Table of Contents

Important Notes

The Occam 2D MT inversion code (v3.0) presented here is an implementation of the general Occam procedure of Constable, et al. (1987) extended to 2D by deGroot-Hedlin and Constable (1990). The 2D MT forward calculations are carried out with code provided by Wannamaker, et al (1987) using reciprocity to calculate the Jacobian (de Lugao and Wannamaker, 1996).

MT responses along topography

The user should take special note of how orientation is treated by the MT forward code in the presence of topography or bathymetry that is not level:

Land MT:

Electric fields are computed along the slope, magnetic fields are computed horizontally. I.e. the code assumes that the magnetic coils were leveled when they were placed, but the electric field electrodes were not.

Seafloor MT:

Electric and magnetic fields are computed along the seafloor slope, which is best for broadband marine MT instruments (e.g., Constable et al, 1998). You will need to modify the forward code if you are using long-period marine MT data collected with fluxgate magnetometers and want to model the horizontal magnetic field components in the presence of a sloping seafloor.

Compatibility with v2.0

Occam2D MT version 3.0 is fully compatible with the file structures of the previous version of Occam. The experienced user will note that though there is a new format for the startup file, the old format is still supported by the code.

Occam2D MT Command Line

The Occam2D program takes several optional command line parameters:

Occam2D [-F] [<startupfile> [<outputprefix>]]

-F = run a forward model only. The structure to be modeled is described by the startup file's log10 resistivity values.

<startupfile> = the name of the startup file. This file contains instructions to Occam on how to proceed with the inversion. If not given, the name "startup" will be assumed.

<outputprefix> = the prefix to apply to all output from Occam2D. You must specify a startupfile parameter in order to specify this parameter. Without this parameter, model iteration files will be named iterxx.iter and model response files will be named respxx.resp. The automatic log will be called logfile.logfile. If an output prefix is specified, then the names will be testxx.iter, testxx.resp, and test.logfile.

Examples:

Occam2d

Uses file "startup" in the current directory and outputs iterxx.iter, respxx.resp, and logfile.logfile.

Occam2d iter17.iter

Uses file "iter17.iter" in the current directory (presumably from a previous run of occam2d) and outputs as above.

Occam2d BlockyZone.iter Blocky

Uses file "BlockyZone.iter" as the startup file. Outputs Blockyxx.iter, Blockyxx.resp, and Blocky.logfile.

You can optionally redirect the output from Occam2d to a text file in the normal way (i.e. using the greater than symbol followed by a file name: occam2d > catchthis.log)

Occam2D MT Input Files

Required Files

The following files are required for running Occam2D, both in inversion and forward-only mode. Each file is described in more detail in further sections.

Startup file – this file describes the options that configure the inversion, and contains the starting model parameter values. It also contains the file names of the data file and the model description file.

Data file – this file describes the instrument site locations and data. For inversion, this is data that you have already collected. For forward-only mode, this is the data that you want the process to output.

Model file – this file describes the configuration of the finite element model space. This file also contains the name of the mesh file.

Mesh file – this file describes the mesh used for the forward calculation step in the inversion. Occam2D MT uses a two-mesh approach: the mesh is a finely divided space for calculation of the forward model (in both forward-only and in inversion modes). It can contain tens of thousands of bricks. The model is a coarser mesh which aggregates lots of mesh blocks for the matrix inversion. The matrix inversion is expensive in both time and memory, so the model space can contain at most about 7000 blocks under the current OS-enforced memory limits of 2Gb. Using a finer mesh than model allows the forward calculation to minimize edge affects caused by large blocks. Using a coarser model allows Occam2D to run in a reasonable amount of time and still fit within the memory limit of the compiler.

A standard set of files might be like this:

Startup.iter	the startup iteration file
Occam.model	the model description
Occam.mesh	the forward mesh description
Mysurvey.dat	the data to invert (or calculate in forward-only mode)

Optional Files

The following files are optional. They not required by Occam 2D MT and are only looked for if they are referred to by the header of the model file.

Statics File – this file contains entries allowing the inversion to account for the effects of static shift at individual sites. Values in this file can either force an amount of static shift or allow the inversion to invert for it.

Prejudice File – this file lists special model prejudices for the model parameters and any static shift parameters.

Startup File

The startup file is composed of a header of properties followed by a list of model parameter values. The iteration files generated by Occam 2D are themselves startup files with updated model parameter values. Below is a typical header, followed by explanations of the various options.

Each line in the header consists of an option keyword or phrase, a colon, and appropriate values. Extra spaces before and after the colon are ignored. Capitalization is unimportant in the keywords. Capitalization of file names, however, may be important on UNIX systems (typically Linux, but not OS/X).

