EML-610

A Users' Guide for the Computer Program ISDMAP: Analysis and Mapping of *In-Situ* Gamma-Ray Spectrometry and Soil Sample Data

Marcel Reginatto, Paul Bailey, and Peter Shebell

Environmental Measurements Laboratory
U.S. Department of Energy
201 Varick Street, 5th Floor
New York, NY 10014-4811

November 2000

DISCLAIMER

"This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof."

This report has been reproduced directly from the best available copy.

Available to DOE and DOE Contractors from the Office of Scientific and Technical Information, P. O. Box 62, Oak Ridge, TN 37831; prices available from (423) 576-8401.

Available to the public from the U.S. Department of Commerce, Technology Administration, National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161, (703) 487-4650.

ABSTRACT

The computer program ISDMAP was written to analyze data from a set of *in situ* gamma-ray spectrometry measurements on a grid. It can also do a combined analysis of this type of data and data from soil samples. One well-known difficulty with attempting this type of analysis is that such sets of data can never provide enough information to determine a *unique* solution. This can be understood intuitively since a finite number of measurements can not be sufficient to determine a continuous distribution (this observation is not restricted to data collected with the *in situ* technique, but holds for any set of discrete measurements, such as a series of soil samples). One can, however, restrict to particular *types* of solutions by requiring that they satisfy other conditions in addition to the measurements, and this is the approach taken by ISDMAP. In ISDMAP, the data are analyzed in a different manner depending on whether the data is from a characterization survey or from a post-remediation survey. The “characterization” option creates a map of contamination in surface soil that is smooth and fits the data. The “post-remediation” option creates a map with a summary of potential hot spots over a constant background level, providing a map of hot spots that might be “hidden” in the data. This report describes the operation of ISDMAP in sufficient detail to allow a user to prepare the necessary input files and run the program. The program requires a PC with DOS or a DOS emulator (most Windows machines have this).

TABLE OF CONTENTS

INTRODUCTION

In the *in situ* gamma-ray spectrometry technique, a detector is positioned above the ground facing downwards to measure the spectral distribution of the photon fluence due to gamma emitting radionuclides in the soil (Beck et al. 1972; ICRU 53 1994; Miller and Shebell 1995). Due to the relative ease of data collection, the technique is particularly well suited for quickly determining levels of contamination over large areas. Data analysis requires some care, however, as each *in situ* measurement provides a weighted average over the detector's field of view, which typically measures on the order of a hundred square meters at a detector height of 1 m. For example, in the case of uniformly depth-distributed ^{137}Cs about 30% of the measured fluence rate originates beyond a 3 m radius with a detector positioned 1 m above the soil surface. In the case of a surface source distribution, about half the fluence comes from the area beyond 10 m from the detector (ICRU 53 1994). A series of measurements performed on the order of meters apart will, therefore, provide overlapping coverage of the ground area, which will ensure that any areas of elevated activity (hot spots) between grid points are not missed. A grid spacing of 5 m and a detector height of 1 m is adequate for most radionuclides and depth profiles of interest.

A set of measurements performed on a grid will provide valuable information on the spatial distribution of radionuclides in soil, but it can never provide enough information to determine a unique distribution of the contaminant. In general, a large number of distributions will fit the same data equally well. This can be understood intuitively, since a finite number of measurements can not be sufficient to determine a continuous distribution. (This observation is not restricted to the data collected with the *in situ* technique but holds for any set of discrete measurements, such as a series of soil samples.) Analysis of this type of data is therefore best viewed as a problem of inference, in which we make use of the set of measurements together with some additional assumptions to choose our best estimate of the distribution. There are several approaches to dealing with the issue of inferring a distribution from a set of *in situ* measurements performed over a grid, and we discuss two of them here.

One approach that is often used is to plot the measurements on a map and then draw a contour plot. The assumption that is made here is that the activity levels of deposition on the ground follow the pattern of the contour plot of the count rates. This approach is very easy to implement. One can calculate conversion factors from count rate to deposition the ground (by using the assumption of spatial uniformity or constant levels of activity), and then "plug in" those conversion factors to obtain activity levels of deposition on the ground. However, there are two problems that can arise when using this method. The first, and most troubling problem, is that the distribution of activity on the ground constructed in this way usually will *not* fit the measured data. This is because the assumption of spatial uniformity does not hold when the contamination is non-uniform. The second problem is that contour plots will not allow for any result that exceeds the highest value measured at a grid point, and therefore cannot be used to single out potential hot spots that might be present between grid points. These hot spots, as will be shown later, can occur at places where one might not at first expect them to occur when just looking at the contour plot.

Another approach to the analysis of *in situ* gamma-ray spectrometry data is to do a deconvolution (or unfolding) of the data. This is the method used in ISDMAP (*in situ* deconvolution and mapping program). Using ISDMAP's method of data deconvolution results in distributions of activity on the ground that always fit the data to within experimental error.

RUNNING ISDMAP

Prior to running the program, the user will need to create the response function file, the input file and (if required) the default distribution file, which must be in the same directory as the executable file (file specifications are detailed in the next section of this report). Enter into a DOS prompt and go to the directory that contains the ISDMAP program.

While most of the information needed to run the code is read from input data files, some information is provided by the user at run time. Generally, after the user has entered their response to the question, he or she should type an "Enter". After starting the program the user will be prompted as follows:

The user chooses the mode to run the program in:
"ENTER YOUR CHOICE OF DECONVOLUTION,
"1 FOR CHARACTERIZATION (TO ESTIMATE THE DISTRIBUTION)
"2 FOR POST-REMEDIATION (TO SEARCH FOR HOT SPOTS):"
For help with this, see discussion in the section titled ISDMAP Operating Modes.

The user is prompted for the name of the input file:
"NAME OF FILE WITH INPUT DATA?"

The user should type a response followed by a return. The file name can be any legal DOS file name (up to 32 characters for the file name, followed by a period, followed by up to 3 characters for the extension).

The user is prompted for the name of the output files:
“NAME OF OUTPUT FILE? (USE 8 CHARACTERS)”
The user should type in no more than eight characters and no extension. **Warning:** ISDMAP will not overwrite an existing file. If ISDMAP output files with this name already exist in this directory, the program will quit after execution, without saving the output files.

The user is given a choice for modeling the depth profile of the activity on the ground:
“ENTER YOUR CHOICE OF SOURCE DEPTH PROFILE,
“1 FOR UNIFORM DISTRIBUTION
“2 FOR PLANE DISTRIBUTION”
These are the two extreme cases. In the uniform distribution, the contaminant is assumed to be distributed uniformly in depth, while in the surface distribution it is assumed to be only on the surface. The assumption of uniform distribution is more conservative, and its use is normally recommended. Note: if you included data from soil samples, you must choose a uniform source distribution.

If the user chooses a uniform distribution, the following options are available:
“ENTER YOUR CHOICE OF UNITS FOR THE ACTIVITY PER MASS,
“1 FOR Bq/kg
“2 FOR pCi/g :”

If the user chooses a plane distribution, the following options are available:
“ENTER YOUR CHOICE OF UNITS FOR THE ACTIVITY PER AREA,
“1 FOR Bq/m**2
“2 FOR pCi/m**2”

The user chooses a method of determining the background distribution:
“ENTER YOUR CHOICE OF METHOD TO DETERMINE THE BACKGROUND,”
the options depend on whether the user is in post-remediation mode or characterization mode.
For post-remediation mode the options are:
“1 TO DETERMINE THE BACKGROUND FROM THE MEASURED DATA
“2 TO ENTER A BACKGROUND COUNT RATE”
and for characterization mode they are:

"1 TO USE THE DATA TO ESTIMATE A SMOOTH BACKGROUND
"2 TO READ IN BACKGROUND DATA FROM A FILE :"

If the user chooses to enter a background count rate, they are prompted to enter it in:
"ENTER COUNT RATE (>0) IN counts per minute (cpm)"
Note that this must appear in cpm.

If the user chooses to read in the background data from a file, the name of the file must be entered:
"NAME OF FILE WITH BACKGROUND DATA?"

If the user has chosen to do a "POST-REMEDIATION" deconvolution, the area of the hot spot (in meters) must be entered:
"AREA OF HOT SPOT? (IN SQUARE METERS, e.g.: 1.0)"
The hot spot is modeled as a square. Note that this is an area, so when searching for a hot spot that is 2 m by 2 m the user should enter 4.0. If the area specified by the user does not match the size of a square that can be constructed out of the cells used to model the ground, the program will approximate it by a smaller square, and will print to screen the size chosen.

ISDMAP OPERATING MODES

Each run of ISDMAP operates the program in one of two modes: the characterization mode or the post-remediation (hot spot detection) mode.

When ISDMAP operates in the characterization mode, the program produces a map that provides a smooth, "best estimate" of the distribution of the contaminant. The characterization mode is useful, for example, during characterization surveys.

When ISDMAP operates in the post-remediation mode, the program produces a map that, given a background level, shows potential hot spots that might be "hidden" in the data. The post-remediation mode is useful, for example, for final status surveys after remediation.

CHARACTERIZATION MODE

In this mode, ISDMAP will create a map of the contamination in surface soil that fits the experimental data within error. The user is given the option of having ISDMAP produce a solution without using *a priori* information, or of allowing the user to introduce *a priori* information into the analysis process. In maximum entropy data deconvolution, such *a priori* information can be included in the deconvolution through what is known as a default distribution. In the first case, ISDMAP uses the data to make an initial estimate of the average concentration of the contaminant at the measurement points, and then creates a smooth distribution using radial basis functions interpolation. If this initial distribution does not fit the data (i.e., if the chi-square per degree of freedom is different from 1), it uses an iterative process to improve upon the previous estimate to create a new smooth distribution using radial basis functions that fits the data better. [For a review of radial basis functions as well as several other interpolation methods, see Franke (1982)]. The distribution that results from this process is used as the initial estimate for the maximum entropy deconvolution. In the second case, the user prepares a file with a default distribution that takes into account *a priori* information that is specific to the site.

POST-REMEDIATION MODE

In this mode, ISDMAP will create a Hot Spot Map (with extension 8HSM) showing potential hot spots that might be “hidden” in the data.

The default distribution is always set to a constant background level. The user is given the option of having ISDMAP produce a solution without using *a priori* information, in which case the default is set to the concentration that minimizes the chi-squared, or of allowing the user to introduce *a priori* information into the analysis process, in which case the user enters a background level.

The post-remediation option makes use of an algorithm that is almost identical to the one developed for the computer program ISD97, described in detail in Reginatto et al. (1997) [see also Reginatto et al, (1998)]. A complete description of the algorithm is beyond the scope of this report, and we provide here only a brief summary of the approach used by ISDMAP. To generate a Hot Spot Map, the program uses a two-step process. In the first step, a maximum entropy deconvolution of the data is carried out assuming a constant default distribution. In the second step, this initial maximum entropy solution is analyzed, and the information is used to find the locations and magnitudes of potential hot spots. New deconvolutions are then done for each potential hot spot. This second series of deconvolutions is carried out using default distributions that are constant everywhere, except at the location of the potential hot spot, where it is assumed that there is an area of elevated activity. The results of this second series of deconvolutions are all plotted on the Hot Spot Map.

FILE SPECIFICATIONS OF ISDMAP

In order to perform a deconvolution of a data set, the user needs to provide ISDMAP information about the detector's response and the readings that were taken with the detector. To do this, the user needs to prepare at least two comma delimited data files—these files can be created with a spread sheet and exported as comma delimited files (*.CSV files), or with a simple text editor, separating datum with commas (do not attempt to use a word processor, as it will insert extraneous characters, causing the program to crash).

The numbers in these files are in either integer or real numbers format, which will be noted in parentheses for each entry. Real numbers can be written in standard notation (e.g., 638.5) or in exponential notation (e.g., 6.385E+02).

THE DETECTOR RESPONSE FUNCTION (RF) INPUT FILE

Note: We use the same terminology used in Miller and Shebell (1995). The parameters are organized onto lines as follows:

1. Number of Response Functions

Form: NR

NR = the number of response functions in the file (an integer).

2. Individual Response Functions

Form: ID, E, GPD, CR, A_1, A_2, A_3,

ID = The detector number is a number used to identify the detector (an integer).

E = The photon energy is a number used to identify the gamma line being measured (an integer).

GPD = The gamma-per-disintegration—the fraction of photons, emitted per disintegration of the radionuclide (a real number).

CR = The full absorption peak count rate per unit incident fluence rate (a real number).

A_i = Coefficients to a third order polynomial fit of the relative angular response as a function of the angle incidence, in degrees (three real numbers).

Note: there should be as many lines of response function parameters as the number of measurements specified in the first line of the ID file (NR).

Use: E should be set equal to the photon energy, in keV, rounded to the nearest integer. CR is a measure of the efficiency of the detector for flux perpendicular to the face of the detector, in units of cpm/(gamma cm^{-2} sec^{-1}). The subscript of the coefficients, A, corresponds to the *power of the variable*. For example A_1 is the coefficient of the linear term, A_2 is the coefficient of the quadratic term. For a description of the relative angular response see EML Procedures Manual (1997).

Figure 1 shows an example of a RF file, with the response function data for the examples discussed in Examples A and B.

THE DATA INPUT FILE

Figures 2 and 3 show the input files used for the examples in Examples A and B respectively. These figures can be used as examples of typical data input files. The first five lines contain general information about the measurements, and the other lines contain information specific to each measurement. The file is organized as follows:

1. Number of Measurements

Form: N

N = total number of measurements in the file (an integer).

2. The Grid

Form: NX, NY, X_0, Y_0

NX = number of cells in the X direction (an integer).

NY = number of cells in the Y direction (an integer).

X_0 = X coordinate of the origin for the area to modeled (a real number).

Y_0 = Y coordinate of the origin for the area to modeled (a real number).

Use: When choosing the size of an area to be modeled, one should choose an area that is larger than the area covered by the measurements. This is because more than half of the field of view of a detector located at the perimeter of the measurement area can lay outside of the measurement area and the contributions to the count rate from background activity that is outside of the measurement area can not be neglected. An example is shown in Figure 3. The parameters X_0 and Y_0 can be used to shift the cells to correspond with a user-defined or standard coordinate system, such as State Plane Coordinates. Figure 4 also illustrates use of this option—notice how the lower right hand corner of the modeled area is shifted.

3. Cell Size and Grid Spacing

Form: CS, D

CS = The size, in meters, of the cells used to model the ground (a real number).

D = The distance between two detectors on the grid, in meters (a real number).

Use: The optimal cell size is determined from the following considerations: the cells must be small enough to ensure that the approximation of modeling the ground by discrete cells does not introduce a substantial error in the calculation, but large enough to reduce the total number of cells needed to model the ground—to reduce the computational burden. Cells with sides measuring 0.5 m have proven adequate to balance these two conflicting needs for a detector that is 1 m above the ground and on a 5 m grid. For other choices of detector height, the cell size and grid spacing should be scaled according to the detector height.

4. Detector ID, File Name

Form: ID, AF, RF

ID = the detector I.D. number (an integer).

AF = the name, with path (if necessary), of the attenuation file (text).

RF = the name, with path (if necessary), of the response function file (text).

Use: The ID must correspond to the detector I.D. in the RF file.

5. Gamma-Ray Energy

Form: E

E = the gamma-ray energy of the nuclide of interest, in keV (an integer).

Use: This number must be the same as the number that appears in the Response File.

6. Measurement Parameters (*In Situ* Data)

Form: n_i, H, x_i, y_I, CR_i, U_i

n_i = The I.D. number of the i^{th} measurement (an integer) ($1 < i < N$).

H = The detector height, in meters (a real number).

X_i = X coordinates of the i^{th} measurement, in meters (a real number).

Y_i = Y coordinates of the i^{th} measurement, in meters (a real number).

CR_i = The measured count rate, in counts per minute (a real number).

U_i = The error associated with the i^{th} measurement, in counts per minute (a real number).

Note: 6 and 7 combined should contain as many lines as the number of measurements specified in the first line of the ID file (N).

Use: Format corresponds to *in situ* data. These lines contain information specific to each measurement. Data from *in situ* data as well as soil samples must be contained in the same file if they are both going to be used in the deconvolution.

7. Measurement Parameters (Soil Data)

Form: n_i, 0, x_i, y_I, A_i, U_i

n_i = The I.D. number of the i^{th} measurement (an integer) ($1 < i < N$).

0 = The number 0, signaling no detector height, i.e., a soil sample (an integer).

x_i = X coordinates of the i^{th} measurement, in meters (a real number).

y_i = Y coordinates of the i^{th} measurement, in meters (a real number).

A_i = the measured activity (a real number).

U_i = The error associated with the i^{th} activity measurement (a real number).

Note: 6 and 7 combined should contain as many lines as the number of measurements specified in the first line of the ID file (N).

Use: Format corresponds to soil data. For activity measurements, the same units (e.g., pCi/g, Bq/g) must be chosen at run time for reporting the results of the deconvolution ***Warning:*** *once units are chosen for soil samples, the same units must be chosen at run time to avoid erroneous results.*

THE DEFAULT DISTRIBUTION (DD) INPUT FILE

Figure 5 shows the first 10 lines of a DD file.

1. Minimum Acceptable Value

Form: minV

minV = The lowest acceptable value, in the chosen units of concentration (a positive, non-zero real number).

Use: Any concentration in the rest of the file that is below this value is reset to this value by the program. This can be useful for making sure that small or negative values generated by an interpolation program do not make their way into the deconvolution process.

2. Cell Coordinates and Activity

Form: X, Y, *A*

X = X coordinates of the cell, in meters (a real number).

Y = Y coordinates of the cell, in meters (a real number).

A = the activity assigned to the cell at the coordinates specified above (a real number).

Note: there should be as many activities assigned to cells listed in the file DD as there are cells used to model the ground (for example, if the ground is modeled by a rectangle made out of 40x60 cells, the DD file must have 1 (lowest acceptable value) + 40 x 60(x,y,z triplets)=2401 lines).

Use: The coordinates (X, Y) must line up exactly with those used to model the ground. Note that the first value must be (X-offset, Y-offset), which are the x and y coordinates (in meters) assigned to the lower left corner of the first cell. For example, if there is no offset, the first value must be (0.0,0.0). *A* must be in the same units that are chosen for the deconvolution. It is suggested that a DD file be produced by estimating the concentrations at several points, using all data available, and then filling in the rest of the grid with an interpolation method. For a review of interpolation methods see Franke (1982).

THE ATTENUATION INPUT FILE

A file with mass attenuation coefficients for air and soil is included with the program, and need not be edited before running the program. There are, however, some variables that, while reasonably accurate for most cases, may need to be adjusted to achieve higher levels of accuracy (refer to IRCU report 53 for more details). For soil surfaces that are not excessively wet and for sites below 1.6 km (1 mile) above sea level, the attenuation data should be within 10% of the actual value without correction.

The attenuation file has two sections. The first has data related to the mass attenuation coefficients for air, and the second section has data related to the mass attenuation coefficients for soil. The data for air is taken from Table 4 (air, dry, near sea level) of Hubell and Seltzer (1995), and the data for soil is taken from Table 2 (soil, 10% H_2O) of Beck et al. (1972). The lines of this file are as follows:

1. Number of Air Attenuation Values and Density of Air

Form: NAA, ρ

NAA = The number of air attenuation values listed (an integer).

ρ = The density of the air measured in g/cm^3 (a real number).

2. Air Attenuation Values

Form: E, μ_a

E = The photon energy in keV (a real number).

μ_a = The mass attenuation coefficient, at the photon energy E, of the air measured in cm^2/g (a real number).

Note: there should be as many lines as the number specified in the first line of the ID file (NAA).

3. Number of Soil Attenuation Values and Density of Soil

Form: NSA, ρ

NSA = The number of soil attenuation values listed (an integer).

ρ = The density of the soil measured in g/cm^3 (a real number).

4. Soil Attenuation Values

Form: E, μ_s

E = The photon energy in keV (a real number).

μ_s = The mass attenuation coefficient, at the given photon energy E, of the soil measured in $cm^2\ g^{-1}$ (a real number).

Note: there should be as many lines as the number specified in item 3 (NSA).

Use: The photon energies not specifically listed in this file will be determined by log-log interpolation from the two nearest data values. To set the value for a given run, simply add the value you want used to the file at the appropriate energy, updating the NAA or NSA to reflect your changes.

