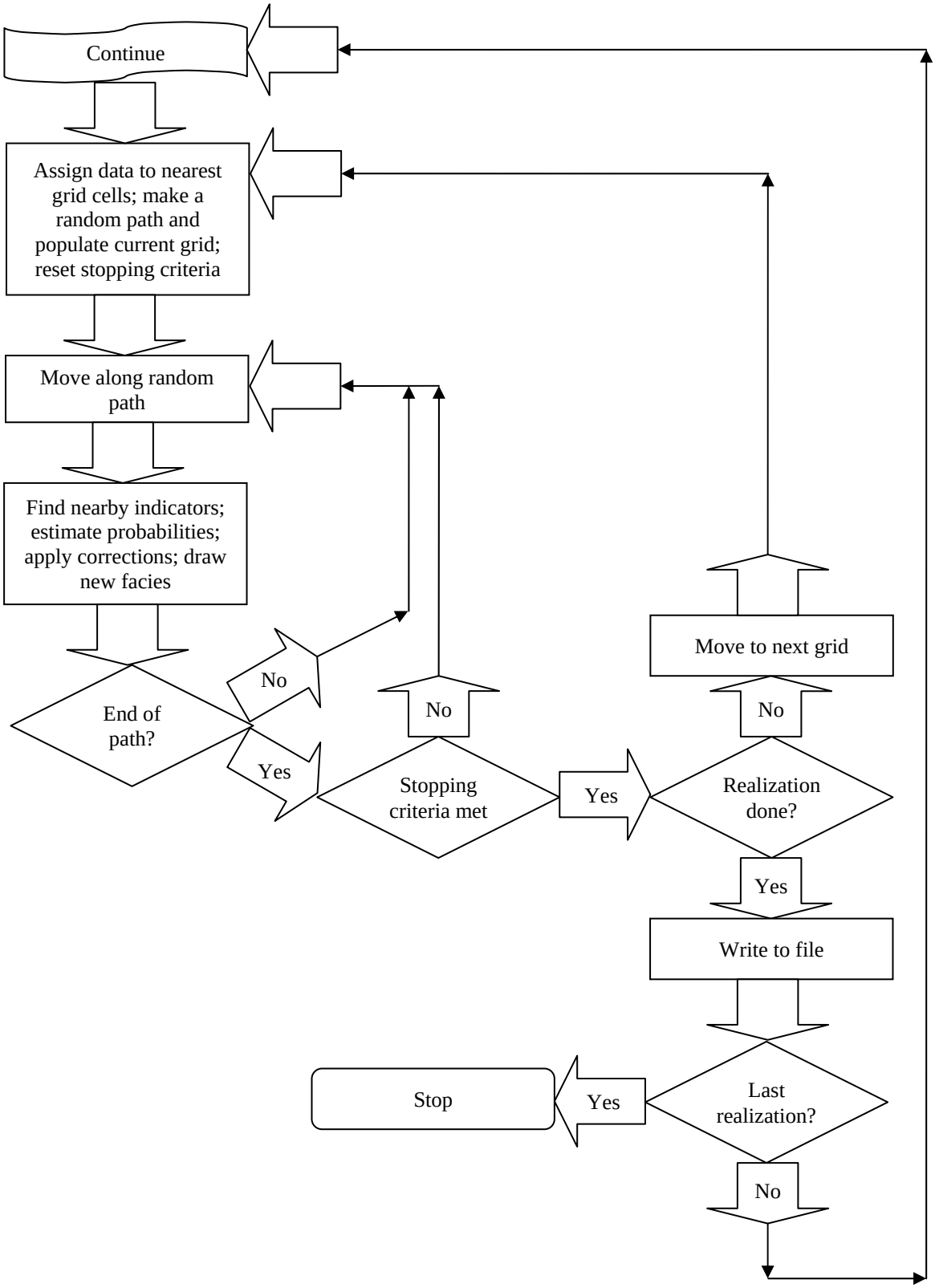

Figure 3: Workflow of the MPESIM2 program for simulation.

```
START OF PARAMETERS:
mpstats.mps                    -File for MPS
4                                 -Number of grids to use
8                                 -Number of MP events to consider
4                                 -Number of points per event
1.0e-6                            -Minimum MP frequency to consider
tifile.dat                     -Training Image file
1                                 -Column for TI data
256  256  128                     -Size of TI field
template.temp                  -File containing point offsets
1  2  3                           -Columns for X,Y,Z offsets
5  5  3                           -Maximum point offsets
5                              -Number of codes/facies
0  1  2  4  5                     -Indicators for codes/facies
0.4  0.2  0.2  0.1  0.1           -Global pdf values
data.dat                       -Conditioning data file
1  2  3  4                        -X,Y,Z,Data columns
mpesim2.out                    -File for output
1                                 -Number of realizations
256  0.5  1.0                     -nx,xmin,xsiz
256  0.5  1.0                     -ny,ymin,ysiz
128  0.5  1.0                     -nz,zmin,zsiz
1  0.05  30                       -Stopping number, threshold, max loops
mpesim2.dbg                       -File for debugging
pdf.dat                        -Local pdf file
1  2  3  4  5                     -Columns for all indicators
4                                 -Discretization of local pdf
1.0  0.1                       -Servosystem, connectivity factors
6744889                        -Random number seed value
```

