WSINV3DMT version 1.0.0 for Single Processor Machine

User Manual

Weerachai Siripunvaraporn
Department of Physics, Faculty of Science
Mahidol University
Rama 6 Rd., Rachatawee, Bangkok 10400
THAILAND
wsiripun@yahoo.com

Gary Egbert
College of Oceanic & Atmospheric Science
Oregon State University
USA

Makoto Uyeshima
Earthquake Research Institute
University of Tokyo
JAPAN

Copyright © 2006 Weerachai Siripunvaraporn. All rights reserved.

*This work has been supported by Thailand Research Fund (PDF/37/2543; RSA4780021) to WS, and partly by a JSPS postdoctoral fellowships (P02053) and MEXT, Japan (02053) to WS and MU, and by Department of Energy (DOE-FG0302ER15318) to GE.

1. INTRODUCTION

1.1 What are WSINV3DMT ?

WSINV3DMT (Siripunvaraporn et al., 2005) is a full 3-D inversion program for Magnetotelluric data. It is extended and implemented from the 2-D data space Occam's inversion (Siripunvaraporn and Egbert, 2000). The inversion seeks the smoothest minimum structure model subject to an appropriate fit to the data.

The original Occam's inversion was introduced by Constable et al. (1987) for 1-D MT data. It was later expanded to 2-D MT data by deGroot-Hedlin and Constable (1990). Occam's inversion is stable and converges to the desired misfit in relatively small number of iterations compared to most other methods. They both are based on the model space method. Computational costs associated with construction and inversion of model-space matrices make a model-space Occam approach to 3D MT inversion impractical because all computations depend on the size of model parameter, M.

These difficulties can be overcome with a data-space approach, where matrix dimensions depend on the size of the data set N, rather than the number of model parameters M. Generally, $N << M$ for MT data. As discussed in Siripunvaraporn and Egbert (2000), the transformation of the inverse problem to the data space can significantly improve the computational efficiency for the 2-D MT problem. The WSINV3DMT inversion code is based on the data space approach (Siripunvaraporn et al., 2005). With the transformation to data space the computational costs (i.e. CPU times and RAM required) are significantly reduced making the 3-D inversion practical for PCs and workstations.

1.2 What can WSINV3DMT do?

- Invert full 3-D impedance data set with 3-D inversion (see Reference #1).
- Invert single 2-D profile with 3-D inversion (see Reference # 2).
- Data suited for current version are impedance tensor, Z_{xx}, Z_{xy}, Z_{yx} and Z_{yy} .

1.3 Philosophy of WSINV3DMT Inversion

WSINV3DMT Inversion program is just **a mathematical tool** that gives you a model that possibly fit your data. It is not as intelligent as you are. It does not know anything about geology of your area. If you expect to get a "reasonable" model, you must do some work on your data and understand how the inversion works. Don't expect to just plug the data into the inversion, sit back, relax and Yureka! You may have to adjust your model and data parameters several times before you obtain a reasonable model.

1.4 How to efficiently use WSINV3DMT?

Since WSINV3DMT is based on the data space method – all computations depend on the size of data (N), not the size of model parameter (M). There may be some cases where M could be larger than N. In such cases,

- we recommend choosing a "subset" of data to inversion. Because the MT data is redundant, the "pre-selection" process can be performed by selecting fewer stations uniformly covering the survey region.
- Try to exclude sites that are contaminated by noise.
- Another procedure is to use fewer periods, e.g. 2-3 periods per decade. More than 70% of CPU time of the inversion is mainly dominated by the computational time used in forward modeling. Using more periods will require more forward model runs and will further slow down the inversion.

1.5 Limitations of WSINV3DMT?

- Long runtimes may be required due to the many forward modeling runs (depending on your computer systems)
- Large memory is still required to store the sensitivities.
- If you would like to run with a bigger data set, or with a model that cannot be fit on your computer system, see "collaboration" section on the website.

1.6 Announcement and Software Update

Software update and other news will only be announced on the website at http://mucc.mahidol.ac.th/~scwsp/wsinv3dmt/index.htm; not in the MTNet mailing list.