VIEWING AND INTERPRETING THE RESULTS

The output consists of some or all of the following files:

(1) OUTPUT.TBL
(2) OUTPUT.SCS
(3) OUTPUT.DEF
(4) OUTPUT.HSM

Here, "OUTPUT" is the output file name set by the user. The first two files are always created. The third file is created only if the program is run in "CHARACTERIZATION" mode. The fourth file is created only if the program is run in "POST-REMEDIATION" mode.

The file OUTPUT.TBL provides a summary of the results of the deconvolution. The average count rate and average activity are calculated by averaging over the measurements. Since the background average count rate and background average activity will depend on the choice of method used to determine the background, they will not be equal to the average count rate and average activity in most cases. A successful deconvolution will result in a distribution that fits the data, that is, one for which the chi-square per degree of freedom is equal to 1. The user should always check the chi square of the distributions, because in some instances the algorithm used to maximize the entropy might not converge. In this case, the final distribution calculated by the code will not fit the data. OUTPUT.TBL files for examples A and B are shown in Figures 6 and 8. Note that the column labeled "deviation" provides this value for each measurement, expressed in terms of the number of standard deviations from the mean.

The OUTPUT.SCS and OUTPUT.HSM contain the result of the deconvolution. All lines contain three numbers, the {x,y} coordinates of the cells used to model the ground, in meters, and the activity assigned to that cell by the deconvolution, separated by spaces.

To view these files, the user needs to import them into a graphing package. The files are comma delimited and need to be parsed as such by importing them as comma delimited files. Each individual program has its own parsing package that should be detailed in the documentation.

EXAMPLE A

Details of this field measurement can be found in Shebell et al. (1999). Here, we will take the data as given, and consider the analysis of the data only. A total of 18 measurements were taken on a 5 m triangular grid and eight soil samples were taken at arbitrary points in that grid. The data was run using the characterization option, with a default distribution estimated by the program. The input file (the file INPUT_A.CSV) is shown in Figure 2 and the table summarizing the output (the file OUT_A.TBL) is shown in Figure 6. The resulting contamination levels in soil are represented graphically in Figure 7.

EXAMPLE B

Details of this field measurement can be found in Reginatto et al. (1998). Here, we will take the data as given, and consider the analysis of the data only. A total of 18 measurements were made on a 5 m triangular grid, and the data was run using the post-remediation option. The input file (the file INPUT_B.CSV) is shown in Figure 3, and the table summarizing the output (the file OUT_B.TBL) is shown in Figure 8.

The distribution of activity in the soil that results from the maximum entropy deconvolution is shown in Figure 9. This distribution tends to be constant and equal to the background activity, except in the neighborhood of each measurement point where there is a localized peak or valley depending on whether the measured peak count rate at that point is higher or lower than the background peak count rate. Figure 10 shows on one map all the potential hot spots identified by the code. They are modeled as squares, of 4 m^2 of area (2 m on each side). Nine potential hot spots were identified. Note that the largest hot spot does not coincide with the highest peak of the maximum entropy deconvolution. Instead, it falls in between smaller contiguous peaks of similar size, since in this case the data allows for a large peak between measurement points. The position and magnitude of this peak would not have been singled out by a contour plot.

ACKNOWLEDGMENTS

The authors wish to acknowledge the contributions made by Kevin Miller to the development of the methodology presented here, and would like to thank Matt Monetti for data used in this paper. Funding for this work was supported initially by the Office of Energy Research within the Department of Energy, and most recently by the Office of Environmental Management, including the Fernald Soils Accelerated Site Technology Deployment Project under the EM Office of Science and Technology. Additional funding was also provided by the Division of Regulatory Research within the Nuclear Regulatory Commission under contract W-6147.

REFERENCES

Beck, H. L., J. De Campo, and C. V. Gogolak
In Situ Ge(Li) and NaI(Tl) Gamma-Ray Spectrometry.
U. S. Atomic Energy Commission Report HASL-258, New York, NY (1972)

EML Procedures Manual
U. S. Department of Energy Report HASL-300, 28th Edition, Vol. I, Section 3, February (1997)
[available at www.eml.doe.gov/publications/procman/]

Franke, R.
Scattered Data Interpolation: Tests of Some Methods.
Mathematics of Computation 38:181-200 (1982)

Hubell, J. H. and S. M. Seltzer
Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients 1 keV to 20 MeV for Elements Z=1 to 92 and 48 Additional Substances of Dosimetric Interest
National Institute of Standards and Technology, NISTIR 5632, Gaithersburg, MD (1995)

International Commission on Radiation Units and Measurements (ICRU)
Gamma-ray Spectrometry in the Environment
ICRU Publication 53, Bethesda, Maryland (1994)

Kansa, E. J. and R. E. Carlson,
Radial Basis Functions: A Class of Grid-Free, Scattered Data Approximations
Lawrence Livermore National Laboratory, UCRL-JC-110061 (1992)

Miller, K. M. and P. Shebell
In Situ Gamma-Ray Spectrometry - a Tutorial for Environmental Radiation Scientists
USDOE Report EML-557 (1995)

Miller, K. M., M. Reginatto, P. Shebell, G. A. Klemic, and C. V. Gogolak
Spectrometric Techniques for the Measurement of Radioactive Contamination in Buildings
Proceedings of Air and Waste Management Association's 90th Annual Meeting, Pittsburgh, PA (1997)

Reginatto, M. and P. Goldhagen
MAXED, A Computer Code for the Deconvolution of Multisphere Neutron Spectrometer Data Using the Maximum Entropy Method
USDOE Report EML-595 (1998)

Reginatto, M., P. Shebell, and K. M. Miller
ISD97, a Computer Program to Analyze Data from a Series of *In Situ* Measurements on a Grid and Identify Potential Localized Areas of Elevated Activity
USDOE Report EML-590 (1997)

Reginatto, M., P. Shebell, and K. M. Miller
An Alternative Approach to Hot Spot Identification Using *In Situ* Gamma Spectrometry Measurements on a Grid
Health Physics 74:481-485 (1998)

Shebell, P., M. Reginatto, M. Monetti, S. Faller and L. Davis
Mapping of Depleted Uranium with *In Situ* Spectrometry and Soil Samples
Proceedings of the International Symposium on Restoration of Environments with Radioactive Residues, November 29 – December 3, 1999, IAEA-SM-359, Arlington, VA (1999)

Figure 1. Response Function File RF.CSV for Examples A and B. Note the additional commas, these are included by some spread sheet programs and do not interfere with the operation of ISDMAP. Note also that in this example we have set the coefficients of the third order polynomial fit of the relative angular response equal to zero, which would be adequate if such information was not available and the response of the detector was not strongly dependent on the angle of incidence.

```
8,,,,,,,,
2670,63,0.0392,307.67,-1.55E-02,3.89E-04,-1.80E-06,,
2670,583,0.858,515.16,0,0,0,,
2670,609,0.461,500.42,0,0,0,,
2670,662,0.852,473.92,0,0,0,,
2670,911,0.29,383.58,0,0,0,,
2670,967,0.2295,369.01,0,0,0,,
2670,1001,0.00845,360.44,0,0,0,,
2670,2615,0.998,191.08,0,0,0,,
```

Figure 2. Input File INPUT_A.CSV for Example A. Note that there are no measurements numbered 4, 21, 24, 26, 27, 30, 31. Although the measurements are numbered 1-33, there are only 26 measurements, which agrees with the number on the first line. Also note that soil samples are included (measurements numbered 20-33, though they need not be at the end), and that they are listed with a height of 0 m.

```
26
120,120,0,0
0.5,5
2670,C:\jdk\decon\macoeff.csv,C:\jdk\decon\rf.csv
63
1,1,19.25,20.73,509.2,21.2
2,1,19.25,25.73,640.6,22.8
3,1,19.25,30.73,746.2,25
5,1,19.25,35.73,699,27.4
6,1,23.58,18.23,206.6,14.6
7,1,23.58,23.23,419.2,20
8,1,23.58,28.23,369,20
9,1,23.58,33.23,191.2,12.6
10,1,23.58,38.23,62,14.6
11,1,23.58,43.23,55.6,13.8
12,1,27.91,20.73,102,13.4
13,1,27.91,25.73,76,11.4
14,1,27.91,30.73,63.5,7.6
15,1,27.91,35.73,44.8,9.1
16,1,27.91,40.73,39.4,7.5
17,1,32.24,33.23,35.1,7.1
18,1,32.24,38.23,31,7.1
19,1,36.57,35.73,12.866667,4.733333
20,0,21.4,23.2,809.72,404.858044
22,0,21.4,28.2,336.02,168.007519
23,0,21.4,40.7,7.69,3.846638
25,0,25.8,23.2,13.98,6.98818
28,0,25.8,28.2,5.56,2.778442
29,0,25.8,35.7,7.81,3.905673
32,0,30.1,30.7,2.33,1.16377
33,0,30.1,35.7,1.94,0.969302
```

Figure 3. Input File INPUT_B.CSV for Example B.

```
18
160,160,30.0000,50.0000
0.5,5
2670, C:\windows\isdmap\macoeff.csv, C:\windows\isdmap\rf.csv
1001
1,1.00,65.6667,80.0000,6.3330,1.4000
2,1.00,65.6667,85.0000,8.9330,1.2000
3,1.00,65.6667,90.0000,6.4000,1.0000
4,1.00,65.6667,95.0000,5.4670,1.1330
5,1.00,65.6667,100.0000,4.6670,1.0670
6,1.00,65.6667,105.0000,5.8000,1.0000
7,1.00,70.0000,77.5000,7.9330,1.5330
8,1.00,70.0000,82.5000,9.4000,1.3330
9,1.00,70.0000,87.5000,5.2000,1.2000
10,1.00,70.0000,92.5000,7.2670,1.333
11,1.00,70.0000,97.5000,5.4670,1.2670
12,1.00,70.0000,102.5000,4.8000,1.6000
13,1.00,74.3333,80.0000,7.6000,1.1330
14,1.00,74.3333,85.0000,9.5330,1.4000
15,1.00,74.3333,90.0000,10.2670,1.2670
16,1.00,74.3333,95.0000,9.4000,1.3330
17,1.00,74.3333,100.0000,10.3890,0.6640
18,1.00,74.3333,105.0000,8.0670,1.2000
```

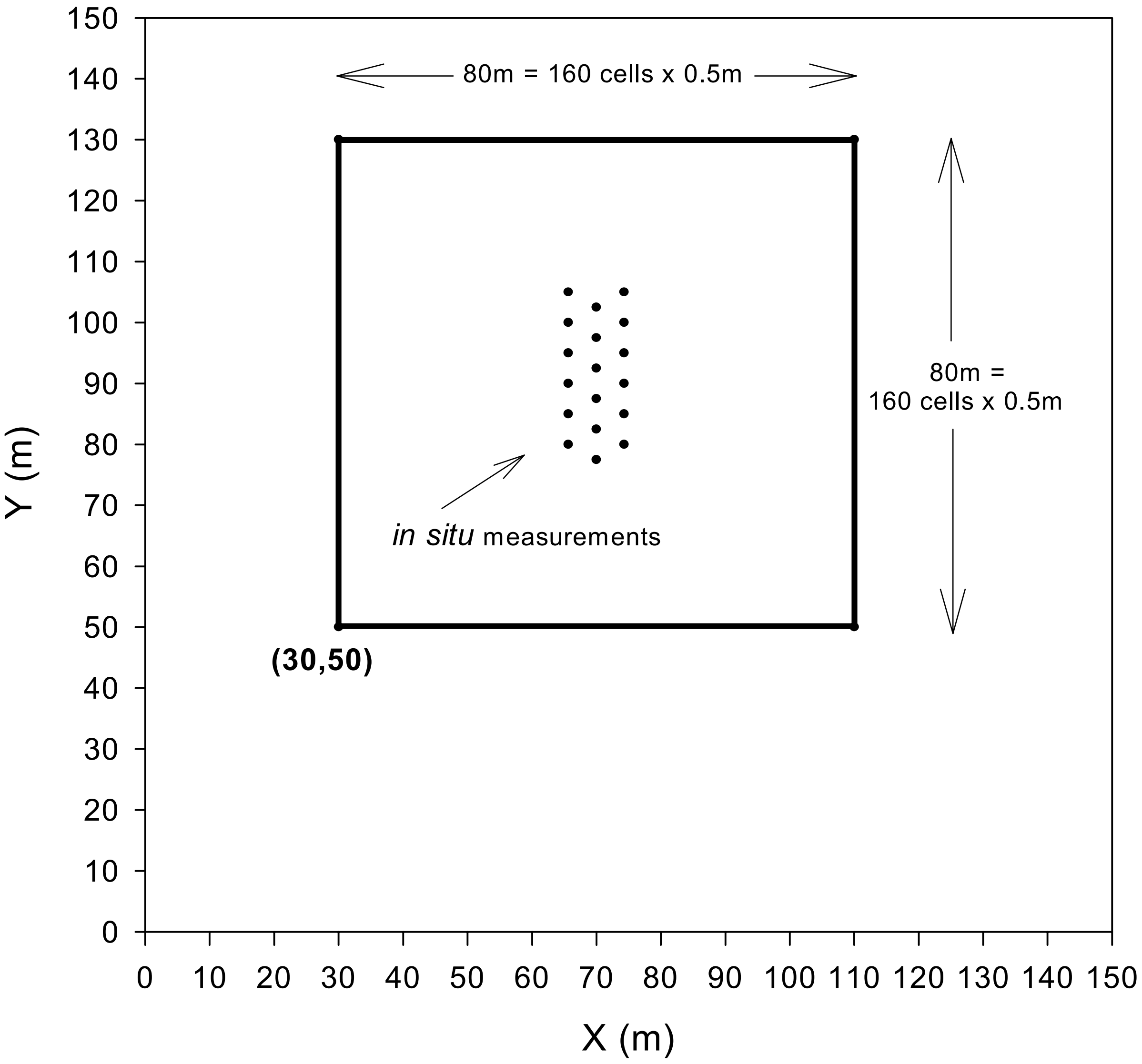

Figure 4. Grid map for Example B.

Figure 5. The first 10 lines of a Default Distribution File. Notice the negative number at location (1.0,0.0). This number, along with all others that are less than 37.05 (the value on the first line of the file) will be set to 37.05 by ISDMAP.

```
37.05
0.0,0.0,28.04523425
0.5,0.0,37.03777778
1.0,0.0,-11.9377037
1.5,0.0,50.63444444
2.0,0.0,80.56666667
2.5,0.0,103.7388889
3.0,0.0,122.2040741
3.5,0.0,137.7848148
4.0,0.0,151.9644444
4.5,0.0,165.8888889
...,...,...
```

Figure 6. Output File OUT_A.TBL for Example A.

```
NAME OF FILE WITH INPUT DATA : input_a.csv
NAME OF OUTPUT FILE :    out_a
NAME OF FILE WITH DETECTOR RESPONSE DATA : C:\windows\isdmap\rf.csv
NAME OF FILE WITH ATTENUATION COEFFICIENTS DATA : C:\windows\isdmap\macoeff.csv

SMOOTH BACKGROUND WAS ESTIMATED FROM THE DATA

DETECTOR NUMBER :       2670
PHOTON ENERGY (KeV) =    63

AVERAGE COUNT RATE (cpm) = 2.3907E+02
AVERAGE ACTIVITY (pCi/g     ) = 2.7075E+02
BACKGROUND AVERAGE COUNT RATE (cpm) : 2.1233E+02
BACKGROUND AVERAGE ACTIVITY (pCi/g     ) : 1.4703E+02

CHI SQUARE PER DEGREE OF FREEDOM / BACKGROUND = 9.7322E-01
CHI SQUARE PER DEGREE OF FREEDOM / MAXIMUM ENTROPY =     0.9732

RESULTS FOR MAXIMUM ENTROPY DECONVOLUTION :
MEASURMENT  MEASURED    CALCULATED  DEVIATION
NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.
         1  5.0920E+02  5.1438E+02     0.24415
         2  6.4060E+02  6.6967E+02     1.27494
         3  7.4620E+02  7.8264E+02     1.45777
         5  6.9900E+02  7.2867E+02     1.08268
         6  2.0660E+02  1.7586E+02    -2.10559
         7  4.1920E+02  4.2258E+02     0.16910
         8  3.6900E+02  3.4734E+02    -1.08304
         9  1.9120E+02  1.9296E+02     0.13950
        10  6.2000E+01  9.8264E+01     2.48380
        11  5.5600E+01  5.3458E+01    -0.15519
        12  1.0200E+02  8.8083E+01    -1.03862
        13  7.6000E+01  7.7888E+01     0.16560
        14  6.3500E+01  5.8754E+01    -0.62447
        15  4.4800E+01  3.7181E+01    -0.83722
        16  3.9400E+01  3.0327E+01    -1.20978
        17  3.5100E+01  2.5062E+01    -1.41376
        18  3.1000E+01  1.9240E+01    -1.65637
        19  1.2867E+01  1.2916E+01     0.01053
        20  8.0972E+02  8.0972E+02     0.00000
        22  3.3602E+02  3.3602E+02     0.00000
        23  7.6900E+00  7.6900E+00     0.00000
        25  1.3980E+01  1.3980E+01     0.00000
        28  5.5600E+00  5.5600E+00     0.00000
        29  7.8100E+00  7.8100E+00     0.00000
        32  2.3300E+00  2.3300E+00     0.00000
        33  1.9400E+00  2.1971E+00     0.26525
```

```
  RESULTS FOR DEFAULT DISTRIBUTION:
  MEASURMENT  MEASURED    CALCULATED  DEVIATION
  NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.
           1  5.0920E+02  5.1438E+02     0.24415
           2  6.4060E+02  6.6967E+02     1.27494
           3  7.4620E+02  7.8264E+02     1.45777
           5  6.9900E+02  7.2867E+02     1.08268
           6  2.0660E+02  1.7586E+02    -2.10559
           7  4.1920E+02  4.2258E+02     0.16910
           8  3.6900E+02  3.4734E+02    -1.08304
           9  1.9120E+02  1.9296E+02     0.13950
          10  6.2000E+01  9.8264E+01     2.48380
          11  5.5600E+01  5.3458E+01    -0.15519
          12  1.0200E+02  8.8083E+01    -1.03862
          13  7.6000E+01  7.7888E+01     0.16560
          14  6.3500E+01  5.8754E+01    -0.62447
          15  4.4800E+01  3.7181E+01    -0.83722
          16  3.9400E+01  3.0327E+01    -1.20978
          17  3.5100E+01  2.5062E+01    -1.41376
          18  3.1000E+01  1.9240E+01    -1.65637
          19  1.2867E+01  1.2916E+01     0.01053
          20  8.0972E+02  8.0972E+02     0.00000
          22  3.3602E+02  3.3602E+02     0.00000
          23  7.6900E+00  7.6900E+00     0.00000
          25  1.3980E+01  1.3980E+01     0.00000
          28  5.5600E+00  5.5600E+00     0.00000
          29  7.8100E+00  7.8100E+00     0.00000
          32  2.3300E+00  2.3300E+00     0.00000
          33  1.9400E+00  2.1971E+00     0.26525

TIME TO RUN THE PROGRAM (MINUTES) =       0.3000
```

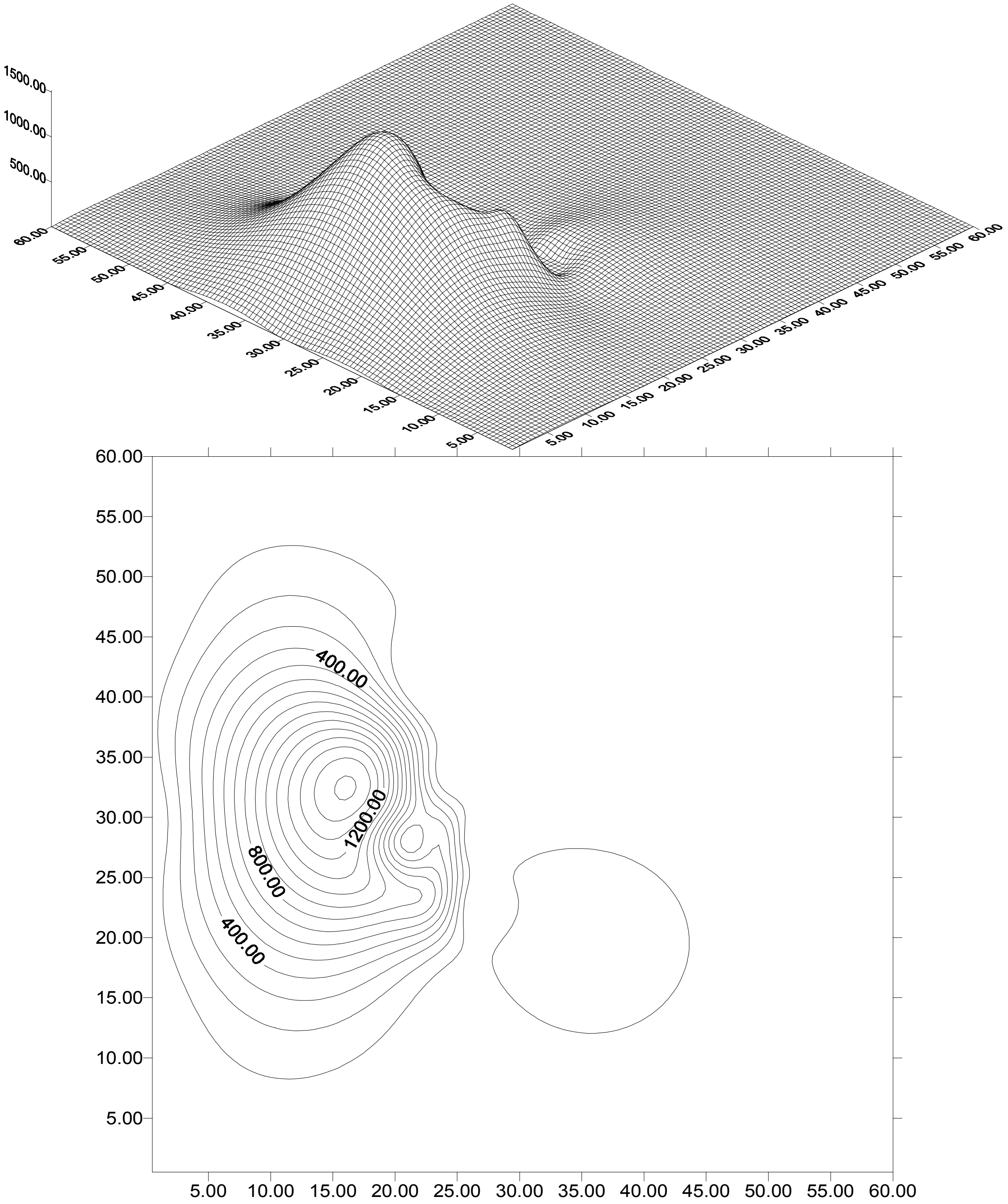

Figure 7. SCS deconvolution (file: OUT_A.SCS) from the run of Example A.