Comments are denoted by either ! or % and may begin anywhere in the header.

The order of the items in the header is unimportant, **except** that the last item **must be** "Param Count:". Occam2D assumes that the model parameter values immediately follow this line.

```
Format:              OCCAMITER_FLEX
Description:         Some description
Model File:          occam.model
Data File:           datafilename.dat
Date/Time:           Wed Dec 20 11:03:17 2006
Iterations to run:   20
Target Misfit:       1.3
Roughness Type:      1
Diagonal Penalties:  0
```

Normally, Occam only calculates roughness penalties horizontally & vertically. Set this option to 1 to add diagonal penalties to the horizontal & vertical.

```
Stepsize Cut Count:  8
```

When Occam is searching model space for the best fitting or smoothest model, it may occasionally run into a case where the current iteration does not fit as well as the previous. In that case, it will cut the size of the steps in the roughness parameter and conduct a finer search. This searching process may occur many times in the same iteration. The parameter above allows you to limit the number of times Occam cuts the step size in a search for a better fitting model. If this count is exceeded, Occam will end the iteration prematurely on the grounds that the best fitting model may have already been found, even though the Target Misfit has not been reached. In any event, human intervention is required at this point so Occam ends.

```
!Model Limits:       min,max
```

This option allows you to impose limits on the values that the model parameters may take. For Occam2D MT, the model parameters are in log10 resistivity, so this min & max are also log10 resistivity. This option is not normally used and so is commented out with the "!" character. To use the option, remove the exclamation point and specify a minimum and maximum value.

Note that there may be cases where Occam2D needs to step into unrealistic model parameter values in order to get around a local minimum. The use of this option will not allow Occam2D to take that step and may cause your inversion to stop at a higher RMS.

```
!Model Value Steps:  stepsize
```

This option discretizes model parameter space into steps "stepsize" large. For Occam2D MT, the model space is in log10 resistivity, so these steps are in the exponent (e.g. for stepsize=0.1, values allowed will be $10^{0.1}$, $10^{0.2}$, $10^{0.3}$, $10^{0.4}$, etc…, subject to whatever "Model Limits" you have imposed.) This option is not normally used and so is commented out

with the "!" character. To use the option, remove the exclamation point and specify a stepsize.

Tests have shown that this option tends to sharpen boundaries in the inversion without changing the basic outcome. So, running an inversion without this option then discretizing the final iteration yields roughly the same result as running the inversion with this option. Some smaller features (e.g. those caused by noise in the data) may disappear from the inversion when using this option.

`Debug Level:` `1` The debug level value may be 0, 1, or 2.
0 minimizes the amount of "chatter" that Occam2D displays while running.
1 (the default), displays progress on a frequency-by-frequency basis.
2 displays more diagnostic info and **changes the way Occam2D searches model space!** Normally, Occam2D uses a "golden section search" to find the minimum RMS during an iteration. Using debug level 2 causes Occam2D to use a brute-force scan through the roughness parameter. This is much more time consuming, but useful if your inversion has reached an area of the model space containing local minima.

`Iteration:` `0` This option tells Occam2D the iteration number represented by the model values **in this file**. The next iteration to be output is this value +1. For example, if you look in the seventeenth iteration file of an inversion, this number will be 17. If you restart the inversion using that iteration file as the startup file, Occam2D will recognize that it has read in iteration 17 and start output at 18.

`Lagrange Value:` `5.000000`
`Roughness Value:` `0.1000000E+11`
`Misfit Value:` `1000.000`
`Misfit Reached:` `0` This flag is either 0 to indicate that the target RMS misfit has not yet been reached, or 1 to indicate that it has. It is most useful when restarting an inversion. For example, you run an inversion with an initial target of 1.0 but find after 10 iterations that it appears to have settled at 1.23. So, you can modify the iteration 10 file, set the target misfit to 1.23 and set this flag to 1 so that Occam will realize that it has converged to the target misfit and now needs to make the model as smooth as possible while maintaining this misfit.

`Param Count:` `1421` This count must be the last item in the header. Occam2D assumes that the model parameter values immediately follow this line.

Data File

Unlike the startup file, the Data file format is picky. Items must be listed in the order below, and spacing after the colon is very important. Occam2D MT assumes that the values begin at or after character 18.