Figure 4: Default parameter file for the MPESIM2 program.

```
Template of points G M N            2           4           4
 3
 X offset
 Y offset
 Z offset
   1    0    0
  -1    0    0
   2    0    0
  -2    0    0
   0    1    0
   0   -1    0
   1   -1    0
  -1    1    0
   1    1    0
  -1   -1    0
   2    1    0
  -2   -1    0
   2   -1    0
  -2    1    0
   1   -2    0
  -1    2    0
   2    0    0
  -2    0    0
   4    0    0
  -4    0    0
.
.
.
```

Figure 5: Example template file for the MPESIM2 program.

```
 1  MPS file for MPESIM v2  G  M  N  K
 2    2  4  4  2
 3   GRID            1
 4      0.696324    0.303676
 5      0.000187    0.000000
        0.002355    0.001886
        0.040112    0.087176
        0.127551    0.313561
        0.829794    0.597377
        0.000000    0.000000
        0.000000    0.000000
 6   EVENT             1
 7       1     0     0
        -1     0     0
         2     0     0
        -2     0     0
 8       1    0.642467   -0.334973    0.334973
         5    0.012859   -0.258374    0.258374
         6    0.025718   -0.073913    0.073913
    ...
 6  EVENT             2
 7       0     1     0
         0    -1     0
         1    -1     0
        -1     1     0
 8       1    0.565051   -0.250738    0.250738
         2    0.006898   -0.231665    0.231665
    ...
    ...
 9  UNIVARIATE INDICATORS
10       1     0     0  0    0.328335   -0.328335
         1     0     0  1   -0.328335    0.328335
        -1     0     0  0    0.340872   -0.340872
        -1     0     0  1   -0.340872    0.340872
    ...
 3  GRID            2
 4      0.691189    0.308811
 5      0.000628    0.000000
        0.008791    0.004129
        0.081122    0.163299
        0.237284    0.618412
        0.672174    0.214160
        0.000000    0.000000
        0.000000    0.000000
 6   EVENT             1
 7       2     0     0
        -2     0     0
         4     0     0
        -4     0     0
 8       1    0.589871    0.062044   -0.062044
         2    0.001790   -0.055254    0.055254
         3    0.001872   -0.074566    0.074566
    ...
    ...
 9  UNIVARIATE INDICATORS
10       2     0     0  0    0.205283   -0.205283
         2     0     0  1   -0.205283    0.205283
        -2     0     0  0    0.189959   -0.189959
        -2     0     0  1   -0.189959    0.189959
    ...
11  END
```

Figure 6: Example MPS file for the MPESIM2 program.

Main File
mpesim2.f
GSLIB and Similar Files
chknam.for getclass.for makeevents.for readstats.for writeevents.for
convmat.for getcodes.for makepar.for readtemp.for writetemp.for
convtemp.for getconn.for mpstatcalc.for readti.for
dsortem.for getindx.for pickem.for sortem.for
LAPACK Subroutine Files
dlacpy.f dlaed4.f dlaeda.f dlarf.f dlaset.f dorm2r.f dsterf.f
dlae2.f dlaed5.f dlaev2.f dlarfb.f dlasr.f dormql.f dsyevd.f
dlaed0.f dlaed6.f dlamrg.f dlarfg.f dlasrt.f dormqr.f dsytd2.f
dlaed1.f dlaed7.f dlanst.f dlarft.f dlassq.f dormtr.f dsytrd.f
dlaed2.f dlaed8.f dlansy.f dlartg.f dlatrd.f dstedc.f
dlaed3.f dlaed9.f dlapy2.f dlascl.f dorm2l.f dsteqr.f
LAPACK Auxiliary Files
dlamch.f ieeeck.f ilaenv.f iparmq.f lsame.f xerbla.f
BLAS1 Files
daxpy.f ddot.f drot.f dswap.f idamax.f
dcopy.f dnrm2.f dscal.f
BLAS2 Files
dgemv.f dger.f dsymv.f dsyr2.f dtrmv.f
BLAS3 Files
dgemm.f dsyr2k.f dtrmm.f

Figure 7: List of FORTRAN files required to compile the MPESIM2 program.