Figure 8. Output File OUT_B.TBL for Example B.

```
NAME OF FILE WITH INPUT DATA : input_b.csv
NAME OF OUTPUT FILE :    out_b
NAME OF FILE WITH DETECTOR RESPONSE DATA : rf.csv
NAME OF FILE WITH ATTENUATION COEFFICIENTS DATA : MACOEFF.CSV
BACKGROUND DISTRIBUTION WAS CALCULATED FROM THE DATA

DETECTOR NUMBER :       2670
PHOTON ENERGY (keV) =  1001

AVERAGE COUNT RATE (cpm) = 7.3846E+00
AVERAGE ACTIVITY ( Bq/kg    ) = 3.5653E+02
BACKGROUND AVERAGE COUNT RATE (cpm) : 7.6175E+00
BACKGROUND AVERAGE ACTIVITY ( Bq/kg    ) : 3.6777E+02

CHI SQUARE PER DEGREE OF FREEDOM / BACKGROUND = 3.0886E+00
CHI SQUARE PER DEGREE OF FREEDOM / MaxEnt DIST. =     1.0000

LARGEST POTENTIAL ELEVATED AREA NEAR MEASUREMENT :   2
AREA (m^2) =     4.0000
MAGNITUDE (x BACKGROUND) =     8.7138
MAGNITUDE ( Bq/kg    ) = 3.2047E+03
CHI SQUARE PER DEGREE OF FREEDOM OF THE
DISTRIBUTION WITH THE LARGEST HOT SPOT =    18.5195

RESULTS FOR MaxEnt DECONVOLUTION, SMALL CELL SIZE:
MEASURMENT  MEASURED    CALCULATED  DEVIATION
NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.
         1  6.3330E+00  7.3019E+00     0.69205
         2  8.9330E+00  8.0592E+00    -0.72815
         3  6.4000E+00  7.0745E+00     0.67454
         4  5.4670E+00  6.7864E+00     1.16455
         5  4.6670E+00  6.4580E+00     1.67855
         6  5.8000E+00  6.8097E+00     1.00965
         7  7.9330E+00  7.7030E+00    -0.15002
         8  9.4000E+00  8.1844E+00    -0.91191
         9  5.2000E+00  6.8992E+00     1.41601
        10  7.2670E+00  7.4678E+00     0.15067
        11  5.4670E+00  6.9430E+00     1.16499
        12  4.8000E+00  6.9674E+00     1.35463
        13  7.6000E+00  7.6456E+00     0.04021
        14  9.5330E+00  8.2133E+00    -0.94265
        15  1.0267E+01  8.5584E+00    -1.34854
        16  9.4000E+00  8.2341E+00    -0.87464
        17  1.0389E+01  9.5129E+00    -1.31950
        18  8.0670E+00  7.7899E+00    -0.23091
```

```
RESULTS FOR DEFAULT DISTRIBUTION:
MEASURMENT  MEASURED    CALCULATED  DEVIATION
NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.
         1  6.3330E+00  7.6163E+00     0.91666
         2  8.9330E+00  7.6212E+00    -1.09319
         3  6.4000E+00  7.6226E+00     1.22256
         4  5.4670E+00  7.6209E+00     1.90102
         5  4.6670E+00  7.6156E+00     2.76346
         6  5.8000E+00  7.6054E+00     1.80539
         7  7.9330E+00  7.6133E+00    -0.20854
         8  9.4000E+00  7.6203E+00    -1.33510
         9  5.2000E+00  7.6234E+00     2.01946
        10  7.2670E+00  7.6232E+00     0.26721
        11  5.4670E+00  7.6198E+00     1.69914
        12  4.8000E+00  7.6123E+00     1.75771
        13  7.6000E+00  7.6161E+00     0.01418
        14  9.5330E+00  7.6209E+00    -1.36578
        15  1.0267E+01  7.6223E+00    -2.08738
        16  9.4000E+00  7.6206E+00    -1.33489
        17  1.0389E+01  7.6154E+00    -4.17718
        18  8.0670E+00  7.6051E+00    -0.38489

POTENTIAL HOT SPOTS FOUND:
MEAS. N.  X COORD.  Y COORD.  MAG(xBKG) CHI SQR.
       2  6.85E+01  8.40E+01  8.71E+00  1.03E+00
       7  7.10E+01  8.20E+01  1.83E+00  9.86E-01
       8  7.10E+01  8.45E+01  2.11E+00  9.93E-01
      13  7.25E+01  8.40E+01  6.02E+00  9.79E-01
      14  7.35E+01  8.70E+01  3.23E+00  9.89E-01
      15  7.50E+01  9.05E+01  1.24E+00  1.00E+00
      16  7.50E+01  9.65E+01  1.46E+00  9.93E-01
      17  7.50E+01  9.95E+01  1.96E+00  9.72E-01
      18  7.50E+01  1.01E+02  1.41E+00  1.00E+00


TIME TO RUN THE PROGRAM (MINUTES) =    21.6500
```

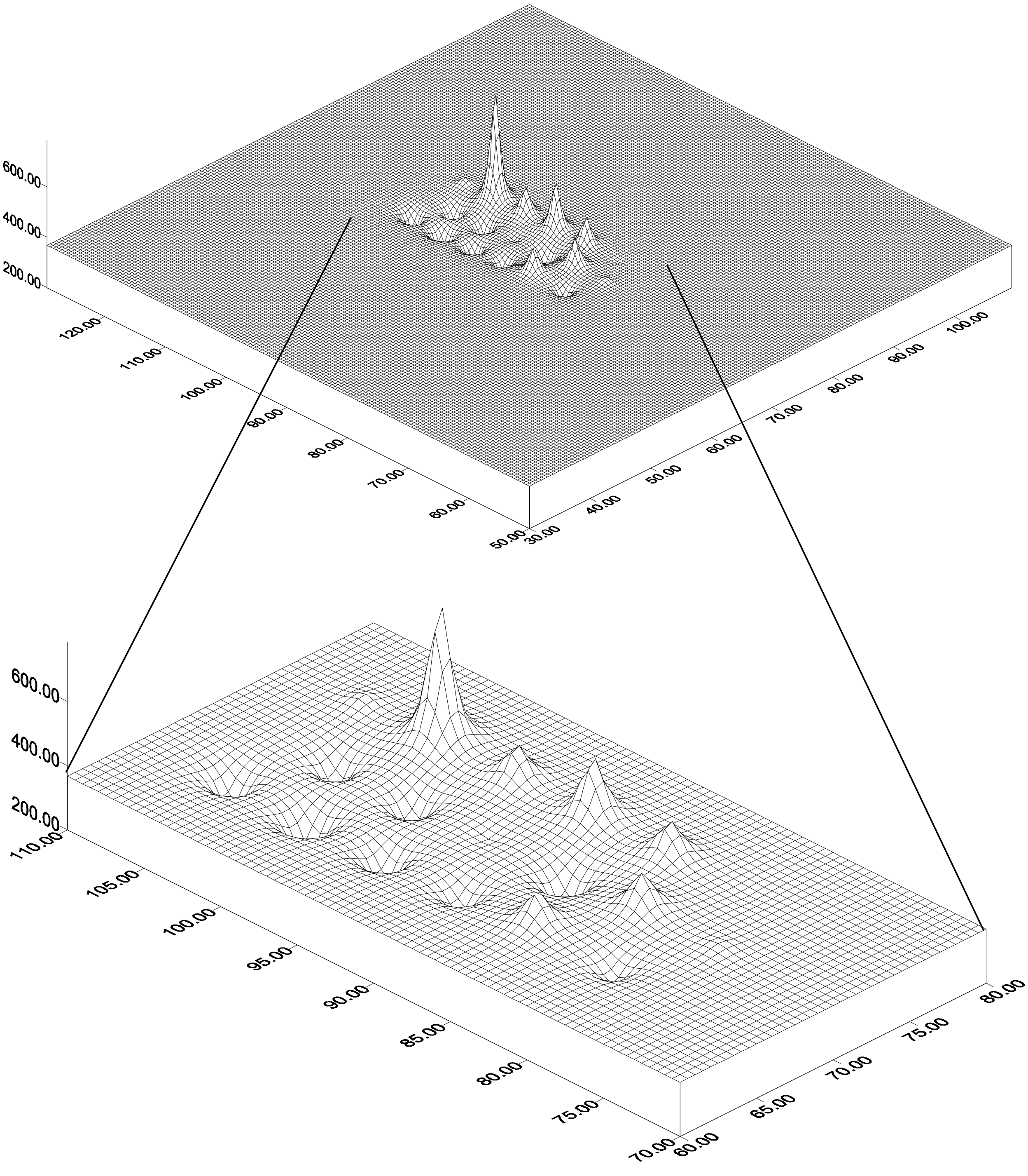

Figure 9. SCS deconvolution (file: OUT_B.SCS) for Example B.

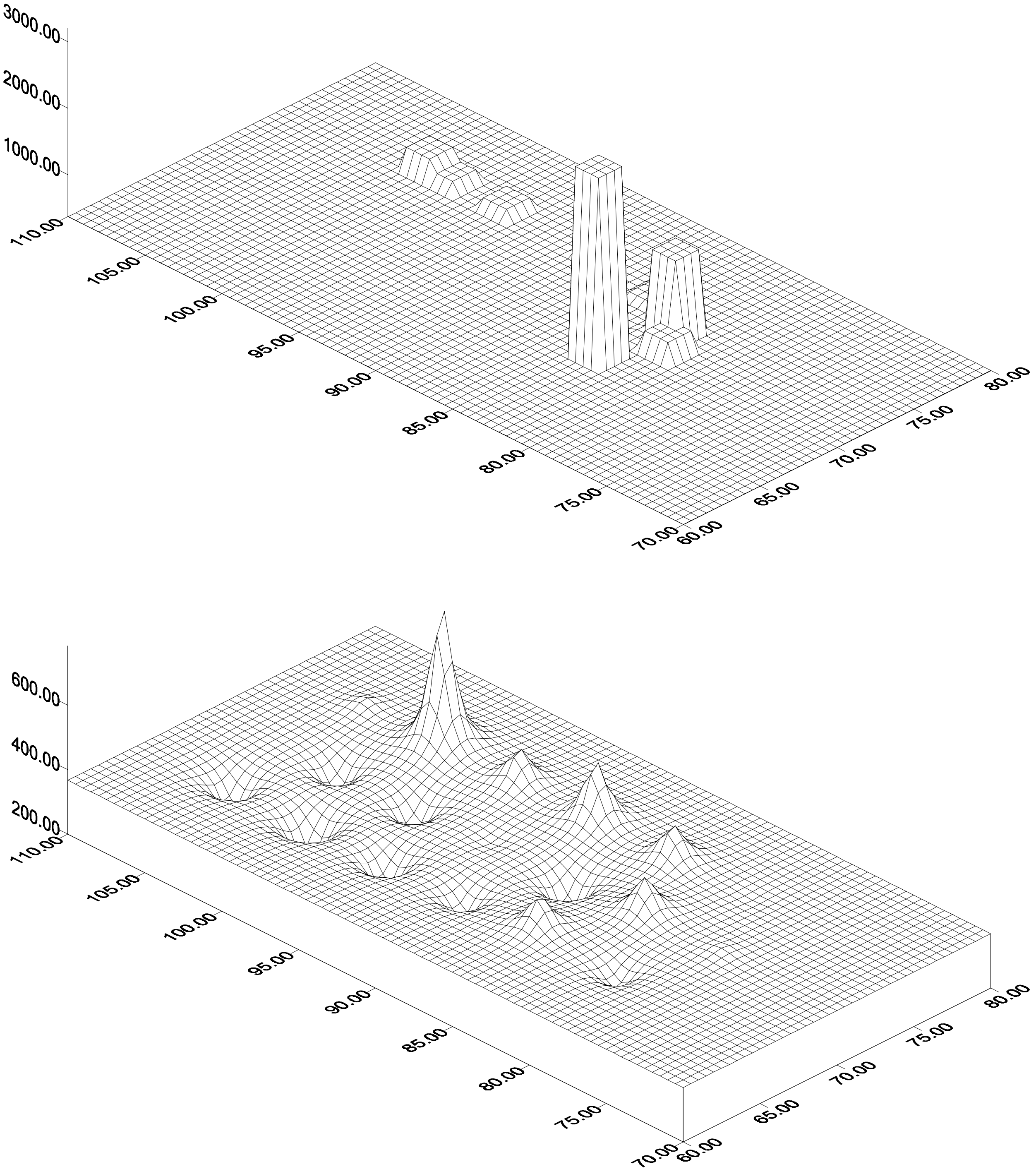

Figure 10. HSM deconvolution (file: OUT_B.HSM) for Example B. Shown with enlargement of the maximum entropy deconvolution for comparison. Notice the different scales on the vertical axis of the two figures.