Example:

```
FORMAT:            OCCAM2MTDATA_1.0
TITLE:             Forward model response from Occam2D
SITES:             3
M10                List of site names.
C00
P10
OFFSETS (M):
  -10000.0         List of site locations along the 2D line in meters. It is not
   0.0             important that these be physical locations like UTM. It is
   10000.0         important that the spacing between sites be accurate.
FREQUENCIES:       3
  1.000000         List of frequencies in decreasing order.
  0.100000
  0.010000
DATA BLOCKS:       47
SITE FREQ TYPE DATUM ERROR
  1    1    1   1.9960  0.0217
  1    1    2  44.6266  1.4500
  1    1    5   1.9975  0.0217
  1    1    6  45.3798  1.4500
  ...
```

The data columns are as follows:

Site - the number of the site from the site list that this data belongs to.
Freq - the number of the frequency from the frequency list.
Type - The type of data:

Apparent Resistivity & Tipper data types:

1 = TE apparent resistivity (log10)
2 = TE phase
3 = real(tipper) Hz/Hy
4 = imag(tipper) Hz/Hy
5 = TM apparent resistivity (log10)
6 = TM phase
7 = reserved for TM Ez/Ey tipper
8 = reserved for TM Ez/Ey tipper
9 = TE apparent resistivity (linear) NEW
10 = TM apparent resistivity (linear) NEW

Impedance data types:

11 = real(Zxx) ! not used until we incorporate site-rotation angle

12 = imag(Zxx) ! not used until we incorporate site-rotation angle
13 = real(Zxy) ! TE
14 = imag(Zxy) ! TE
15 = real(Zyx) ! TM
16 = imag(Zyx) ! TM
17 = imag(Zyy) ! not used until we incorporate site-rotation angle
18 = imag(Zyy) ! not used until we incorporate site-rotation angle

Datum - the data value

Error - the size of the error for this measurement.

For log10 resistivity this value can look a little strange. It is derived from the calculation d(log_10(x))/dx = 1/[x ln(10)]. So for 10% error, dx = 0.10x thus d(log_10(x)) = 0.10x / x ln(10) = .1/ln(10) = 0.0434.

For phase, start in the complex plane. You have a vector representing one of the complex components of Z. (Recall that phase = arctan(imag(Zxy) / real(Zxy)).) Most time series analysis will give equal errors in Re and Im. So, for small errors, the error describes a circle around the tip of the vector. So, a RELATIVE amplitude error of Eamp (along the direction of the vector) corresponds to a phase error of Epz in radians. So, an amplitude error of 10% equals a phase error or 0.1 x 57 degrees, or 5.7 degrees. BUT, apparent resistivity goes as amplitude squared, so for small errors the linearization halves the phase error to 2.85 degrees for an apparent resistivity error of 10%.

Model File

Like the data file, the model file is picky about spacing. Values are expected to begin on or after the 18th character position and the order of the items is important.

```
FORMAT:             OCCAM2MTMOD_1.0
MODEL NAME:         Descriptive text here
DESCRIPTION:        More descriptive text
MESH FILE:          Occam.mesh
```
The name of the mesh file.
```
MESH TYPE:          PW2D
STATICS FILE:       none
```
This is the file name of the optional statics file (see following section). This file is not required by Occam 2D MT and is usually not provided. The value "`none`" indicates no file is present.
```
PREJUDICE FILE:     none
```
This is the file name of the optional prejudice file (see following section). This file is not required by Occam 2D MT and is usually not provided. The value "`none`" indicates no file is present.
```
BINDING OFFSET:     -63640.0
```
This parameter is the source of much confusion and anguish when creating models by hand. It is the location along the 2D profile of the **right**-hand side of the **left**-most model block. Note that this is the **model** block and not the **mesh** block. A model block can be / is usually an aggregate of many mesh blocks (see the required file section in this document).
```
NUM LAYERS:         31
```
This is the number of layers of model blocks defined in this file. This line must be immediately followed by a description of those model block layers. Each layer is specified by two lines as described below.
```
2 64
```
This is the first in the pair of lines that describes a layer of model blocks. The first number is the number of mesh layers that are aggregated in this model layer. The second number is the number of model blocks in this layer – as described by the next line.
```
5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  2 2 2 2 2 2 2 2 2 2 2 5
```
This is the 2nd in the pair of lines that describes a layer of model blocks. Each number here describes the number of mesh blocks that are aggregated in each model block. The sum of this line must equal the sum of columns in the mesh.
```
. . .
1 33
5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
  4 4 4 4 4 5
```
It is usual in MT to make the model bricks larger as the layers go deeper in the model. **WARNING:** The Occam2D MT code requires that the model block boundaries line up with the boundaries of the blocks above. If this is not the case, the program will still run, but roughness penalties will not be calculated properly for the lower blocks.
```
. . .
1 3
5 124 5
```
Your model should narrow down to a half-space or nearly one at the bottom.
```
NUMBER EXCEPTIONS -73
223 288 0
224 288 0
```

Your model can specify changes in the roughness penalty calculations using the “Number Exceptions” section of the model file. This allows you to specify blocks between which there should be either no penalty or modified penalty.