Appendix

FORTRAN Source Code

```
C     Last change:  ES   29 Aug 2000   10:58 am
      PROGRAM ISDMAP
C
C     "LF95 ISDMAP -FIX -DBL"
C
      INTEGER M,NX,NY,NXY,MNXY,ISP,DETNUM,PE,IG,NSD,IQ,HSS,IX
      INTEGER IC1,ICR1,ICM1
      INTEGER MNUM(1000)
      REAL A,HSA,DBKG,GPD,AA,AS,NOP,ANG1,ANG2,ANG3,RHO,X0,Y0
      REAL D(1000),S(1000),HH(1000),X(1000),Y(1000)
      CHARACTER DH*60, RH*60, EH*8, UNITS*10, FBKGD*60,MH*60
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      CALL SYSTEM_CLOCK(IC1,ICR1,ICM1)
C
      CALL INPUT(DH,EH,UNITS,HSA,HSS,HH,X,Y,D,S,DETNUM,PE,MNUM,IG,NSD,
     1IQ,GPD,AA,AS,NOP,ANG1,ANG2,ANG3,DBKG,RH,RHO,X0,Y0,IX,FBKGD,MH)
C
      CALL ISD00(DH,EH,UNITS,HSA,HSS,HH,X,Y,D,S,DETNUM,PE,MNUM,IG,NSD,
     1IQ,GPD,AA,AS,NOP,ANG1,ANG2,ANG3,DBKG,RH,RHO,IC1,X0,Y0,IX,FBKGD,MH)
C
      END
C
C     **********************************************************
C
      SUBROUTINE ISD00(DH,EH,UNITS,HSA,HSS,HH,X,Y,D,S,DETNUM,PE,MNUM,IG
     1,NSD,IQ,GPD,AA,AS,NOP,ANG1,ANG2,ANG3,DBKG,RH,RHO,IC1,X0,Y0,IX,
     1FBKGD,MH)
C
      INTEGER M,NX,NY,NXY,MNXY,ISP,DETNUM,PE,IG,NSD,IQ,I,IHP,HSS,IX
      INTEGER IC1
      INTEGER MNUM(M)
      REAL A,HSA,DBKG,GPD,AA,AS,NOP,ANG1,ANG2,ANG3,RHO,X0,Y0
      REAL FBKG,DAVG,FAVG,HSMAX,CSQBKG,CSQMENXY
      REAL RCSQBKG,RCSQMENXY,RCSQHSI
      REAL CSQ,RKP,RLP,HHP
      REAL B(MNXY),D(M),S(M),HH(M),X(M),Y(M),EBKG(M),EMENXY(M)
      REAL FDEF(NXY),FMENXY(NXY),LAMBDANXY(M)
      REAL PFE(M),RKHS(M),RLHS(M),MHS(M),CSQHS(M)
      REAL HSFDEF(NXY),HSF(NXY),LAMBDAHS(M),HSE(M),HSFMAP(NXY)
      REAL XHS(M),YHS(M),XHSABS(M),YHSABS(M)
      CHARACTER DH*60, RH*60, EH*8, UNITS*10, FBKGD*60, MH*60
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     Calculate the detector response matrix B
C
      PRINT*,' CALCULATING THE DETECTOR RESPONSE MATRIX '
      PRINT*,' '
```

```
      CALL DETRESP(B,A,X,Y,NSD,IQ,AA,NOP,GPD,AS,ANG1,ANG2,ANG3,
     1HH,RHO,M,NXY,MNXY,NX,NY)
C
C     Calculate the background activity
C
      PRINT*,' CALCULATING THE BACKGROUND DISTRIBUTION '
      PRINT*,' '
      CALL
BACKGROUND(D,S,B,FDEF,IG,DBKG,FBKG,DAVG,FAVG,HH,FBKGD,X,Y,X0,
     1Y0)
      CALL CHISSQ(D,EBKG,S,FDEF,B,CSQBKG)
      PRINT*,' AVERAGE COUNT RATE (cpm) = ',DAVG
      PRINT*,' '
      PRINT*,' AVERAGE ACTIVITY (', UNITS, ') = ', FAVG
      PRINT*,' '
      PRINT*,' BACKGROUND AVERAGE COUNT RATE (cpm) = ',DBKG
      PRINT*,' '
      PRINT*,' BACKGROUND AVERAGE ACTIVITY (', UNITS, ') = ', FBKG
      PRINT*,' '
      RCSQBKG = CSQBKG/REAL(M)
      PRINT*,' CHI SQUARE PER DEGREE OF FREEDOM/ BACKGROUND = ',RCSQBKG
      PRINT*,' '
C
C     Do a MaxEnt deconvolution with NXY>>M  number of cells
C
      PRINT*,' CALCULATING THE MAXIMUM ENTROPY DISTRIBUTION '
      PRINT*,' '
      CALL LAMBDAZERO(LAMBDANXY,M)
      CALL MAXENTNXY(LAMBDANXY,D,S,B,FDEF,FMENXY,CSQBKG)
      CALL CHISSQ(D,EMENXY,S,FMENXY,B,CSQMENXY)
C
C     If the first choice of initial lambdas did not lead to a
C     succesful deconvolutions, choose a new set of labdas and
C     try again.  Note: lambdas have units of 1/B
C
      IF (CSQMENXY .GT. (M+SQRT(REAL(M))) ) THEN
       IF (CSQMENXY .EQ. CSQBKG) THEN
        DO 5 I=1,M
         LAMBDANXY(I) = 10.0*(FAVG/DAVG)*(EMENXY(I)-D(I))/S(I)
5       CONTINUE
        CALL MAXENTNXY(LAMBDANXY,D,S,B,FDEF,FMENXY,CSQBKG)
        CALL CHISSQ(D,EMENXY,S,FMENXY,B,CSQMENXY)
       END IF
      END IF
      IF (CSQMENXY .GT. (M+SQRT(REAL(M))) ) THEN
       PRINT*,' MAXIMUM ENTROPY DECONVOLUTION DID NOT CONVERGE '
       PRINT*,' '
      END IF
      RCSQMENXY = CSQMENXY/M
      PRINT*,' CHI SQUARE PER DEGREE OF FREEDOM / MaxEnt = ',RCSQMENXY
      PRINT*,' '
C
      IF (IX .EQ. 1) THEN
      CALL
OUTPUTLCS(PE,MNUM,DETNUM,DAVG,FAVG,DBKG,FBKG,EBKG,RCSQBKG,
     1RCSQMENXY,D,S,X,Y,FMENXY,EH,UNITS,DH,RH,IC1,IG,FBKGD,EMENXY,X0,
     1Y0,FDEF,MH)
      GO TO 999
      END IF
C
C     Look for hot spots
C
      PRINT*,' CALCULATING THE EXCESS OR DEFICIT OF ACTIVITY ASSOCIATED'
      PRINT*,' WITH EACH MEASUREMENT '
      PRINT*,' '
      CALL PEAKXELS(FBKG,FMENXY,PFE,X,Y)
      CALL
SETHOTSPOTPARAM(RKHS,RLHS,MHS,CSQHS,M,HSMAX,XHS,YHS,HSFMAP,
     1NXY,FBKG,XHSABS,YHSABS)
      DO 10 I=1,M
      IF (PFE(I).GT. 0.0) THEN
       IF (HH(I) .GT. 0.0) THEN
        PRINT*,' SEARCHING FOR HOT SPOT NEAR MEASUREMENT : ',MNUM(I)
        PRINT*,' '
        CALL HOTSPOTDEF(I,HSS,FBKG,D,S,B,X,Y,PFE,HSFDEF,FDEF,RKP,RLP,HH)
        CALL LAMBDASET(LAMBDANXY,LAMBDAHS,M)
        CALL MAXENTNXYHS(LAMBDAHS,D,S,B,HSFDEF,HSF)
        CALL CHISSQ(D,HSE,S,HSF,B,CSQ)
        CALL HOTSPOTPARAM(HSF,RKP,RLP,HHP,CSQ,RKHS,RLHS,MHS,CSQHS,
     1HSMAX,IHP,FBKG,HSS,I,XHS,YHS,X0,Y0,XHSABS,YHSABS)
        RCSQHSI = CSQHS(I)/M
        PRINT*,' POTENTIAL HOT SPOT/MEASUREMENT NUMBER = ',MNUM(I)
        PRINT*,' POTENTIAL HOT SPOT/X COORDINATE = ',XHSABS(I)
        PRINT*,' POTENTIAL HOT SPOT/Y COORDINATE = ',YHSABS(I)
        PRINT*,' POTENTIAL HOT SPOT/MAGNITUDE (x BKG) = ',MHS(I)
        PRINT*,' POTENTIAL HOT SPOT/CHI SQUARE PER DEGREE OF FREEDOM = '
     1,RCSQHSI
        PRINT*,' '
        CALL MAKEHSMAP(RKP,RLP,HSFMAP,HHP,X,Y,HSS,FBKG)
       END IF
      END IF
10    CONTINUE
C
      RCSQHSI = CSQHS(IHP)/M
      PRINT*,' PARAMETERS OF THE LARGEST POTENTIAL HOT SPOT FOUND: '
      PRINT*,' '
      PRINT*,' MEASUREMENT NUMBER = ',MNUM(IHP)
      PRINT*,' X COORDINATE = ',XHSABS(IHP)
      PRINT*,' Y COORDINATE = ',YHSABS(IHP)
      PRINT*,' MAGNITUDE (x BKG) = ',MHS(IHP)
      PRINT*,' CHI SQUARE PER DEGREE OF FREEDOM  = ',RCSQHSI
      PRINT*,' '
C
      CALL
OUTPUT(PE,IHP,MNUM,DETNUM,DAVG,FAVG,DBKG,FBKG,EBKG,RCSQBKG,
     1RCSQMENXY,HSA,D,S,X,Y,MHS,CSQHS,FMENXY,HSFMAP,EH,UNITS,
     1DH,RH,PFE,XHSABS,YHSABS,IC1,IG,FBKGD,EMENXY,X0,Y0,MH)
C
999   PRINT*,' NAME OF OUTPUT FILE: ',EH
      PRINT*,' '
C
      RETURN
      END
C
C     ***********************************************************
C
      SUBROUTINE CHISSQ(D,E,S,F,B,CSQ)
C
      INTEGER I,L,M,NXY,MNXY
      REAL CSQ
      REAL D(M),E(M),S(M),F(NXY),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      CSQ = 0.0
      DO 2 I=1,M
        E(I) = 0.0
        DO 1 L=1,NXY
          E(I) = E(I) + B(NXY*(I-1)+L)*F(L)
1       CONTINUE
      CSQ = CSQ + ((D(I)-E(I))**2)/(S(I)**2)
2     CONTINUE

C
      RETURN
      END
C
C     ***********************************************************
C
      SUBROUTINE
INPUT(DH,EH,UNITS,HSA,HSS,HH,X,Y,D,S,DETNUM,PE,MNUM,IG,

1NSD,IQ,GPD,AA,AS,NOP,ANG1,ANG2,ANG3,DBKG,RH,RHO,X0,Y0,IX,FBKGD,MH)
C
      INTEGER M,NX,NY,NXY,MNXY,ISP,DETNUM,PE,IG,NSD,IQ,IX
      INTEGER MVB,NXVB,NYVB,IDETNUM,IPE,NLINES,INT1,INT2,HSS
      INTEGER MNUM(1000)
      REAL A,HSA,AVB,DBKG,RHO,X0,Y0,AGRID
      REAL RE1,RE2,RE3,RE4,RE5,GPD,AA,AS,NOP,ANG1,ANG2,ANG3
      REAL D(1000),S(1000),HH(1000),X(1000),Y(1000)
      REAL EA(100),ES(100),AMAC(100),SMAC(100),NAMAC,NSMAC,RHOAIR
      CHARACTER DH*60, RH*60, EH*8, UNITS*10, FBKGD*60, MH*60
      CHARACTER INPUTFILE*40
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C   next approx. 15 lines added 8/3/2000
C
C       '123456789012345678901234567890123456789012345678901234567890'
C
      DH ='                                                            '
C
C   DETERMINE NAME OF INPUT FILE (COMMAND LINE ARGUMENT OR CON)
C   I5=7 FOR READING OF A INPUT FILE
C   I5=5 FOR READING TERMINAL INPUT
C
      CALL NAMEOF(INPUTFILE,I5)
C
      IF(I5.EQ.7)
     1OPEN(i5,FILE=INPUTFILE,STATUS='OLD',FORM='FORMATTED')
C
C   Enter data about the source distribution
C
      PRINT*,' '
      PRINT*,' ENTER YOUR CHOICE OF DECONVOLUTION, '
      PRINT*,' 1 FOR "CHARACTERIZATION" (TO ESTIMATE THE DISTRIBUTION)'
      PRINT*,' 2 FOR "POST-REMEDIATION" (TO SEARCH FOR HOT SPOTS) : '
      READ(I5,*) IX
      IF (I5 .EQ. 7) THEN
        PRINT*,",IX
      END IF
      PRINT*,' '
C
C   Enter data about the names of the input/output files
C
      PRINT*,' NAME OF FILE WITH INPUT DATA? : '
      READ(I5,'(A60)') DH
      IF (I5 .EQ. 7) THEN
        PRINT*,' ',DH
      END IF
      PRINT*,' NAME OF OUTPUT FILE? (USE 8 CHARACTERS) : '
      READ(I5,'(A8)') EH
      IF (I5 .EQ. 7) THEN
        PRINT*,' ',EH
      END IF
      PRINT*,' '
C
      PRINT*,' ENTER YOUR CHOICE OF SOURCE DEPTH PROFILE, '
      PRINT*,' 1 FOR UNIFORM DISTRIBUTION '
      PRINT*,' 2 FOR PLANE DISTRIBUTION '
      READ(I5,*) NSD
```

```
      IF (I5 .EQ. 7) THEN
        PRINT*,",NSD
      END IF
      PRINT*,' '
C
      IF (NSD .EQ. 1 ) THEN
       PRINT*,' ENTER YOUR CHOICE OF UNITS FOR THE ACTIVITY PER MASS,'
       PRINT*,' (NOTE: IF YOU INCLUDED DATA FROM SOIL SAMPLES, YOU MUST'
       PRINT*,' CHOOSE THE SAME UNITS THAT YOU USED FOR THIS DATA) '
       PRINT*,' 1 FOR Bq/kg   '
       PRINT*,' 2 FOR pCi/g : '
        READ(I5,*) IQ
        IF (I5 .EQ. 7) THEN
          PRINT*,",IQ
        END IF
        PRINT*,' '
        IQ = IQ-1
        UNITS = ' Bq/kg'
        IF (IQ .EQ. 1) THEN
          UNITS = 'pCi/g'
        END IF
      ELSE IF (NSD .EQ. 2) THEN
       PRINT*,' ENTER YOUR CHOICE OF UNITS FOR THE ACTIVITY PER AREA,'
       PRINT*,' (NOTE: IF YOU INCLUDED DATA FROM SOIL SAMPLES, YOU MUST'
       PRINT*,' CHOOSE THE SAME UNITS THAT YOU USED FOR THIS DATA) '
       PRINT*,' 1 FOR Bq/m**2   '
       PRINT*,' 2 FOR pCi/m**2 : '
        READ(I5,*) IQ
        IF (I5 .EQ. 7) THEN
          PRINT*,",IQ
        END IF
        PRINT*,' '
        IQ = IQ-1
        UNITS = ' Bq/m**2'
        IF (IQ .EQ. 1) THEN
          UNITS = 'pCi/m**2'
        END IF
      END IF
C
      PRINT*,' ENTER YOUR CHOICE OF METHOD TO DETERMINE THE
BACKGROUND,'
      IF (IX .EQ. 1) THEN
        PRINT*,' 1 TO USE THE DATA TO ESTIMATE A SMOOTH BACKGROUND '
        PRINT*,' 2 TO READ IN BACKGROUND DATA FROM A FILE : '
        READ(I5,*) IG
        IF (I5 .EQ. 7) THEN
          PRINT*,",IG
        END IF
        IF (IG .EQ. 1) THEN
          IG = 4
        ELSE IF (IG .EQ. 2) THEN
          IG = 3
        END IF
      ELSE IF (IX .EQ. 2) THEN
        PRINT*,' 1 TO USE THE DATA TO ESTIMATE A CONSTANT BACKGROUND '
        PRINT*,' 2 TO ENTER A BACKGROUND COUNT RATE : '
        READ(I5,*) IG
        IF (I5 .EQ. 7) THEN
          PRINT*,' ',IG
        END IF
      END IF
      PRINT*,' '
      IF (IG .EQ. 2) THEN
        PRINT*,' ENTER COUNT RATE (>0) IN cpm : '
        READ(I5,*) DBKG
        IF (I5 .EQ. 7) THEN
          PRINT*,",DBKG
        END IF
        PRINT*,' '
      END IF
      IF (IG .EQ. 3) THEN
        PRINT*,' NAME OF FILE WITH BACKGROUND DATA? : '
        READ(I5,*) FBKGD
        IF (I5 .EQ. 7) THEN
          PRINT*,",FBKGD
        END IF
        PRINT*,' '
      END IF
C
C     The measurement data is read in from the file DH:
C     (1) M is the number of measurements.
C     (2) NX and NY are the number of cells used to model the ground, in
C         the X and Y direction
C     (3) A is the size of each cell (in meters).
C     (4) ISP is a parameter that is set up equal to one half the number
C         of cells between measurements (or slightly more if the number
C         of cells between is odd.
C     (5) DETNUM is the detector number.
C     (6) PE is an INTEGER, the photon energy.
C     (7) MNUM is the measurement number.  MUST HAVE MNUM(I) > 0.
C     (8) HH(I) is the detector height (in meters).
C     (8) The X(I),Y(I) are the coordinates of the measurements (in
C         meters).
C     (10) The D(I) are the measurements (in cpm).
C     (11) The S(I) are the corresponding sigmas (in cpm)for each
C          measurement.
C
      OPEN (11, FILE=DH, STATUS='OLD')
C
      READ (11,FMT=*) MVB
```

```
      READ (11,FMT=*) NXVB,NYVB,X0,Y0
      READ (11,FMT=*) AVB,AGRID
      READ (11,FMT=*) DETNUM,MH,RH
      READ (11,FMT=*) PE
      DO 10 I=1,MVB
        READ (11,FMT=*) MNUM(I),HH(I),X(I),Y(I),D(I),S(I)
10    CONTINUE
      CLOSE (11)
C
C     Convert the X(I),Y(I) to relative distance
C
      DO 15 I=1,MVB
        X(I) = X(I) - X0 + AVB
        Y(I) = Y(I) - Y0 + AVB
15    CONTINUE
C
      A = AVB
      ISP = CEILING(0.5*AGRID/AVB)
      M = MVB
      NX = NXVB
      NY = NYVB
      MNXY = M*NX*NY
      NXY = NX*NY
C
C     Enter data about the elevated area
C
      HSA = 0.0
      IF (IX .EQ. 2) THEN
        PRINT*,' AREA OF HOT SPOT? (IN SQUARE METERS, e.g.: 1.0) : '
        READ(I5,*) HSA
        IF (I5 .EQ. 7) THEN
          PRINT*,",HSA
        END IF
        IF (HSA .LE. (A**2)) THEN
          HSA = A**2
          PRINT*,' NOTE: AREA OF HOT SPOT CAN NOT BE LESS THAN : ',HSA
          PRINT*,' '
        END IF
        HSS = FLOOR((SQRT(HSA))/A)
        HSA = (A*HSS)**2
        PRINT*,' AREA OF HOT SPOT (IN SQUARE METERS) SET TO : ',HSA
        PRINT*,' '
      END IF
C
C     Look up information needed to calculate the detector response
C
C     GPD are gamma per disintegration
C     NOP is in cpm/(gamma/cm**2.second)
c    ANG2, ANG3 are parameters for the angular response function,
C      which is of the form
```

```
C      1.0+ANG1*(THETA)+ANG2*(THETA)^2+ANG3*(THETA)^3,
C      where THETA is measured in degrees.
C     AA is the mass attenuation coefficient for air, in cm**2/g
C     AS is the mass attenuation coefficient for soil, in cm**2/g
C     RHO is the soil density, in g/cm**3
C
      IDETNUM = 0
      IPE = 0
      OPEN (12, FILE=RH, STATUS='OLD')
C
      READ (12,FMT=*) NLINES
      DO 20 I =1,NLINES
       READ (12,FMT=*) INT1,INT2,RE1,RE2,RE3,RE4,RE5
       IF (INT1 .EQ. DETNUM) THEN
        IDETNUM = 1
        IF (INT2 .EQ. PE) THEN
         GPD = RE1
         NOP = RE2
         ANG1 = RE3
         ANG2 = RE4
         ANG3 = RE5
         IPE = 1
        END IF
       END IF
20     CONTINUE
      CLOSE (12)
C
      IF (IDETNUM .EQ. 0) THEN
       PRINT*,' '
       PRINT*,'  THE NUMBER OF THE DETECTOR DOES NOT MATCH ANY OF THE '
       PRINT*,'  DETECTOR NUMBERS LISTED IN THE SUBROUTINE RESPONSE. '
       PRINT*,' '
       PRINT*,'  PLEASE CHECK YOUR INPUT FILE. '
       STOP
      END IF
      IF (IPE .EQ. 0) THEN
       PRINT*,' '
       PRINT*,'  THE PHOTON ENERGY DOES NOT MATCH ANY OF THE ENERGIES'
       PRINT*,'  LISTED FOR THIS DETECTOR IN THE SUBROUTINE RESPONSE. '
       PRINT*,' '
       PRINT*,'  PLEASE CHECK YOUR INPUT FILE. '
       STOP
      END IF
C
C     Read in table of mass attenuation coefficient for air and soil
C
      OPEN (13, FILE=MH, STATUS='OLD')
      READ (13,FMT=*) NAMAC,RHOAIR
      DO 22 I =1,NAMAC
       READ (13,FMT=*) EA(I),AMAC(I)
22     CONTINUE
      READ (13,FMT=*) NSMAC,RHO
      DO 24 I =1,NSMAC
       READ (13,FMT=*) ES(I),SMAC(I)
24     CONTINUE
      CLOSE (13)
C
C     Interpolate the mass attenuation coefficients linearly in log-log
C
      DO 26 I=2,NAMAC
       IF (PE .GE. EA(I-1)) THEN
        IF (PE .LT. EA(I)) THEN
      AA = LOG(AMAC(I-1))
      AA = AA+LOG(PE/EA(I-1))*LOG(AMAC(I)/AMAC(I-1))/LOG(EA(I)/EA(I-1))
      AA = EXP(AA)
        END IF
       END IF
26     CONTINUE
C
      DO 28 I=2,NSMAC
       IF (PE .GE. ES(I-1)) THEN
        IF (PE .LT. ES(I)) THEN
      AS = LOG(SMAC(I-1))
      AS = AS+LOG(PE/ES(I-1))*LOG(SMAC(I)/SMAC(I-1))/LOG(ES(I)/ES(I-1))
      AS = EXP(AS)
        END IF
       END IF
28     CONTINUE
c
C     Multiply the mass attenuation coefficients by the density
C
      AA = AA*RHOAIR
      AS = AS*RHO
C
      PRINT*,' VALUES OF MASS ATTENUATION COEFFICIENTS OBTAINED BY '
      PRINT*,' INTERPOLATION USING DATA FROM THE FILE macoeff.csv: '
      PRINT*,' attenuation coefficient for air  (1/cm) = ',AA
      PRINT*,' attenuation coefficient for soil (1/cm) = ',AS
      PRINT*,' '
      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE
BACKGROUND(D,S,B,FDEF,IG,DBKG,FBKG,DAVG,FAVG,HH,FBKGD,
     1X,Y,X0,Y0)
C
C     FILE: BACKGR01.FOR
C     Modified 5/10/99
```

```
C
      INTEGER I,K,L,M,NX,NY,NXY,MNXY,IG,KK,LL,ISP,MD
      REAL SD,DAVG,SB,FAVG,SUM1,SUM2,DBKG,FBKG,MIS,XFILE,YFILE,ZFILE
      REAL ZMIN,MR,A,X0,Y0
      REAL D(M),S(M),B(MNXY),BI(M),FDEF(NXY),HH(M),EBKG(M),X(M),Y(M)
      CHARACTER FBKGD*60

      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      MR = REAL(M)
C
      MIS = 0.0
      SD = 0.0
      DO 5 I=1,M
        IF (HH(I) .GT. 0.0) THEN
          MIS = MIS + 1
          SD = SD + D(I)
        END IF
5      CONTINUE
      DAVG = SD/MIS
C
      SB = 0.0
      DO 10 I=1,M
        IF (HH(I) .GT. 0.0) THEN
          DO 8 K=1,NX
            DO 7 L=1,NY
              SB = SB + B(NXY*(I-1)+NY*(K-1)+L)
7            CONTINUE
8          CONTINUE
        END IF
10    CONTINUE
      FAVG = SD/SB
C
      FBKG = FAVG
C
      IF (IG .EQ. 1) THEN
        DO 16 I=1,M
         IF (HH(I) .GT. 0.0) THEN
          BI(I) = 0.0
          DO 14 K=1,NX
            DO 12 L=1,NY
              BI(I) = BI(I) + B(NXY*(I-1)+NY*(K-1)+L)
12            CONTINUE
14          CONTINUE
         END IF
16      CONTINUE
        SUM1 = 0.0
        SUM2 = 0.0
        DO 18 I=1,M
         IF (HH(I) .GT. 0.0) THEN
          SUM1 = SUM1 + D(I)*BI(I)/(S(I)**2)
          SUM2 = SUM2 + (BI(I)/S(I))**2
         END IF
18      CONTINUE
        FBKG = SUM1/SUM2
        DBKG = DAVG*(FBKG/FAVG)
      ELSE IF (IG .EQ. 2) THEN
        FBKG = FAVG*(DBKG/DAVG)
      END IF
      DO 20 K=1,NXY
        FDEF(K) = FBKG
20    CONTINUE
C
      IF (IG .EQ. 3) THEN
        FBKG = 0.0
        OPEN (11, FILE=FBKGD, STATUS='OLD')
        READ (11,FMT=*) ZMIN
        DO 22 K=1,NXY
          READ (11,FMT=*) XFILE,YFILE,ZFILE
          KK = INT((XFILE-X0)/A) + 1
          LL = INT((YFILE-Y0)/A) + 1
          IF (ZFILE .LT. ZMIN) THEN
            ZFILE = ZMIN
          END IF
          FDEF(NY*(KK-1)+LL) = ZFILE
          FBKG = FBKG + ZFILE/NXY
22      CONTINUE
        CLOSE (11)
        DBKG = 0.0
        DO 26 I=1,M
         EBKG(I) = 0.0
         DO 25 K=1,NX
          DO 24 L=1,NY
           EBKG(I) = EBKG(I) + B(NXY*(I-1)+NY*(K-1)+L)*FDEF(NY*(K-1)+L)
24          CONTINUE
25         CONTINUE
         DBKG = DBKG + EBKG(I)/MR
26      CONTINUE
      END IF
C
      IF (IG .EQ. 4) THEN
        MD = M+16
        CALL RBFINTER(MD,HH,X,Y,D,FDEF,DAVG,FAVG,B,S)
        FBKG = 0.0
        DO 33 K=1,NX
          DO 32 L=1,NY
            FBKG = FBKG + FDEF(NY*(K-1)+L)/NXY
32          CONTINUE
33      CONTINUE
```

```
      DBKG = 0.0
      DO 36 I=1,M
       EBKG(I) = 0.0
       DO 35 K=1,NX
        DO 34 L=1,NY
         EBKG(I) = EBKG(I) + B(NXY*(I-1)+NY*(K-1)+L)*FDEF(NY*(K-1)+L)
34        CONTINUE
35       CONTINUE
       DBKG = DBKG + EBKG(I)/MR
36      CONTINUE
      END IF
C
      RETURN
      END
C
C     *********************************************************
C
      SUBROUTINE
OUTPUT(PE,IHP,MNUM,DETNUM,DAVG,FAVG,DBKG,FBKG,EBKG,

1RCSQBKG,RCSQMENXY,HSA,D,S,X,Y,MHS,CSQHS,FMENXY,HSFMAP,EH,UNITS,
     1DH,RH,PFE,XHS,YHS,IC1,IG,FBKGD,EMENXY,X0,Y0,MH)
C
      INTEGER M,NX,NY,MNXY,NXY,KU,KL,LU,LL,ISP,PE,IHP,DETNUM,IG
      INTEGER IC1,IC2,ICR2,ICM2
      INTEGER MNUM(M)
      REAL A,DAVG,FAVG,DBKG,FBKG
      REAL RCSQBKG,RCSQMENXY,HSA,EAM
      REAL DTIME,XK,YL,X0,Y0,RCSQHSI
      REAL RERR(M),D(M),S(M),X(M),Y(M),MHS(M),CSQHS(M),EBKG(M)
      REAL PFE(M),EMENXY(M)
      REAL FMENXY(NXY),HSFMAP(NXY),XHS(M),YHS(M)
      CHARACTER EH*8,TH*12,WH*12,ZH*12,UNITS*10,DH*60,RH*60,MH*60
      CHARACTER FBKGD*60
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     Convert the X(I),Y(I) back to absolute distance
C
      DO 5 I=1,M
        X(I) = X(I) + X0 - A
        Y(I) = Y(I) + Y0 - A
5     CONTINUE
C
C     Prepare the activity maps
C
      KU = NX
      KL = 1
      LU = NY
      LL = 1
```

```
C
      EH = ADJUSTR(EH)
C
      ZH = EH//'.SCS'
      TH = EH//'.HSM'
C
      TH = ADJUSTL(TH)
      ZH = ADJUSTL(ZH)
C
      OPEN (12, FILE=ZH, STATUS='NEW')
      OPEN (14, FILE=TH, STATUS='NEW')
C
103   FORMAT(2X,A10)
C
      DO 25 K=KL,KU
        DO 23 L=LL,LU
          XK = (K-1)*A + X0
          YL = (L-1)*A + Y0
          WRITE (12,106) XK,YL,FMENXY(NY*(K-1)+L)
          WRITE (14,106) XK,YL,HSFMAP(NY*(K-1)+L)
23        CONTINUE
25      CONTINUE
106   FORMAT( 3 (2X,F10.4) )
C
C
      CLOSE (12, STATUS='KEEP')
      CLOSE (14, STATUS='KEEP')
C
C     Prepare the output table
C
      WH = EH//'.TBL'
      WH = ADJUSTL(WH)
C
      OPEN (16, FILE=WH, STATUS='NEW')
C
      WRITE (16,110) DH
110   FORMAT(/2X,'NAME OF FILE WITH INPUT DATA : ',A)
      WRITE (16,111) EH
111   FORMAT(2X,'NAME OF OUTPUT FILE : ',A)
      WRITE (16,112) RH
112   FORMAT(2X,'NAME OF FILE WITH DETECTOR RESPONSE DATA : ',A)
      WRITE (16,117) MH
117   FORMAT(2X,'NAME OF FILE WITH ATTENUATION COEFFICIENTS DATA : ',A)
      IF (IG .EQ. 1) THEN
        WRITE (16,113)
      ELSE IF (IG .EQ. 2) THEN
        WRITE (16,114)
      ELSE IF (IG .EQ. 3) THEN
        WRITE (16,115) FBKGD
      END IF
```

```
113   FORMAT(/2X,'BACKGROUND DISTRIBUTION WAS CALCULATED FROM THE
DATA')
114   FORMAT(/2X,'BACKGROUND DISTRIBUTION WAS SET BY THE USER')
115   FORMAT(/2X,'NAME OF FILE WITH BACKGROUND DATA : ',A)

C
      WRITE (16, 119) DETNUM
119   FORMAT(/2X,'DETECTOR NUMBER : ',I10)
      WRITE (16, 120) PE
120   FORMAT(2X,'PHOTON ENERGY (KeV) = ',I5)
C
      WRITE (16, 125) DAVG
125   FORMAT(/2X,'AVERAGE COUNT RATE (cpm) = ',1P,E10.4)
      WRITE (16, 130) UNITS, FAVG
130   FORMAT(2X,'AVERAGE ACTIVITY (', A10, ') = ',1P,E10.4)
      WRITE (16, 135) DBKG
135   FORMAT(2X,'BACKGROUND AVERAGE COUNT RATE (cpm) : ',1P,E10.4)
      WRITE (16, 140) UNITS, FBKG
140   FORMAT(2X,'BACKGROUND AVERAGE ACTIVITY (', A10, ') : ',1P,E10.4)
C
      WRITE (16, 150) RCSQBKG
150   FORMAT(/2X,'CHI SQUARE PER DEGREE OF FREEDOM / BACKGROUND =
',1P,E
     110.4)
      WRITE (16, 156) RCSQMENXY
156   FORMAT(2X,'CHI SQUARE PER DEGREE OF FREEDOM / MaxEnt DIST. = ',F10
     1.4)
C
      WRITE (16, 160) MNUM(IHP)
160   FORMAT(/2X,'LARGEST POTENTIAL ELEVATED AREA NEAR MEASUREMENT
: '
     1,I3)
      WRITE (16, 165) HSA
165   FORMAT(2X,'AREA (m^2) = ',F10.4)
      WRITE (16, 175) MHS(IHP)
175   FORMAT(2X,'MAGNITUDE (x BACKGROUND) = ',F10.4)
      EAM = MHS(IHP)*FBKG
      WRITE (16, 180) UNITS,EAM
180   FORMAT(2X,'MAGNITUDE (', A10, ') = ',1P,E10.4)
      WRITE (16, 182)
182   FORMAT(2X,'CHI SQUARE PER DEGREE OF FREEDOM OF THE')
      WRITE (16, 183) CSQHS(IHP)
183   FORMAT(2X,'DISTRIBUTION WITH THE LARGEST HOT SPOT = ',F10.4)
C
      DO 330 I=1,M
        RERR(I) = (EMENXY(I)-D(I))/S(I)
330   CONTINUE

      WRITE (16, 385)
      WRITE (16, 386)
```

```
      WRITE (16, 387)
385   FORMAT(/2X,'RESULTS FOR MaxEnt DECONVOLUTION, SMALL CELL SIZE:')
386   FORMAT(2X,'MEASURMENT  MEASURED    CALCULATED  DEVIATION ')
387   FORMAT(2X,'NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.')
      DO 340 I=1,M
        WRITE (16, 390) MNUM(I),D(I),EMENXY(I),RERR(I)
340    CONTINUE
390   FORMAT(2X,I10,2X,1P,E10.4,2X,E10.4,2X,0P,F10.5)
C
      DO 430 I=1,M
        RERR(I) = (EBKG(I)-D(I))/S(I)
430   CONTINUE

      WRITE (16, 485)
      WRITE (16, 486)
      WRITE (16, 487)
485   FORMAT(/2X,'RESULTS FOR DEFAULT DISTRIBUTION:')
486   FORMAT(2X,'MEASURMENT  MEASURED    CALCULATED  DEVIATION ')
487   FORMAT(2X,'NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.')
      DO 440 I=1,M
        WRITE (16, 490) MNUM(I),D(I),EBKG(I),RERR(I)
440    CONTINUE
490   FORMAT(2X,I10,2X,1P,E10.4,2X,E10.4,2X,0P,F10.5)

C
      WRITE (16, 195)
      WRITE (16, 196)
195   FORMAT(/2X,'POTENTIAL HOT SPOTS FOUND:')
196   FORMAT(2X,'MEAS. N.  X COORD.  Y COORD.  MAG(xBKG) CHI SQR.')
C
      DO 50 I=1,M
        IF (PFE(I) .GT. 0.0) THEN
          RCSQHSI = CSQHS(I)/M
          WRITE (16, 200) MNUM(I),XHS(I),YHS(I),MHS(I),RCSQHSI
        END IF
50    CONTINUE
200   FORMAT(2X,I8,2X,1P,E8.2,2X,E8.2,2X,E8.2,2X,E8.2)
C
      CALL SYSTEM_CLOCK(IC2,ICR2,ICM2)
      IF (ICR2 .GT. 0) THEN
        DTIME = ((IC2 - IC1)/ICR2)/60.0
        WRITE (16, 205) DTIME
205     FORMAT(//2X,'TIME TO RUN THE PROGRAM (MINUTES) = ',F10.4)
      END IF
      CLOSE (16, STATUS='KEEP')
C
      RETURN
      END
C
C     ********************************************************
```

```
C
      SUBROUTINE
OUTPUTLCS(PE,MNUM,DETNUM,DAVG,FAVG,DBKG,FBKG,EBKG,
     1RCSQBKG,RCSQMENXY,D,S,X,Y,FMENXY,EH,UNITS,DH,RH,IC1,IG,FBKGD,
     1EMENXY,X0,Y0,FDEF,MH)
C
      INTEGER M,NX,NY,MNXY,NXY,KU,KL,LU,LL,ISP,PE,DETNUM,IG
      INTEGER IC1,IC2,ICR2,ICM2
      INTEGER MNUM(M)
      REAL A,DAVG,FAVG,DBKG,FBKG
      REAL RCSQBKG,RCSQMENXY
      REAL DTIME,XK,YL,X0,Y0
      REAL RERR(M),D(M),S(M),X(M),Y(M),EBKG(M)
      REAL EMENXY(M)
      REAL FMENXY(NXY)
      REAL FDEF(NXY)
      CHARACTER EH*8,WH*12,ZH*12,UNITS*10,DH*60,RH*60,QH*12,MH*60
      CHARACTER FBKGD*60
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     Prepare the activity maps
C
C     Convert the X(I),Y(I) back to absolute distance
C
      DO 5 I=1,M
        X(I) = X(I) + X0 - A
        Y(I) = Y(I) + Y0 - A
5     CONTINUE
C
      KU = NX
      KL = 1
      LU = NY
      LL = 1
C
      EH = ADJUSTR(EH)
C
      ZH = EH//'.SCS'
      QH = EH//'.DEF'
C
      ZH = ADJUSTL(ZH)
      QH = ADJUSTL(QH)
C
      OPEN (12, FILE=ZH, STATUS='NEW')
      OPEN (14, FILE=QH, STATUS='NEW')
C
103   FORMAT(2X,A10)
C
      DO 25 K=KL,KU
        DO 23 L=LL,LU
          XK = (K-1)*A + X0
          YL = (L-1)*A + Y0
          WRITE (12,106) XK,YL,FMENXY(NY*(K-1)+L)
          WRITE (14,106) XK,YL,FDEF(NY*(K-1)+L)
23        CONTINUE
25      CONTINUE
106   FORMAT( 3 (2X,F10.4) )
C
      CLOSE (12, STATUS='KEEP')
      CLOSE (14, STATUS='KEEP')
C
C     Prepare the output table
C
      WH = EH//'.TBL'
      WH = ADJUSTL(WH)
C
      OPEN (16, FILE=WH, STATUS='NEW')
C
      WRITE (16,110) DH
110   FORMAT(/2X,'NAME OF FILE WITH INPUT DATA : ',A)
      WRITE (16,111) EH
111   FORMAT(2X,'NAME OF OUTPUT FILE : ',A)
      WRITE (16,112) RH
112   FORMAT(2X,'NAME OF FILE WITH DETECTOR RESPONSE DATA : ',A)
      WRITE (16,117) MH
117   FORMAT(2X,'NAME OF FILE WITH ATTENUATION COEFFICIENTS DATA : ',A)
      IF (IG .EQ. 1) THEN
        WRITE (16,113)
      ELSE IF (IG .EQ. 2) THEN
        WRITE (16,114)
      ELSE IF (IG .EQ. 3) THEN
        WRITE (16,115) FBKGD
      ELSE IF (IG .EQ. 4) THEN
        WRITE (16,116)
      END IF
113   FORMAT(/2X,'CONSTANT BACKGROUND WAS ESTIMATED FROM THE
DATA')
114   FORMAT(/2X,'BACKGROUND DISTRIBUTION WAS SET BY THE USER')
115   FORMAT(/2X,'NAME OF FILE WITH BACKGROUND DATA : ',A)
116   FORMAT(/2X,'SMOOTH BACKGROUND WAS ESTIMATED FROM THE DATA')
C
      WRITE (16, 119) DETNUM
119   FORMAT(/2X,'DETECTOR NUMBER : ',I10)
      WRITE (16, 120) PE
120   FORMAT(2X,'PHOTON ENERGY (KeV) = ',I5)
C
      WRITE (16, 125) DAVG
125   FORMAT(/2X,'AVERAGE COUNT RATE (cpm) = ',1P,E10.4)
      WRITE (16, 130) UNITS, FAVG
130   FORMAT(2X,'AVERAGE ACTIVITY (', A10, ') = ',1P,E10.4)
```

```
      WRITE (16, 135) DBKG
135   FORMAT(2X,'BACKGROUND AVERAGE COUNT RATE (cpm) : ',1P,E10.4)
      WRITE (16, 140) UNITS, FBKG
140   FORMAT(2X,'BACKGROUND AVERAGE ACTIVITY (', A10, ') : ',1P,E10.4)
C
      WRITE (16, 150) RCSQBKG
150   FORMAT(/2X,'CHI SQUARE PER DEGREE OF FREEDOM / BACKGROUND =
',1P,E
     110.4)
      WRITE (16, 156) RCSQMENXY
156   FORMAT(2X,'CHI SQUARE PER DEGREE OF FREEDOM / MAXIMUM ENTROPY
= ',
     1F10.4)
C
      DO 330 I=1,M
        RERR(I) = (EMENXY(I)-D(I))/S(I)
330   CONTINUE

      WRITE (16, 385)
      WRITE (16, 386)
      WRITE (16, 387)
385   FORMAT(/2X,'RESULTS FOR MAXIMUM ENTROPY DECONVOLUTION :')
386   FORMAT(2X,'MEASURMENT  MEASURED    CALCULATED  DEVIATION ')
387   FORMAT(2X,'NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.')
      DO 340 I=1,M
        WRITE (16, 390) MNUM(I),D(I),EMENXY(I),RERR(I)
340    CONTINUE
390   FORMAT(2X,I10,2X,1P,E10.4,2X,E10.4,2X,0P,F10.5)
C
      DO 430 I=1,M
        RERR(I) = (EBKG(I)-D(I))/S(I)
430   CONTINUE

      WRITE (16, 485)
      WRITE (16, 486)
      WRITE (16, 487)
485   FORMAT(/2X,'RESULTS FOR DEFAULT DISTRIBUTION:')
486   FORMAT(2X,'MEASURMENT  MEASURED    CALCULATED  DEVIATION ')
487   FORMAT(2X,'NUMBER      COUNT RATE  COUNT RATE  (C.-M.)/S.')
      DO 440 I=1,M
        WRITE (16, 490) MNUM(I),D(I),EBKG(I),RERR(I)
440    CONTINUE
490   FORMAT(2X,I10,2X,1P,E10.4,2X,E10.4,2X,0P,F10.5)
C
      CALL SYSTEM_CLOCK(IC2,ICR2,ICM2)
      IF (ICR2 .GT. 0) THEN
        DTIME = ((IC2 - IC1)/ICR2)/60.0
        WRITE (16, 205) DTIME
205     FORMAT(//2X,'TIME TO RUN THE PROGRAM (MINUTES) = ',F10.4)
      END IF
```

```
      CLOSE (16, STATUS='KEEP')
C
      RETURN
      END
C
C     ******************************************************
C
      SUBROUTINE DETRESP(B,A,X,Y,NSD,IQ,AA,NOP,GPD,AS,ANG1,ANG2,ANG3,
     1HH,RHO,M,NXY,MNXY,NX,NY)
C
C     FILE: RFFLAT01.FOR
C     LAST MODIFIED: 5/6/99
C
C     Calculates the instrument response ("blur") matrix B.
C
      INTEGER I,K,L,M,NXY,MNXY,IQ,NSD,NX,NY,KMIN,LMIN
      REAL
AA,RR,XK,XL,R,ASR,NOP,GPD,TH,ANGC,RHO,AS,PHIOA,ANG1,ANG2,ANG3
      REAL A,RMIN
      REAL B(MNXY),X(M),Y(M),HH(M)
C
C     NOTE: X(I),Y(I),A are measured in meters.
C     GPD are gamma per disintegration
C     NOP is in cpm/(gamma/cm**2.second)
c    ANG2, ANG3 are parameters for the angular response function,
C       which is of the form
C       1.0 + ANG1*(THETA) + ANG2*(THETA)^2 + ANG3*(THETA)^3
C       where THETA is measured in degrees.
C     AA is the attenuation coefficient for air, in units of 1/cm
C     AS is the attenuation coefficient for soil, in units of 1/cm
C     RHO is the soil density, in g/cm**3
C
C     Note: IF HH(I)=0.0, THEN DATA IS FROM A SOIL SAMPLE.
C           SOIL SAMPLE DATA MUST BE IN THE CONCENTRATION UNITS USED
C           FOR THE DECONVOLUTION.
C
      IF (NSD .EQ. 1) THEN