There are two ways to do this, the easiest of which is shown here. If you specify a negative number of exceptions, then Occam2D MT expects the line to be followed by three columns of numbers: block #1, block #2, weight on the exception between them. Use the MatLab routine plot2DModel to show the block numbers in your model.

If you specify a positive number of exceptions, then Occam2D MT expects the line to be followed by 5 columns of numbers: layer #1, column #1, layer #2, column #2, and weight. Layer and column indicate a block.

In either case, by specifying a zero weight, you remove the penalty between the blocks. Weight less than 1 decreases the penalty between blocks, while weight greater than 1 increases the penalty. Also, you can create new relationships between two blocks (for example, the left & right edges of the model) by specifying block numbers and a weight of 1.0.

Mesh File

Like the data file, the mesh file is picky about the order of items it contains. However, unlike the other files, it does not contain illuminating comments or headers. There are two main sections to the mesh file: specification of the mesh, identification of free parameters in the mesh.

```
Descriptive text on first line.
0 135 52 0 0 2  These first six numbers specify the characteristics of the mesh.
```

1st – always zero
2nd – # of horizontal "nodes" (i.e. blocks + 1)
3rd – # of vertical "nodes" (i.e. layers + 1)
4th – # of fixed resistivities in the mesh, not including seawater – usually zero, but can be used to force particular regions of the mesh to have specified resistivities. There can be up to 25 fixed resistivities. More on these below.
5th – always zero
6th – always two

```
<List of fixed resistivity values>
```

If parameter 4 is not zero above, then the resistivity values for these fixed zones should be listed here. Do **not** list them as log10, but as actual resistivities. These values will be referenced in the mesh parameter list below by capital letters A through Y. Z is reserved for seawater.

```
<List of column widths in meters>  Example:
115440.0 53580.0
24870.0 11540.0 5360.0 2490.0 1150.0
 1000.0  1000.0 1000.0 1000.0 1000.0
 1000.0  1000.0 1000.0 1000.0 1000.0
. . .
1000.0  1000.0 1000.0 1000.0 1000.0
 1150.0  2490.0 5360.0 11540.0 24870.0
53580.0 115440.0
```

In the example above, notice how the first few and last few columns are quite wide compared to the rest. This is a requirement of the Wannamaker 2D forward code used in Occam 2D MT. This code calculates a 1D solution at the edge of the mesh and uses it as a boundary condition. The 2D fields inside the mesh are calculated as perturbations that must resolve down to the 1D solution at the edge of the mesh. If the mesh is not wide enough, then the variability in the 2D solutions will suppressed.

```
<List of layer heights in meters>  Example:
1000.0  1000.0 1000.0 1000.0 1000.0
. . .
2000.0 3000.0 4000.0 5000.0 6000.0
7000.0 8000.0 14000.0 30000.0 60000.0
```

In the example above, note that the layers get thicker with depth. This is to be expected in a diffusive situation like MT where resolution broadens with depth.

```
0
```

There should always be a zero before the next section.

```
ZZZZZZZZZZZZZ...
```

This is the beginning of parameter specs. See below.

```
ZZZZZZZZZZZZZ...
ZZZZZZZZZZZZZ...
```

```
ZZZZZZZZZZZZZ...
?????????????...
?????????????...
?????????????...
?????????????...
```

Parameter specification is, well, tedious. There must be four lines for each layer in the mesh. The four lines contain symbols which represent the value of four triangles in each mesh block. Imagine that each mesh block is cut by two diagonal lines, making four triangles. These four lines stand for the top, left, bottom, and right-hand triangles of each block in a layer. The character indicates what resistivity value is to be assigned to the triangle. '?' means that the triangle is a free parameter to be calculated. 'Z' means that the triangle contains seawater and '0' (zero) means that it contains air. Other letters 'A' through 'Y' indicate fixed resistivity values as specified above.

Note that the division of blocks into triangles allows you to create a mesh that closely follows topography or bathymetry.