        IF (IQ .EQ. 1) THEN
          NOP = NOP/0.02703
        END IF
        DO 15 I=1,M
          RMIN = 10*A
          DO 10 K=1,NX
            DO 5 L=1,NY
              XK = A*(REAL(K))
              XL = A*(REAL(L))
              RR = ((X(I)-XK)**2)+((Y(I)-XL)**2)+((HH(I))**2)
              R = SQRT(RR)
              IF (HH(I) .GT. 0.0) THEN
                ASR = AS*(R*100.0)
```

```
           TH = 57.30*ACOS(HH(I)/R)
           ANGC = 1.0 + ANG1*TH + ANG2*(TH**2) + ANG3*TH*(TH**2)
           PHIOA = ((A*100.0)**2)*(RHO/1000.0)*(HH(I)/R)/12.566
           IF (ASR .GT. 10.0) THEN
            PHIOA = (PHIOA/(R*100.0))*(EXPZE2Z(ASR))
           ELSE
            PHIOA = (PHIOA/(R*100.0))*EXP(ASR)*(EXPINT(2,ASR))
           END IF
        B(NXY*(I-1)+NY*(K-1)+L) = EXP(-AA*R*100.0)*PHIOA*NOP*GPD*ANGC
          END IF
          IF (HH(I) .EQ. 0.0) THEN
           B(NXY*(I-1)+NY*(K-1)+L) = 0.0
           IF (R .LT. RMIN) THEN
            KMIN = K
            LMIN = L
            RMIN = R
           END IF
          END IF
5        CONTINUE
10       CONTINUE
        IF (HH(I) .EQ. 0.0) THEN
         B(NXY*(I-1)+NY*(KMIN-1)+LMIN) = 1.0
        END IF
15      CONTINUE
      ELSE IF (NSD .EQ. 2) THEN
       IF (IQ .EQ. 1) THEN
        NOP = NOP/27.03
       END IF
       DO 35 I=1,M
        DO 30 K=1,NX
         DO 25 L=1,NY
          XK = A*(REAL(K))
          XL = A*(REAL(L))
          RR = ((X(I)-XK)**2)+((Y(I)-XL)**2)+(HH(I)**2)
          R = SQRT(RR)
          TH = 57.30*ACOS(HH(I)/R)
          ANGC = 1.0 + ANG1*TH + ANG2*(TH**2) + ANG3*TH*(TH**2)
          PHIOA = (A**2)/(12.566*RR*10000.0)
        B(NXY*(I-1)+NY*(K-1)+L) = EXP(-AA*R*100.0)*PHIOA*NOP*GPD*ANGC
25        CONTINUE
30       CONTINUE
35      CONTINUE
      END IF
C
      RETURN
      END
C
C      Note: To calculate B use (see tutorial) the formula
C      N(cpm) = (air atten.)*(N0/PHI)*(gamma/dis)*(ang. correction)*PHI
C      For a uniform distribution use the following formula for PHI:
```

```
C      PHI =  {(area)*(rho-soil)*cos(theta)/4*PI}
C           * {exp{SA*R}*ExpInt2(SA*R)/R}
C           * {activity/mass}
C      where SA = soil attenuation, AA = air attenuation, R = distance
C      from source to detector, cos(theta) = HH/SQRT(RR)
C
C      ****************************************************************
C
      FUNCTION expint(n,x)
      INTEGER n,MAXIT
      REAL expint,x,EPS,FPMIN,EULER
      PARAMETER (MAXIT=100,EPS=1.e-7,FPMIN=1.e-30,EULER=.5772156649)
      INTEGER i,ii,nm1
      REAL a,b,c,d,del,fact,h,psi
      nm1=n-1
      if(n.lt.0.or.x.lt.0..or.(x.eq.0..and.(n.eq.0.or.n.eq.1)))then
       pause 'bad arguments in expint'
      else if(n.eq.0)then
       expint=exp(-x)/x
      else if(x.eq.0.)then
       expint=1./nm1
      else if(x.gt.1.)then
       b=x+n
       c=1./FPMIN
       d=1./b
       h=d
       do 11 i=1,MAXIT
        a=-i*(nm1+i)
        b=b+2.
        d=1./(a*d+b)
        c=b+a/c
        del=c*d
        h=h*del
        if(abs(del-1.).lt.EPS)then
         expint=h*exp(-x)
         return
        endif
11      continue
       pause 'continued fraction failed in expint'
      else
       if(nm1.ne.0)then
        expint=1./nm1
       else
        expint=-log(x)-EULER
       endif
       fact=1.
       do 13 i=1,MAXIT
        fact=-fact*x/i
        if(i.ne.nm1)then
         del=-fact/(i-nm1)
```

```
          else
            psi=-EULER
            do 12 ii=1,nm1
              psi=psi+1./ii
12            continue
            del=fact*(-log(x)+psi)
          endif
          expint=expint+del
          if(abs(del).lt.abs(expint)*EPS) return
13        continue
        pause 'series failed in expint'
      endif
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ***************************************************************
C
      FUNCTION EXPZE2Z(X)
C
C     Function to calculate exp(x)*E2(x), where E2(x) is an exponential
C     integral.  Uses the rational approximation by Paul Verbeeck,
C     "Rational approximations for exponential integrals En(x)", in
C     Bulletin de l'Academie royale de Belgique (Classe de Sciences),
C     page 1064, 5e Serie - Tome LVI, 1970.
C
C     exp(x)*E2(x) = (1/x)*(v1(z)/v2(z)) where v1,v2 are third order
C     polynomials, z=1/x, x>1, and stated maximum error < 9.E-06.
C
      REAL X,A2,A3,B2,B3
C
      A2 = 2.6731571
      A3 = 0.1212641
      B2 = 4.6660820
      B3 = 3.7340708
C
      EXPZE2Z = (1.0+(A2/X)+(A3/(X**2)))/(X*(1.0+(B2/X)+(B3/(X**2))))
C
      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE LAMBDAZERO(LAMBDA,M)

      INTEGER I,M
      REAL LAMBDA(M)

      DO 10 I=1,M
        LAMBDA(I) = 0.0
10    CONTINUE

      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE LAMBDASET(LAMBDAIN,LAMBDAOUT,M)

      INTEGER I,M
      REAL LAMBDAIN(M),LAMBDAOUT(M)

      DO 10 I=1,M
        LAMBDAOUT(I) = LAMBDAIN(I)
10    CONTINUE

      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE MAXENTNXY(LAMBDANXY,D,S,B,FDEF,FMENXY,CSQBKG)

      INTEGER M,NXY,MNXY
      INTEGER ITER
      REAL FTOL,FRET,CSQBKG,REALM
      REAL D(M),S(M)
      REAL LAMBDANXY(M),FMENXY(NXY),FDEF(NXY),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      FTOL = 1.0E-20
C
      REALM = REAL(M)
      IF (CSQBKG .LE. REALM) THEN
        DO 10 K=1,NXY
          FMENXY(K) = FDEF(K)
10        CONTINUE
      ELSE IF (CSQBKG .GT. REALM) THEN
        CALL FRPRMN(LAMBDANXY,M,FTOL,ITER,FRET,FDEF,D,S,B,M,NXY,MNXY)
        CALL LMBDAS(FTOL,LAMBDANXY,ITER,FDEF,D,S,B)
        CALL FRPRMN(LAMBDANXY,M,FTOL,ITER,FRET,FDEF,D,S,B,M,NXY,MNXY)
        CALL FMAXENTNXY(LAMBDANXY,FDEF,FMENXY,B)
      END IF
C
      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE MAXENTNXYHS(LAMBDAHS,D,S,B,FDEF,FMENXY)
      INTEGER M,NXY,MNXY
      INTEGER ITER
      REAL FTOL
C       REAL FRET
      REAL D(M),S(M)
      REAL LAMBDAHS(M),FMENXY(NXY),FDEF(NXY),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      FTOL = 1.0E-20
C
C       CALL FRPRMN(LAMBDAHS,M,FTOL,ITER,FRET,FDEF,D,S,B,M,NXY,MNXY)
      CALL LMBDAS(FTOL,LAMBDAHS,ITER,FDEF,D,S,B)
      CALL FMAXENTNXY(LAMBDAHS,FDEF,FMENXY,B)
C
      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE FMAXENTNXY(LAMBDANXY,FDEF,FMENXY,B)
C
      INTEGER I,L,M,NXY,MNXY
      REAL LAMBDANXY(M),FDEF(NXY),FMENXY(NXY),B(MNXY)
      REAL SUM(NXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      DO 20 L=1,NXY
        SUM(L) = 0.0
        DO 10 I=1,M
          SUM(L) = SUM(L) + LAMBDANXY(I)*B(NXY*(I-1)+L)
10        CONTINUE
20      CONTINUE
C
      DO 30 L=1,NXY
        FMENXY(L) = FDEF(L)*EXP(-SUM(L))
30      CONTINUE
C
      RETURN
      END
C
C     ********************************************************
C
C     FILE MEMIN01.FOR
C     2/19/98
C
      FUNCTION FUNC(LAMBDA,FDEF,D,S,B,M,NXY,MNXY)
C
      INTEGER L,M,NXY,MNXY
      REAL OMEGA,SUM1,SUM2,SUM3,SUM4,SUM5
```

```
      REAL LAMBDA(M),FDEF(NXY),D(M),S(M),B(MNXY)
C
      OMEGA = REAL(M)
C
      SUM1 = 0.0
      SUM5 = 0.0
      DO 12 L=1,NXY
        SUM2 = 0.0
        DO 10 I=1,M
          SUM2 = SUM2 + LAMBDA(I)*B(NXY*(I-1)+L)
10        CONTINUE
        IF (SUM2 .GT. 174.0) THEN
          SUM2 = 174.0
        END IF
        IF (SUM2 .LT. -180.0) THEN
          SUM2 = -180.0
        END IF
        SUM1 = SUM1 + FDEF(L)*EXP(-SUM2)
        SUM5 = SUM5 + FDEF(L)
12      CONTINUE
C
      SUM3 = 0.0
      SUM4 = 0.0
      DO 16 I=1,M
        SUM3 = SUM3 + (S(I)*LAMBDA(I))**2
        SUM4 = SUM4 + D(I)*LAMBDA(I)
16      CONTINUE
C
      FUNC = -SUM1-SQRT(OMEGA*SUM3)-SUM4+SUM5
      FUNC = -FUNC
C
      RETURN
      END
C
C     ***************************************************************
C
      SUBROUTINE DFUNC(LAMBDA,GRADZ,FDEF,D,S,B,M,NXY,MNXY)

      INTEGER L,M,NXY,MNXY
      REAL OMEGA,A1,A2,SUM1,SUM2(NXY),SUM3
      REAL LAMBDA(M),GRADZ(M),FDEF(NXY),D(M),S(M),B(MNXY)
C
      OMEGA = REAL(M)
C
      A1 = 0.0
      A2 = 0.0
      DO 8 I=1,M
        A1 = A1 + (S(I)*LAMBDA(I))**2
        A2 = A2 + (S(I))**2
8      CONTINUE
```

```
      DO 12 L=1,NXY
        SUM2(L) = 0.0
        DO 10 I=1,M
          SUM2(L) = SUM2(L) + LAMBDA(I)*B(NXY*(I-1)+L)
10        CONTINUE
12      CONTINUE
C
      DO 30 I=1,M
        SUM1 = 0.0
        DO 16 L=1,NXY
        IF (SUM2(L) .GT. 174.0) THEN
          SUM2(L) = 174.0
        END IF
        IF (SUM2(L) .LT. -180.0) THEN
          SUM2(L) = -180.0
        END IF
          SUM1 = SUM1 + FDEF(L)*EXP(-SUM2(L))*B(NXY*(I-1)+L)
16        CONTINUE
C
        SUM3 = ((S(I))**2)*(SQRT(OMEGA/A2))
        IF (A1 .GT. 1.0E-20) THEN
          SUM3 = ((S(I))**2)*LAMBDA(I)*(SQRT(OMEGA/A1))
        END IF
C
        GRADZ(I) = SUM1 - SUM3 - D(I)
        GRADZ(I) = -GRADZ(I)
30      CONTINUE
C
      RETURN
      END
C
C     ****************************************************************
C
      SUBROUTINE frprmn(p,n,ftol,iter,fret,FDEF,D,S,B,M,NXY,MNXY)
      INTEGER iter,n,NMAX,ITMAX
      REAL fret,ftol,p(n),EPS,func
      EXTERNAL func
      PARAMETER (NMAX=500,ITMAX=200,EPS=1.e-10)
CU    USES dfunc,func,linmin
      INTEGER its,j
      REAL dgg,fp,gam,gg,g(NMAX),h(NMAX),xi(NMAX)
      REAL FDEF(NXY),D(M),S(M),B(MNXY)
      fp=func(p,FDEF,D,S,B,M,NXY,MNXY)
      call dfunc(p,xi,FDEF,D,S,B,M,NXY,MNXY)
      do 11 j=1,n
        g(j)=-xi(j)
        h(j)=g(j)
        xi(j)=h(j)
11      continue
```

```
      do 14 its=1,ITMAX
        iter=its
C         PRINT*,' SUBROUTINE FRPRMN/NUMBER OF ITERATIONS : ',ITER
C         PRINT*,' '
        call linmin(p,xi,n,fret,FDEF,D,S,B,M,NXY,MNXY)
        if(2.*abs(fret-fp).le.ftol*(abs(fret)+abs(fp)+EPS))return
        fp=func(p,FDEF,D,S,B,M,NXY,MNXY)
        call dfunc(p,xi,FDEF,D,S,B,M,NXY,MNXY)
        gg=0.
        dgg=0.
        do 12 j=1,n
          gg=gg+g(j)**2
C           dgg=dgg+xi(j)**2
          dgg=dgg+(xi(j)+g(j))*xi(j)
12        continue
        if(gg.eq.0.)return
        gam=dgg/gg
        do 13 j=1,n
          g(j)=-xi(j)
          h(j)=g(j)+gam*h(j)
          xi(j)=h(j)
13        continue
14      continue
C        pause 'frprmn maximum iterations exceeded'
c       PRINT*,'  frprmn maximum iterations exceeded'
c       PRINT*,' '
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ****************************************************************
C
      SUBROUTINE linmin(p,xi,n,fret,FDEF,D,S,B,M,NXY,MNXY)
      INTEGER n,NMAX
      REAL fret,p(n),xi(n),TOL
      PARAMETER (NMAX=500,TOL=1.e-6)
CU    USES brent,f1dim,mnbrak
      INTEGER j,ncom
      REAL ax,bx,fa,fb,fx,xmin,xx,pcom(NMAX),xicom(NMAX),brent
      REAL FDEF(NXY),D(M),S(M),B(MNXY)
      COMMON /f1com/ pcom,xicom,ncom
      EXTERNAL f1dim
      ncom=n
      do 11 j=1,n
        pcom(j)=p(j)
        xicom(j)=xi(j)
11      continue
      ax=0.
      xx=1.
      call mnbrak(ax,xx,bx,fa,fx,fb,f1dim,FDEF,D,S,B,M,NXY,MNXY)
      fret=brent(ax,xx,bx,f1dim,TOL,xmin,FDEF,D,S,B,M,NXY,MNXY)
      do 12 j=1,n
        xi(j)=xmin*xi(j)
        p(j)=p(j)+xi(j)
12      continue
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ****************************************************************
C
      SUBROUTINE mnbrak(ax,bx,cx,fa,fb,fc,func,FDEF,D,S,B,M,NXY,MNXY)
      REAL ax,bx,cx,fa,fb,fc,func,GOLD,GLIMIT,TINY
      EXTERNAL func
      PARAMETER (GOLD=1.618034, GLIMIT=100., TINY=1.e-20)
      REAL dum,fu,q,r,u,ulim
      REAL FDEF(NXY),D(M),S(M),B(MNXY)
      fa=func(ax,FDEF,D,S,B,M,NXY,MNXY)
      fb=func(bx,FDEF,D,S,B,M,NXY,MNXY)
      if(fb.gt.fa)then
        dum=ax
        ax=bx
        bx=dum
        dum=fb
        fb=fa
        fa=dum
      endif
      cx=bx+GOLD*(bx-ax)
      fc=func(cx,FDEF,D,S,B,M,NXY,MNXY)
1     if(fb.ge.fc)then
        r=(bx-ax)*(fb-fc)
        q=(bx-cx)*(fb-fa)
        u=bx-((bx-cx)*q-(bx-ax)*r)/(2.*sign(max(abs(q-r),TINY),q-r))
        ulim=bx+GLIMIT*(cx-bx)
        if((bx-u)*(u-cx).gt.0.)then
          fu=func(u,FDEF,D,S,B,M,NXY,MNXY)
          if(fu.lt.fc)then
            ax=bx
            fa=fb
            bx=u
            fb=fu
            return
          else if(fu.gt.fb)then
            cx=u
            fc=fu
            return
          endif
          u=cx+GOLD*(cx-bx)
          fu=func(u,FDEF,D,S,B,M,NXY,MNXY)
        else if((cx-u)*(u-ulim).gt.0.)then
```

```
          fu=func(u,FDEF,D,S,B,M,NXY,MNXY)
          if(fu.lt.fc)then
            bx=cx
            cx=u
            u=cx+GOLD*(cx-bx)
            fb=fc
            fc=fu
            fu=func(u,FDEF,D,S,B,M,NXY,MNXY)
          endif
        else if((u-ulim)*(ulim-cx).ge.0.)then
          u=ulim
          fu=func(u,FDEF,D,S,B,M,NXY,MNXY)
        else
          u=cx+GOLD*(cx-bx)
          fu=func(u,FDEF,D,S,B,M,NXY,MNXY)
        endif
        ax=bx
        bx=cx
        cx=u
        fa=fb
        fb=fc
        fc=fu
        goto 1
      endif
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ****************************************************************
C
      FUNCTION brent(ax,bx,cx,f,tol,xmin,FDEF,DD,S,BB,M,NXY,MNXY)
      INTEGER ITMAX
      REAL brent,ax,bx,cx,tol,xmin,f,CGOLD,ZEPS
      EXTERNAL f
      PARAMETER (ITMAX=100,CGOLD=.3819660,ZEPS=1.0e-10)
      INTEGER iter
      REAL a,b,d,e,etemp,fu,fv,fw,fx,p,q,r,tol1,tol2,u,v,w,x,xm
      REAL FDEF(NXY),DD(M),S(M),BB(MNXY)
      a=min(ax,cx)
      b=max(ax,cx)
      v=bx
      w=v
      x=v
      e=0.
      fx=f(x,FDEF,DD,S,BB,M,NXY,MNXY)
      fv=fx
      fw=fx
      do 11 iter=1,ITMAX
        xm=0.5*(a+b)
        tol1=tol*abs(x)+ZEPS
      tol2=2.*tol1
      if(abs(x-xm).le.(tol2-.5*(b-a))) goto 3
      if(abs(e).gt.tol1) then
        r=(x-w)*(fx-fv)
        q=(x-v)*(fx-fw)
        p=(x-v)*q-(x-w)*r
        q=2.*(q-r)
        if(q.gt.0.) p=-p
        q=abs(q)
        etemp=e
        e=d
        if(abs(p).ge.abs(.5*q*etemp).or.p.le.q*(a-x).or.p.ge.q*(b-x))
     *goto 1
        d=p/q
        u=x+d
        if(u-a.lt.tol2 .or. b-u.lt.tol2) d=sign(tol1,xm-x)
        goto 2
      endif
1      if(x.ge.xm) then
        e=a-x
      else
        e=b-x
      endif
      d=CGOLD*e
2      if(abs(d).ge.tol1) then
        u=x+d
      else
        u=x+sign(tol1,d)
      endif
      fu=f(u,FDEF,DD,S,BB,M,NXY,MNXY)
      if(fu.le.fx) then
        if(u.ge.x) then
          a=x
        else
          b=x
        endif
        v=w
        fv=fw
        w=x
        fw=fx
        x=u
        fx=fu
      else
        if(u.lt.x) then
          a=u
        else
          b=u
        endif
        if(fu.le.fw .or. w.eq.x) then
          v=w
```

```
            fv=fw
            w=u
            fw=fu
          else if(fu.le.fv .or. v.eq.x .or. v.eq.w) then
            v=u
            fv=fu
          endif
        endif
11      continue
C        pause 'brent exceed maximum iterations'
c       PRINT*,'  brent exceed maximum iterations'
c       PRINT*,' '
3      xmin=x
      brent=fx
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ***************************************************************
C
      FUNCTION f1dim(x,FDEF,D,S,B,M,NXY,MNXY)
      INTEGER NMAX
      REAL f1dim,func,x
      PARAMETER (NMAX=500)
CU    USES func
      INTEGER j,ncom
      REAL pcom(NMAX),xicom(NMAX),xt(NMAX)
      REAL FDEF(NXY),D(M),S(M),B(MNXY)
      COMMON /f1com/ pcom,xicom,ncom
      do 11 j=1,ncom
        xt(j)=pcom(j)+x*xicom(j)
11      continue
      f1dim=func(xt,FDEF,D,S,B,M,NXY,MNXY)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ***************************************************************
C
      SUBROUTINE LMBDAS(FTOL,LAMBDA,ITER,FDEF,D,S,B)
C
C     FILE: MEITER01.FOR
C     2/19/98
C
      INTEGER K,L,M,NX,NY,NXY,ITER,ITERMAX,MNXY
      REAL OMEGA,LB(NX,NY),EPS
      REAL TWOMU,Z,ZOLD,FTOL,P,C
      REAL LAMBDA(M),LAMBDAOLD(M),DL,DLMIN
      REAL FDEF(NXY),D(M),E(M),S(M),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
```

```
C
      OMEGA = REAL(M)
      ITER = 1
      ITERMAX = 250
      EPS = 1.0E-10
      DLMIN = 0.1
      P = 0.01
      DO 2 I=1,M
        LAMBDAOLD(I) = LAMBDA(I)
2      CONTINUE
      ZOLD = FUNC(LAMBDA,FDEF,D,S,B,M,NXY,MNXY)
C
      DO 8 K=1,NX
        DO 6 L=1,NY
          LB(K,L) = 0.0
          DO 4 I=1,M
            LB(K,L) = LB(K,L) + LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
4           CONTINUE
6         CONTINUE
8      CONTINUE
      C = 0.0
      DO 14 I=1,M
        E(I) = 0.0
        DO 12 K=1,NX
          DO 10 L=1,NY
      E(I)=E(I)+FDEF(NY*(K-1)+L)*EXP(-LB(K,L))*B(NXY*(I-1)+NY*(K-1)+L)
10          CONTINUE
12        CONTINUE
        C = C + ((E(I)-D(I))/S(I))**2
14      CONTINUE
C
C       Use the integral equation to modify the lambdas.
C
15      TWOMU = 0.0
      DO 16 I=1,M
        TWOMU = TWOMU + (S(I)*LAMBDA(I))**2
16      CONTINUE
      TWOMU = SQRT(TWOMU/OMEGA)
C
      DO 18 I=1,M
        LAMBDA(I) = (1.0-P)*LAMBDA(I) + P*(E(I)-D(I))*TWOMU/(S(I)**2)
18      CONTINUE
      Z = FUNC(LAMBDA,FDEF,D,S,B,M,NXY,MNXY)
C
      IF (Z .GT. ZOLD) THEN
        DO 24 I=1,M
          LAMBDA(I) = LAMBDAOLD(I)
24        CONTINUE
        IF (ABS(C-OMEGA) .LT. 1.0) THEN
          RETURN
```

```
        END IF
        CALL FRPRMN(LAMBDA,M,FTOL,ITER,FRET,FDEF,D,S,B,M,NXY,MNXY)
        RETURN
      END IF
C
      IF (ITER .GT. ITERMAX) THEN
        DO 26 I=1,M
          LAMBDA(I) = LAMBDAOLD(I)
26        CONTINUE
        IF (ABS(C-OMEGA) .LT. 1.0) THEN
          RETURN
        END IF
        CALL FRPRMN(LAMBDA,M,FTOL,ITER,FRET,FDEF,D,S,B,M,NXY,MNXY)
        RETURN
      END IF
C
      IF (2.0*ABS(Z-ZOLD) .LE. FTOL*(ABS(Z)+ABS(ZOLD)+EPS)) THEN
        RETURN
      END IF
C
      DO 34 K=1,NX
        DO 32 L=1,NY
          LB(K,L) = 0.0
          DO 30 I=1,M
            LB(K,L) = LB(K,L) + LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
30          CONTINUE
32        CONTINUE
34      CONTINUE
      C = 0.0
      DO 44 I=1,M
        E(I) = 0.0
        DO 42 K=1,NX
          DO 40 L=1,NY
      E(I)=E(I)+B(NXY*(I-1)+NY*(K-1)+L)*FDEF(NY*(K-1)+L)*EXP(-LB(K,L))
40          CONTINUE
42        CONTINUE
        C = C + ((E(I)-D(I))/S(I))**2
44      CONTINUE
C
      IF (P .LT. 0.1) THEN
        DL = 0.0
        DO 46 I=1,M
          DL = DL + (LAMBDA(I)-LAMBDAOLD(I))**2
46        CONTINUE
        DL = SQRT(DL) + EPS
        IF (DL .LT. DLMIN) THEN
          P = P*(DLMIN/DL)
          IF (P .GT. 0.1) THEN
            P = 0.1
          END IF
        END IF
      END IF
C
      DO 48 I=1,M
        LAMBDAOLD(I) = LAMBDA(I)
48    CONTINUE
      ZOLD = Z
      ITER = ITER + 1
      GO TO 15
C
      END
C
C     ***************************************************************
C
      SUBROUTINE
SETHOTSPOTPARAM(RKHS,RLHS,MHS,CSQHS,M,HSMAX,XHS,YHS,
     1HSFMAP,NXY,FBKG,XHSABS,YHSABS)
C
      INTEGER M,NXY
      REAL HSMAX,FBKG
      REAL RKHS(M),RLHS(M),MHS(M),CSQHS(M),XHS(M),YHS(M),HSFMAP(NXY)
      REAL XHSABS(M),YHSABS(M)
C
      DO 10 I=1,M
        RKHS(I) = 0.0
        RLHS(I) = 0.0
        XHS(I) = 0.0
        YHS(I) = 0.0
        MHS(I) = 0.0
        CSQHS(I) = 0.0
        XHSABS(I) = 0.0
        YHSABS(I) = 0.0
10    CONTINUE
      HSMAX = 0.0
C
      DO 20 I=1,NXY
        HSFMAP(I) = FBKG
20    CONTINUE
C
      RETURN
      END
C
C     ********************************************************
C
      SUBROUTINE PEAKXELS(Z,F,PFE,X,Y)
C
C     The subroutine PEAKXELS evaluates PFE(I) = total (counts - bkg)
C     under each peak at each measurement position.
C
C     X(I), Y(I) are the x, y coordinates of the measurement in meters.
C     A is the size of a "pixel" in meters.  MHD=ISP is usually one half
C     the number of "pixels" between measurements.
C
      INTEGER M,NX,NY,NXY,I,K,L,KL,KU,LL,LU,MHD,MHDSQ,RR
      INTEGER ISP
      REAL XX,YY,Z
      REAL A
      REAL F(NXY),PFE(M)
      REAL X(M),Y(M)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      DO 20 I=1,M
        MHD = ISP
        MHDSQ = MHD**2
        XX = X(I)/A
        YY = Y(I)/A
        KL = INT(XX) - MHD
        KU = INT(XX) + MHD
        LL = INT(YY) - MHD
        LU = INT(YY) + MHD
C
        IF (KL .LT. 1) THEN
          KL = 1
        END IF
        IF (KU .GT. NX) THEN
          KU = NX
        END IF
        IF (LL .LT. 1) THEN
          LL = 1
        END IF
        IF (LU .GT. NY) THEN
          LU = NY
        END IF
C
        PFE(I) = 0.0
        DO 10 K=KL,KU
          DO 8 L=LL,LU
            RR = ((INT(XX)-K)**2)+((INT(YY)-L)**2)
            IF (RR .LE. MHDSQ) THEN
              PFE(I) = PFE(I) + F(NY*(K-1)+L) - Z
            END IF
8           CONTINUE
10        CONTINUE
20      CONTINUE
C
      RETURN
      END
C
C     ********************************************************
```

```
C
      SUBROUTINE HOTSPOTDEF(I,HSS,FBKG,D,S,B,X,Y,PFE,HSFDEF,FDEF,RKP,
     1RLP,HH)
C
      INTEGER I,HSS,M,MNXY,NXY
      REAL RKP,RLP,FBKG,HHP,HHPNN
      REAL D(M),S(M),B(MNXY),X(M),Y(M),PFE(M),HSFDEF(NXY),FDEF(NXY)
      REAL HH(M)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      DO 10 K=1,NXY
        HSFDEF(K) = FDEF(K)
10      CONTINUE
C
      CALL FINDHS(X,Y,I,HSS,PFE,RKP,RLP,FBKG)
      CALL MAGPS(D,S,RKP,RLP,I,HHP,B,X,Y,FDEF,HSS)
      CALL MAGPSNN(D,S,RKP,RLP,HHPNN,B,A,X,Y,FDEF,HSS,HH)
      IF (HHPNN .GT. HHP) THEN
        HHP = HHPNN
      END IF
      IF (HHP .LT. 1.0) THEN
        HHP = 1.0
      END IF
C
C       The ELEVATED AREA is added to the FP and to the ZKLP (which are
C       set to background activity).  These will be used in a final MaxEnt
C       deconvolution with background + ELEVATED AREA.
C
      CALL MAKEHS(RKP,RLP,FDEF,HSFDEF,HHP,X,Y,HSS)
C
      RETURN
      END
C
C       ***********************************************************
C
      SUBROUTINE
HOTSPOTPARAM(HSF,RKP,RLP,HHP,CSQ,RKHS,RLHS,MHS,CSQHS,
     1HSMAX,IHP,FBKG,HSS,I,XHS,YHS,X0,Y0,XHSABS,YHSABS)
C
      INTEGER G1,G2,RK,RL,KL,KU,LL,LU,I,IHP,HSS
      REAL RKP,RLP,HHP,HSMAX,FBKG,CSQ,X0,Y0
      REAL HSF(NXY),RKHS(M),RLHS(M),MHS(M),CSQHS(M),XHS(M),YHS(M)
      REAL XHSABS(M),YHSABS(M)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C       The final magnitude of the ELEVATED AREA is calculated.
C
      G1 = FLOOR(0.5*HSS)
      G2 = CEILING(0.5*HSS)
      RK = NINT(RKP)
      RL = NINT(RLP)
      KL = RK - G1
      KU = RK + G1
      LL = RL - G1
      LU = RL + G1
      IF (G1 .EQ. G2) THEN
        RK = FLOOR(RKP)
        RL = FLOOR(RLP)
        KL = RK - (G1-1)
        KU = RK + G1
        LL = RL - (G1-1)
        LU = RL + G1
      END IF
C
      HHP = 0.0
      DO 20 K=KL,KU
        DO 10 L=LL,LU
          HHP = HHP + HSF(NY*(K-1)+L)/(FBKG*HSS*HSS)
10        CONTINUE
20      CONTINUE
C
      RKHS(I) = RKP
      RLHS(I) = RLP
      XHS(I) = RKP*A
      YHS(I) = RLP*A
      MHS(I) = HHP
      CSQHS(I) = CSQ
      XHSABS(I) = XHS(I) + X0
      YHSABS(I) = YHS(I) + Y0
      IF (HHP .GT. HSMAX) THEN
        HSMAX = HHP
        IHP = I
      END IF
C
      RETURN
      END
C
C       ***********************************************************
C
      SUBROUTINE FINDHS(X,Y,IHP,HSS,PFE,RKP,RLP,Z)
C
C       FINDHS calculates the optimal coordinates (RKP,RLP) of the hot
C       spot.
C
      INTEGER M,NX,NY,NXY,MNXY,ISP,IHP,HSS,IHPK,IHPL,K
      INTEGER I,J,KK,LL,KKL,KKU,LLL,LLU
      REAL DSQ,R2,RR,K0,XH,YH
      REAL A,EPSSQMIN,RKP,RLP,Z
      REAL X(M),XX(M),Y(M),YY(M),PFE(M),PPFE(M)
      REAL RRIJ(M,M),RRHI(M),PHS(NX,NY),EPS(M),EPSSQ(NX,NY)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C       R2 is defined in such a way that for a square grid it includes
C       the eight closest measurement points (DSQ is the square of the
C       distance from the center of the square to any of the corners;
C       R2 is the square of [1.5*the distance to closest measurement]).
C       (For a triangular grid same but includes six closest meas.).
C
      DSQ = 2.0*(2.0*A*REAL(ISP))**2
      R2 = 1.125*DSQ
C
C       Calculate (XX,YY), the x and y coordinates for measurement points
C       with PFE>0.  Evaluate K, the number of such measurement points.
C
      K = 0
      DO 10 I=1,M
        RR = (X(IHP)-X(I))**2 + (Y(IHP)-Y(I))**2
        IF (RR .LE. R2) THEN
          IF (PFE(I) .GE. 0.0) THEN
            K = K+1
            XX(K) = X(I)
            YY(K) = Y(I)
            PPFE(K) = PFE(I)
          END IF
        END IF
10      CONTINUE
      K0 = REAL(K)
C
C       Calculate the square of the distance between detectors 100cm
C       above the ground and points on the ground directly under the
C       detectors.  Call it RRIJ(I,J).
C
      DO 14 I=1,K
        DO 12 J=1,K
          RRIJ(I,J) = (XX(I)-XX(J))**2 + (YY(I)-YY(J))**2 + 10000.0
12        CONTINUE
14      CONTINUE
C
      IHPK = NINT(X(IHP)/A)
      IHPL = NINT(Y(IHP)/A)
      KKL = IHPK - 3*ISP
      KKU = IHPK + 3*ISP
      LLL = IHPL - 3*ISP
      LLU = IHPL + 3*ISP
C
      IF (KKL .LT. 1) THEN
        KKL = 1
```

```
      END IF
      IF (KKU .GT. NX) THEN
        KKU = NX
      END IF
      IF (LLL .LT. 1) THEN
        LLL = 1
      END IF
      IF (LLU .GT. NY) THEN
        LLU = NY
      END IF
C
      DO 28 KK=KKL,KKU
        DO 26 LL=LLL,LLU
          XH = A*(REAL(KK))
          YH = A*(REAL(LL))
          RR = (X(IHP)-XH)**2 + (Y(IHP)-YH)**2
          IF (RR .LE. R2) THEN
C
C       Calculate the optimal activity PHS for a ELEVATED AREA (point source)
C       at "pixel" coordinates (KK,LL).
C
            PHS(KK,LL) = 0.0
            DO 18 I=1,K
              RRHI(I) = (XH-XX(I))**2 + (YH-YY(I))**2 + 10000.0
              DO 16 J=1,K
                PHS(KK,LL) = PHS(KK,LL)+(RRHI(I)*PPFE(J)/RRIJ(I,J))/K0
16              CONTINUE
18            CONTINUE
C
C       Calculate the additional activity EPS needed at each measurement
C       point to keep the number of counts invariant in the
C       presence of the ELEVATED AREA.
C
            DO 22 I=1,K
              EPS(I) = 0.0
              DO 20 J=1,K
                EPS(I) = EPS(I) + RRHI(I)*PPFE(I)/RRIJ(I,J)
20              CONTINUE
              EPS(I) = EPS(I) + PHS(KK,LL)
22            CONTINUE
C
C       Calculate EPSSQ(KK,LL), the sum of the square of the EPSs.
C
            EPSSQ(KK,LL) = 0.0
            DO 24 I=1,K
              EPSSQ(KK,LL) = EPSSQ(KK,LL) + (EPS(I))**2
24            CONTINUE
C
          END IF
26        CONTINUE
```

```
28   CONTINUE
C
C    Place the ELEVATED AREA at the "pixel" coordinates (KK,LL) that
C    minimize EPSSQ(KK,LL).  Set (RKP,RLP)=(XH/A,YH/A).
C
     EPSSQMIN = EPSSQ(IHPK,IHPL)
     DO 38 KK=KKL,KKU
      DO 36 LL=LLL,LLU
       XH = A*REAL(KK)
       YH = A*REAL(LL)
       RR = (X(IHP)-XH)**2 + (Y(IHP)-YH)**2
       IF (RR .LE. R2) THEN
        IF (EPSSQ(KK,LL) .LE. EPSSQMIN) THEN
         EPSSQMIN = EPSSQ(KK,LL)
         RKP = XH/A
         RLP = YH/A
        END IF
       END IF
36     CONTINUE
38   CONTINUE
C
     RETURN
     END
C
C    ***********************************************************
C
     SUBROUTINE MAGPS(D,S,RKP,RLP,IHP,HHP,B,X,Y,ZKL,HSS)
C
     INTEGER M,NX,NY,NXY,MNXY,I,K,L,KP,LP,IHP
     INTEGER ISP,HSS
     REAL DSQ,RR,R2,RKP,RLP,SUM1,SUM2,HHP,HPS
     REAL A
     REAL D(M),S(M),E(M),DE(M),FD(NXY)
     REAL B(MNXY),X(M),Y(M)
     REAL ZKL(NXY)
     COMMON/AISP/A,ISP
     COMMON/MN/M,NX,NY,MNXY,NXY
C
C    MAGPS scales the point source at (RKP,RLP) to the optimal height,
C    which is given by the multiplicative factor HPS.  It does not
C    modify the FP.
C
C    HPS is calculated by minimizing the chi square FOR THE
C    MEASUREMENTS NEAR THE ELEVATED AREA under the assumption that the
C    surface activity is given by the ZKL with a point source
C    added at (RKP,RLP).
C
C    Set the FD = ZKL
C
     DO 110 K=1,NXY
      FD(K) = ZKL(K)
110   CONTINUE
C
C    Calculate the E(I) due to the background.
C
     DO 6 I=1,M
      E(I) = 0.0
      DO 4 K=1,NX
       DO 2 L=1,NY
        E(I) = E(I) + B(NXY*(I-1)+NY*(K-1)+L)*FD(NY*(K-1)+L)
2      CONTINUE
4     CONTINUE
6    CONTINUE
C
C    Calculate the DE(I) due to activity at the point source.
C
     KP = INT(RKP)
     LP = INT(RLP)
C
     DO 10 I=1,M
      DE(I) = B(NXY*(I-1)+NY*(KP-1)+LP)*FD(NY*(KP-1)+LP)
10    CONTINUE
C
C    Calculate the optimal activity of the point source.
C
     DSQ = 2.0*(2.0*A*REAL(ISP))**2
     R2 = 1.125*DSQ
     SUM1 = 0.0
     SUM2 = 0.0
C
     DO 20 I=1,M
      RR = (X(IHP)-X(I))**2 + (Y(IHP)-Y(I))**2
      IF (RR .LE. R2) THEN
       SUM1 = SUM1 + DE(I)*(D(I)-E(I))/(S(I)**2)
       SUM2 = SUM2 + DE(I)*DE(I)/(S(I)**2)
      END IF
20    CONTINUE
C
     HPS = 1.0 + SUM1/SUM2
C
C    Add the ELEVATED AREA to the FD.
C
     FD(NY*(KP-1)+LP) = FD(NY*(KP-1)+LP) + (HHP-1.0)*FD(NY*(KP-1)+LP)
C
C    Set HPP = HPS/(area ELEVATED AREA in "pixels").
C
     HHP = HPS/(HSS**2)
C
     RETURN
     END
```

```
C
C    *********************************************************
C
      SUBROUTINE MAGPSNN(D,S,RKP,RLP,HHP,B,A,X,Y,ZKL,HSS,HH)
C
      INTEGER M,NX,NY,NXY,MNXY,I,K,L,RK,RL,NHS,HSS
      REAL RR,RKP,RLP,HHP,HPS,A
      REAL XX,YY,RRN,EBKG,DE
      REAL D(M),S(M),FD(NXY)
      REAL B(MNXY),X(M),Y(M)
      REAL ZKL(NXY),HH(M)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C    "Magnitude of point source - Nearest Neighbour":
C
C    HHP is calculated by fitting the point source's magnitude to the
C    total counts at the nearest's measurement point - under the
C    assumption that the surface activity is given by the ZKL
C    with a point source added at (RKP,RLP).
C
C    Set the FD = ZKL
C
      DO 10 K=1,NXY
        FD(K) = ZKL(K)
10    CONTINUE
C
C    Find the measurement point that is nearest to the ELEVATED AREA.
C
      XX = X(1)/A
      YY = Y(1)/A
      NHS = 1
      RRN = (XX-RKP)**2 + (YY-RLP)**2
      DO 12 I=2,M
        XX = X(I)/A