Statics File

The statics file is **optional**. Occam2D MT will only look for this file if it is named in the model file's header. Like the data file, the statics file is picky about spacing. Values are expected to begin on or after the 18th character position and the order of the items is important.

```
FORMAT:            OCCAM2MTSHIFT_1.0
DESCRIPTION:       STATICS
DATA FILE:         DATA.DAT
```

The first three lines of this file are not used, but must be present. The data file name is taken from the startup file, not the statics file. Its listing here is left-over from a previous version and is now ignored. Occam 2D MT v3.0 reads and discards these lines.

```
CONSTRAINT TYPE:   0
```

There are three types of summing constraints available in static processing. 0 = no summing constraint. 1 = add a constraint involving the TE + TM sum. 2 = add separate constraints for TE and TM sums.

```
CONSTRAINT ERROR: 0.1
```

This is the error value to use for the sums if the constraint type = 1 or 2.

```
NO. SHIFT BLOCKS: 12
```

This is the number of rows of data below.

```
SITE    DATA TYPE    SHIFT    INVERT/NOT
1           1           0.0         1
1           5           0.0         1
2           1           0.0         1
...
```

The 4 columns above identify the shift data. Site numbers and data type are as in the data file. The INVERT/NOT flag indicates whether to invert for this parameter (flag = 1) or to accept the value in the SHIFT column (flag = 0). Static shift data is output in the iteration files following the spatial model parameters.

Prejudice File

The prejudice file is **optional**. Occam2D MT will only look for this file if it is named in the model file's header. There are currently two supported formats for the optional prejudice file. Both of these formats are picky about spacing. Values are expected to begin on or after the 18th character position and the order of the items is important.

Format 1.0 – In this format, prejudices and weights are supplied for all inversion parameters. Any parameters which you don't want prejudiced must have prejudice weights of zero.

```
FORMAT:            OCCAM2MTPREJ_1.0  The descriptor on this line
      indicates which format the file is in.
DESCRIPTION:       Blah blah blah    Description is ignored.
NO. PARMS:         212               This is the number of
      parameters contained in the prejudice file. This number must be equal to the
      number of model parameters (see "Param Count" in the startup file) + the
      number of static shift parameters (see "No. Shift Blocks" in the statics file).
      Occam then expects to read 2x this number of values below, in this example, 424
      values in all. The first 212 values are the actual prejudice values for the model
      parameters and static shift parameters (if any). The last 212 values are the
      weights to apply to the prejudice values.
2.5  2.5  2.5  2.5  2.5
. . .
1 1 1 1 0 0 1 1 0 0 0 0
. . .
```

Format 2.0 - In this format, prejudices and weights are supplied for only particular model parameters.

```
FORMAT:            OCCAM2MTPREJ_2.0  The descriptor on this line
      indicates which format the file is in.
NO. PARMS:         26                This is the number of rows
      included below. Each row must contain three values which are interpreted as the
      parameter number, prejudice value, and prejudice weight.
15 2.5 1.0      The parameter number is the 1-based number of the model
16 2.5 1.0      parameter or static shift parameters. Static shift parameter
27 2.0 0.7      numbers begin after the model parameter numbers.
. . .
```

Output Files

Occam 2D produces the following output files. Note that the actual names may differ from those given below if you specify a new output prefix in the command line parameters. (See "Occam2D MT Command Line" for details.)

Logfile.logfile - This is simply a text log of the steps taken during each iteration of the inversion. The value of the misfit and lagrange multiplier at each step are given.

ITERxx.iter - One of these files is output at the end of each successful iteration. It is a copy of the startup file with the model parameters set to the values from the end of iteration "xx". Note that if the optional statics file is included in the run, then there will be additional values following the spatial model parameters; one for each statics row in the statics file.

RESPxx.resp - This file is paired with the iteration file above. One is output at the end of each successful iteration. The file is a table containing columns for the data parameters (receiver number, frequency number, data type, strike angle), the input data value, the data value produced by the forward code using the model from ITERxx.iter, and the normalized difference: (input data – forward data) / error. The "error" value is from the input data file.

References

Occam's Inversion:

Constable, S. C., R. L. Parker, and C. G. Constable, Occam's inversion - A practical algorithm for generating smooth models from electromagnetic sounding data, Geophysics, 52 (03), 289–300, 1987.

deGroot-Hedlin, C., and S. Constable, Occam's inversion to generate smooth two-dimensional models from magnetotelluric data, Geophysics, 55 (12), 1613–1624, 1990.

2DMT Forward code:

Wannamaker, P. E., J. A. Stodt, and L. Rijo, A stable finite-element solution for two-dimensional magnetotelluric modeling, Geophysical Journal of the Royal Astronomical Society, 88, 277–296, 1987.

2DMT Jacobian sensitivity code:

de Lugao, P. P., and P. Wannamaker, Calculating the two-dimensional magnetotelluric Jacobian in finite elements using reciprocity, Geophys. J. Int., 127, 806-810, 1996.