        YY = Y(I)/A
        RR = (XX-RKP)**2 + (YY-RLP)**2
        IF (RR .LT. RRN) THEN
          IF (HH(I) .GT. 0.0) THEN
            NHS = I
            RRN = RR
          END IF
        END IF
12    CONTINUE
C
C    Calculate the EBacKGground for the NHS due to the background.
C
      EBKG = 0.0
      DO 24 K=1,NX
        DO 22 L=1,NY
          EBKG = EBKG + B(NXY*(NHS-1)+NY*(K-1)+L)*FD(NY*(K-1)+L)
22        CONTINUE
24    CONTINUE
C
C    Calculate the DE due to activity at the point source location.
C
      RK = INT(RKP)
      RL = INT(RLP)
C
      DE = DE + B(NXY*(NHS-1)+NY*(RK-1)+RL)*FD(NY*(RK-1)+RL)
C
C    Calculate the (largest) optimal activity of the point source
C    assuming E(NHS) = EBKG+(HPS-1)*DE = D(NHS)+S(NHS).
C
      HPS = 1.0 + (D(NHS)+S(NHS)-EBKG)/DE
C
C    Add the point source to the FD(K,L).
C
      FD(NY*(RK-1)+RL) = FD(NY*(RK-1)+RL) + (HHP-1.0)*FD(NY*(RK-1)+RL)
C
C    Calculate HHP = HPS/(HSS**2).
C
      HHP = HPS/(HSS**2)
C
      RETURN
      END
C
C    *********************************************************
C
      SUBROUTINE MAKEHS(RKP,RLP,FDEF,FP,HHP,X,Y,HSS)
C
      INTEGER K,L,KL,KU,LL,LU,NX,NY,NXY
      INTEGER ISP,M,HSS,G1,G2
      REAL RKP,RLP,RK,RL,HHP
      REAL A
      REAL FDEF(NXY),FP(NXY)
      REAL X(M),Y(M)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C    Model the ELEVATED AREA with a (HSS**2)-"pixel" area.  MAKEHS scales
C    the ELEVATED AREA to the optimal height, which is given by the
C    multiplicative factor HHP, and calculates the final FP.
C
C    HHP was calculated in MAGPS or MAGPSNN.
C
      G1 = FLOOR(0.5*HSS)
      G2 = CEILING(0.5*HSS)
      RK = NINT(RKP)
      RL = NINT(RLP)
      KL = RK - G1
      KU = RK + G1
      LL = RL - G1
      LU = RL + G1
      IF (G1 .EQ. G2) THEN
       RK = FLOOR(RKP)
       RL = FLOOR(RLP)
       KL = RK - (G1-1)
       KU = RK + G1
       LL = RL - (G1-1)
       LU = RL + G1
      END IF
C
      DO 20 L=1,NXY
       FP(L) = FDEF(L)
20     CONTINUE

      DO 34 K=KL,KU
       DO 32 L=LL,LU
        FP(NY*(K-1)+L) = FP(NY*(K-1)+L) + (HHP-1.0)*FP(NY*(K-1)+L)
32      CONTINUE
34     CONTINUE
C
      RETURN
      END
C
C      *********************************************************
C
      SUBROUTINE MAKEHSMAP(RKP,RLP,FP,HHP,X,Y,HSS,Z)
C
      INTEGER K,L,KL,KU,LL,LU,NX,NY,NXY
      INTEGER ISP,M,HSS,G1,G2
      REAL RKP,RLP,RK,RL,HHP
      REAL A,DUMMY,Z
      REAL FP(NXY)
      REAL X(M),Y(M)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C      Model the ELEVATED AREA with a (HSS**2)-"pixel" area.  MAKEHS scales
C      the ELEVATED AREA to the optimal height, which is given by the
C      multiplicative factor HHP, and calculates the final FP.
C
C      HHP was calculated in MAGPS or MAGPSNN.
C
      G1 = FLOOR(0.5*HSS)
      G2 = CEILING(0.5*HSS)
      RK = NINT(RKP)
      RL = NINT(RLP)
      KL = RK - G1
      KU = RK + G1
      LL = RL - G1
      LU = RL + G1
      IF (G1 .EQ. G2) THEN
       RK = FLOOR(RKP)
       RL = FLOOR(RLP)
       KL = RK - (G1-1)
       KU = RK + G1
       LL = RL - (G1-1)
       LU = RL + G1
      END IF
C
      DO 34 K=KL,KU
       DO 32 L=LL,LU
        DUMMY = FP(NY*(K-1)+L)/Z
        IF (DUMMY .LT. HHP) THEN
         FP(NY*(K-1)+L) = HHP*Z
        END IF
32      CONTINUE
34     CONTINUE
C
      RETURN
      END
C
C      *********************************************************
C
      SUBROUTINE RBFINTER(MD,HH,X,Y,D,FRBFI,DAVG,FAVG,B,S)
C
C      Subroutine to calculate an initial estimate of the distribution
C      using the data, Paul Bailey's recipe, and radial basis functions
C      (RBF) interpolation using Hardy's multiquadratic method.
C
      INTEGER M,NX,NY,ISP,MD,NP,NI
      INTEGER INDX(MD)
      REAL A,R2,MINF,BFI,DNR,FOD,CSQ,CSQNEW,DUMMY,REALM,LOG2
      REAL RFCTR,RFCTR0,LMBDZ,LMBDZ0,LMBDZO50
      REAL HH(M),X(M),Y(M),D(M),E(M),XRBF(MD),YRBF(MD),BF(MD,MD),B(MNXY)
      REAL Z(MD),ZOLD(MD),FRBFI(NXY),FRBFINEW(NXY)

      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C      Given the data D (in cpm), calculate the values of activity that
C      will be used for the interpolation.
C
      NI = 1
      LOG2 = LOG(2.0)
C
      RFCTR = 0.67
      RFCTR0 = RFCTR
      LMBDZ = 0.67
      LMBDZ0 = LMBDZ
      LMBDZO50 = LMBDZ/50.0
      FOD = FAVG/DAVG
1     CALL DTOZ(X,Y,D,XRBF,YRBF,Z,MD,R2,FOD,MINF,HH,B,RFCTR,LMBDZ)
C
      DO 6 I=1,MD
       ZOLD(I) = Z(I)
6     CONTINUE
C
      DO 12 I=1,MD
       DO 10 J=1,MD
        BF(I,J) = SQRT((XRBF(I)-XRBF(J))**2+(YRBF(I)-YRBF(J))**2+R2)
10      CONTINUE
12     CONTINUE
C
C      The subroutines LUDCMP and LUBKSB solve the matrix equation
C      required for the RBF interpolation.
C
      NP = MD
      CALL LUDCMP(BF,MD,NP,INDX,DNR)
      CALL LUBKSB(BF,MD,NP,INDX,Z)
C
C      Calculate the RBF activity for all cells.
C
      DO 24 K=1,NX
       DO 22 L=1,NY
        XK = K*A
        YL = L*A
        FRBFINEW(NY*(K-1)+L) = 0.0
        DO 20 I=1,MD
         BFI = SQRT((XRBF(I)-XK)**2+(YRBF(I)-YL)**2+R2)
         FRBFINEW(NY*(K-1)+L) = FRBFINEW(NY*(K-1)+L) + Z(I)*BFI
20       CONTINUE
        IF (FRBFINEW(NY*(K-1)+L) .LT. MINF) THEN
         FRBFINEW(NY*(K-1)+L) = MINF
        END IF
22      CONTINUE
24     CONTINUE
C
C      Check the chi-square, adjust the Zs, and interpolate again if it
C      is too large
C
      CALL CHISSQ(D,E,S,FRBFINEW,B,CSQ)
C
      REALM = REAL(M)
      IF (CSQ .LT. REALM) THEN
       IF (LMBDZ .GE. LMBDZO50) THEN
        LMBDZ = LMBDZ - LMBDZO50
        RFCTR = RFCTR + 0.02*RFCTR0
        DO 25 I=1,NXY
         FRBFI(I) = FRBFINEW(I)
25       CONTINUE
        IIDUMMY = MOD(NI,5)
        IF (IIDUMMY .EQ. 0) THEN
        PRINT*,' NUMBER OF ITERATIONS / BACKGROUND CALCULATION = ',NI
         CSQPDF = CSQ/REAL(M)
C          PRINT*,' CHI SQUARE PER DEGREE OF FREEDOM = ',CSQPDF
         PRINT *,' '
        END IF
        NI = NI+1
        GOTO 1
       END IF
      END IF
C
      CALL CHISSQ(D,E,S,FRBFI,B,CSQ)
C
      IF (CSQ .LE. REALM) THEN
       PRINT*,' NUMBER OF ITERATIONS / BACKGROUND CALCULATION = ',NI
C        PRINT*,' CSQ = ',CSQ
       PRINT*,' '
       RETURN
      END IF
C
26      NI = NI+1
C
       DO 28 I=1,MD
        Z(I) = ZOLD(I)
        IF (I .LE. M) THEN
         IF (HH(I) .GT. 0.0) THEN
C           Z(I) = ZOLD(I)*SQRT(D(I)/E(I))
          Z(I) = ZOLD(I)*LOG(1.0+(D(I)/E(I)))/LOG2
          ZOLD(I) = Z(I)
         END IF
        END IF
28      CONTINUE
C
       CALL LUBKSB(BF,MD,NP,INDX,Z)
C
C       Calculate the RBF activity for all cells.
C
       DO 34 K=1,NX
        DO 32 L=1,NY
         XK = K*A
         YL = L*A
         FRBFINEW(NY*(K-1)+L) = 0.0
         DO 30 I=1,MD
          BFI = SQRT((XRBF(I)-XK)**2+(YRBF(I)-YL)**2+R2)
          FRBFINEW(NY*(K-1)+L) = FRBFINEW(NY*(K-1)+L) + Z(I)*BFI
30        CONTINUE
         IF (FRBFINEW(NY*(K-1)+L) .LT. MINF) THEN
```

```
            FRBFINEW(NY*(K-1)+L) = MINF
          END IF
32        CONTINUE
34      CONTINUE
C
      CALL CHISSQ(D,E,S,FRBFINEW,B,CSQNEW)
C
      IF (CSQNEW .LE. REALM) THEN
        DO 36 I=1,NXY
          FRBFI(I) = FRBFINEW(I)
36        CONTINUE
        RCSQNEW = CSQNEW/REALM
C          PRINT*,' (RCSQNEW .LE. 1), RCSQNEW = ', RCSQNEW
        PRINT*,' NUMBER OF ITERATIONS / BACKGROUND CALCULATION = ',NI
        PRINT*,' '
        RETURN
      END IF
C
      IF (CSQNEW .GE. CSQ) THEN
        RCSQNEW = CSQNEW/REALM
C          PRINT*,' (RCSQNEW .GE. RCSQ), RCSQNEW = ', RCSQNEW
        PRINT*,' NUMBER OF ITERATIONS / BACKGROUND CALCULATION = ',NI
        PRINT*,' '
        RETURN
      END IF
C
      DUMMY = ABS(CSQNEW-CSQ)/CSQNEW
      IF (DUMMY .LT. 0.0001) THEN
        DO 38 I=1,NXY
          FRBFI(I) = FRBFINEW(I)
38        CONTINUE
        RCSQNEW = CSQNEW/REALM
C          PRINT*,' (DUMMY .LT. 0.0001), RCSQNEW = ', RCSQNEW
        PRINT*,' NUMBER OF ITERATIONS / BACKGROUND CALCULATION = ',NI
        PRINT *,' '
        RETURN
      END IF
C
      CSQ = CSQNEW
      DO 40 I=1,NXY
        FRBFI(I) = FRBFINEW(I)
40      CONTINUE
C
      IIDUMMY = MOD(NI,5)
      IF (IIDUMMY .EQ. 0) THEN
        PRINT*,' NUMBER OF ITERATIONS / BACKGROUND CALCULATION = ',NI
        CSQPDF = CSQ/REAL(M)
C          PRINT*,' CHI SQUARE PER DEGREE OF FREEDOM = ',CSQPDF
        PRINT *,' '
      END IF
C
      IF (NI .LT. 1000) THEN
        GOTO 26
      END IF
C
      RETURN
      END
C
C      ********************************************************
C
      SUBROUTINE
DTOZ(X,Y,D,XRBF,YRBF,Z,MD,R2,FOD,MINF,HH,B,RFCTR,LMBDZ)
C
      INTEGER M,MD,NX,NY,ISP,IDUMMY
      REAL A,R2,MINF,LMBDZ,RR,RRMAX,FOD,RFCTR
      REAL X(M),Y(M),D(M),XRBF(MD),YRBF(MD),Z(MD),ZMEAS(M),HH(M),B(MNXY)

      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C      Find the minimum of the in-situ measured data points.  Multiply
C      by the ratio FAVG over DAVG to change to units of activity.
C
      MINF = D(1)
      DO 5 I=1,M
c        IF (HH(I) .GT. 0.0) THEN
          IF (D(I) .LT. MINF) THEN
            MINF = D(I)
          END IF
c        END IF
5      CONTINUE
      MINF = MINF*FOD
C
C      Fix boundry conditions: set the activity at the boundry equal to
C      the minimum activity MINF
C
      XRBF(M+1) = A
      YRBF(M+1) = A
      Z(M+1) = 0.0
      XRBF(M+2) = A
      YRBF(M+2) = NY*A
      Z(M+2) = 0.0
      XRBF(M+3) = NX*A
      YRBF(M+3) = NY*A
      Z(M+3) = 0.0
      XRBF(M+4) = NX*A
      YRBF(M+4) = A
      Z(M+4) = 0.0
C
      XRBF(M+5) = A
      YRBF(M+5) = INT(NY/4)*A
      Z(M+5) = 0.0
      XRBF(M+6) = INT(NX/4)*A
      YRBF(M+6) = NY*A
      Z(M+6) = 0.0
      XRBF(M+7) = NX*A
      YRBF(M+7) = INT(NY/4)*A
      Z(M+7) = 0.0
      XRBF(M+8) = INT(NX/4)*A
      YRBF(M+8) = A
      Z(M+8) = 0.0
C
      XRBF(M+9) = A
      YRBF(M+9) = INT(2*NY/4)*A
      Z(M+9) = 0.0
      XRBF(M+10) = INT(2*NX/4)*A
      YRBF(M+10) = NY*A
      Z(M+10) = 0.0
      XRBF(M+11) = NX*A
      YRBF(M+11) = INT(2*NY/4)*A
      Z(M+11) = 0.0
      XRBF(M+12) = INT(2*NX/4)*A
      YRBF(M+12) = A
      Z(M+12) = 0.0
C
      XRBF(M+13) = A
      YRBF(M+13) = INT(3*NY/4)*A
      Z(M+13) = 0.0
      XRBF(M+14) = INT(3*NX/4)*A
      YRBF(M+14) = NY*A
      Z(M+14) = 0.0
      XRBF(M+15) = NX*A
      YRBF(M+15) = INT(3*NY/4)*A
      Z(M+15) = 0.0
      XRBF(M+16) = INT(3*NX/4)*A
      YRBF(M+16) = A
      Z(M+16) = 0.0
C
C      Estimate activity at a point given the measured in-situ data.
C
C      First, if it's a soil sample (that is, if HH(I) equals zero) then
C      set ZMEAS equal to the data point.  If it's an in-situ
C      measurement, convert to activity and subtract MINF.
C
      DO 10 I=1,M
       ZMEAS(I) = D(I)
       IF (HH(I) .GT. 0.0) THEN
        ZMEAS(I) = D(I)*FOD
       END IF
       XRBF(I) = X(I)
       YRBF(I) = Y(I)
       IF (HH(I) .EQ. 0.0) THEN
        DO 8 K=1,NX
         DO 6 L=1,NY
          IF (B(NXY*(I-1)+NY*(K-1)+L) .EQ. 1.0) THEN
           XRBF(I) = K*A
           YRBF(I) = L*A
          END IF
6         CONTINUE
8        CONTINUE
       END IF
10     CONTINUE
C
C      If it's a soil sample, set Z equal to ZMEAS.  If it's an in-situ
C      measurement, use Paul Bailey's recipe: compare to other in-situ
C      measurements immediately surrounding it, and adjust accordingly.
C
      RRMAX = ((2.0*ISP*A)+A)**2
      DO 20 I=1,M
       Z(I) = ZMEAS(I)
       IF (HH(I) .GT. 0.0) THEN
        Z(I) = 0.0
        IDUMMY = 0
        DO 18 J=1,M
         RR = (XRBF(I)-XRBF(J))**2 + (YRBF(I)-YRBF(J))**2
         IF (RR .LE. RRMAX) THEN
          IF (HH(J) .GT. 0.0) THEN
           Z(I) = Z(I) + ZMEAS(J)
           IDUMMY = IDUMMY + 1
          END IF
         END IF
18       CONTINUE
        Z(I) = Z(I)/IDUMMY
        Z(I) = ZMEAS(I)*EXP( LMBDZ*( 1.0-Z(I)/ZMEAS(I) ) )
       END IF
20     CONTINUE
C
C      Set the parameter used in Hardy's multiquadratic method equal to
C      RFCTR*d, where d is the mean distance from each data point to its
C      nearest neighbour (this distance d is equal or approximately equal
C      to 2.0*ISP*A)
C
      R2 = (RFCTR*2.0*REAL(ISP)*A)**2
C
      RETURN
      END
C
C      *****************************************************************
C
      SUBROUTINE ludcmp(a,n,np,indx,d)
      INTEGER n,np,indx(n),NMAX
      REAL d,a(np,np),TINY
      PARAMETER (NMAX=500,TINY=1.0e-20)
      INTEGER i,imax,j,k
      REAL aamax,dum,sum,vv(NMAX)
      d=1.
      do 12 i=1,n
        aamax=0.
        do 11 j=1,n
          if (abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))
11      continue
        if (aamax.eq.0.) pause 'singular matrix in ludcmp'
        vv(i)=1./aamax
12    continue
      do 19 j=1,n
        do 14 i=1,j-1
          sum=a(i,j)
          do 13 k=1,i-1
            sum=sum-a(i,k)*a(k,j)
13        continue
          a(i,j)=sum
14      continue
        aamax=0.
        do 16 i=j,n
          sum=a(i,j)
          do 15 k=1,j-1
            sum=sum-a(i,k)*a(k,j)
15        continue
          a(i,j)=sum
          dum=vv(i)*abs(sum)
          if (dum.ge.aamax) then
            imax=i
            aamax=dum
          endif
16      continue
        if (j.ne.imax)then
          do 17 k=1,n
            dum=a(imax,k)
            a(imax,k)=a(j,k)
            a(j,k)=dum
17        continue
          d=-d
          vv(imax)=vv(j)
        endif
        indx(j)=imax
        if(a(j,j).eq.0.)a(j,j)=TINY
        if(j.ne.n)then
          dum=1./a(j,j)
          do 18 i=j+1,n
            a(i,j)=a(i,j)*dum
18        continue
        endif
19    continue
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     **************************************************************
C
      SUBROUTINE lubksb(a,n,np,indx,b)
      INTEGER n,np,indx(n)
      REAL a(np,np),b(n)
      INTEGER i,ii,j,ll
      REAL sum
      ii=0
      do 12 i=1,n
        ll=indx(i)
        sum=b(ll)
        b(ll)=b(i)
        if (ii.ne.0)then
          do 11 j=ii,i-1
            sum=sum-a(i,j)*b(j)
11        continue
        else if (sum.ne.0.) then
          ii=i
        endif
        b(i)=sum
12    continue
      do 14 i=n,1,-1
        sum=b(i)
        do 13 j=i+1,n
          sum=sum-a(i,j)*b(j)
13      continue
        b(i)=sum/a(i,i)
14    continue
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     **************************************************************
C
      SUBROUTINE NAMEOF(INPFILE,NUMMER)
C
C     Added 8/3/2000 - DETERMINE NAME OF INPUTFILE
C
C     for use with Lahey compiler
C
      CHARACTER*40 INPFILE
      CHARACTER*40 BLANKNAME
      NUMMER=7
      DO I=1,40
        INPFILE(I:I)=' '
        BLANKNAME(I:I)=' '
      END DO
      CALL GETCL(INPFILE)
C       PRINT*,'INPFILE = ', INPFILE
      IF (INPFILE.EQ.BLANKNAME) THEN
        NUMMER = 5
      END IF
      RETURN
      END
C
C     *****************************************************************
```