1.7 Collaborations

Collaborations between the developers and other researchers are encouraged, including

- If you have "new" idea for inversion but cannot do it by yourself or by your students.
- You cannot run the code on your computer system due to limited memory.
- Exchange students/researchers/postdoc including co-advisor.

Send me email to discuss these or other possible collaborations.

1.8 Before using the code

Please read the conditions of use and agreements for code modifications on the next pages and on the website for any update agreements.

Conditions of Use

1. I will use WSINV3DMT & WSFWD3DMT (hereinafter referred to as "the codes") for academic and non-profit purpose only. I will not accept any payment for use of WSINV3DMT & WSFWD3DMT.
2. I will not redistribute WSINV3DMT & WSFWD3DMT directly to any other person, including colleagues, professors or students. Any third party wishing to use the code should be referred to this website.
3. I will not modify the code without permission of the code developer. See conditions for code modifications.
4. I will allow my name to be included in the "Users and Publications" in the website. I will send lists of publications resulting from use of WSINV3DMT & WSFWD3DMT in the format given in "Users and Publications", and I will also send a copy of any manuscript (a printed hardcopy, or better, electronic pdf) to the code developer.
5. In any publications resulting from use of this code, I will cite the code appropriately. Suggested citations at this time are :
   - inversion of a full 3-D data set (more than one profile simultaneously): Siripunvaraporn et al. (2005a) [an extension of 2-D code of Siripunvaraporn and Egbert (2000)]
   - single profile or transect: Siripunvaraporn et al. (2005a), Siripunvaraporn et al. (2005b).
   - forward modeling code WSFWD3DMT : Siripunvaraporn et al. (2002).

   - Siripunvaraporn W., G. Egbert, Y. Lenbury and M. Uyeshima, 2005a, Three-Dimensional Magnetotelluric: Data Space Method, *Physics of the Earth and Planetary Interiors*, 150, 3-14.
   - Siripunvaraporn W., G. Egbert and M. Uyeshima, 2005b, Interpretation of 2-D Magnetotelluric Profile Data with 3-D Inversion: Synthetic Examples, *Geophys. Jour. Inter.* , 160, 804-814.
   - Siripunvaraporn W., G. Egbert and Y. Lenbury, 2002, Numerical Accuracy of Magnetotelluric Modeling: A Comparison of Finite Difference Approximation, *Earth Planets Space*, 54, 721-725.
   - Siripunvaraporn W. and G. Egbert, 2000, An efficient data-subspace inversion method for 2-D magnetotelluric data, Geophysics, 65, 791-803.

6. I will report bugs to the developer. Note however that bug fixes (especially timing of bug fixeds!) are at the discretion of the developer.
7. If I move to another institution, change affiliations or email addresses, and wish to continue using the codes, I will report my change of address to the developer.
8. Violation of any of the conditions of use here will result in suspension of the right to use the codes for any future purposes. I will delete the code immediately if requested by the developer.
9. The developer is not responsible for any damages to the user caused by usage of this code.

Conditions for Code Modification

Permission is required for any modifications either minor or major.

- Minor modifications include changes to the code required for a different computer system, minor bug fixes, etc.
- Major modifications include modifying the code to invert other responses, changes to boundary conditions, search algorithms, to use with more processors simultaneously, etc.

Before making changes, send a request form (provided in the website) to wsiripun@yahoo.com to describe the desired changes and seek permission. Usually permission for minor changes will be given. For more major modifications permission will be considered on a case by case basis. Once modifications are completed and tested, any new or modified routines, with documentation of all changes, made must be sent to the developer. Once the permission is given for specific changes, only these changes are to be made. New permission must be sought to change the scope of code modifications.

Conditions for Minor Code Modification

1. After my modification to fit with the computer system of my own institutes or debugging the code, I must send the whole codes back to the developer. I am allowed to distribute **only the modified subroutines** to others who already have permission from the developer for code use, but **not whole codes**.
2. I will not modify the code beyond my specific request. If I want to make further modifications, I must make a new request.

Conditions for Major Code Modification

1. I agree with the conditions for minor code modifications above.
2. If the modification itself is the subject of a publication, I will offer the developer the option of co-authorship. The new publication must cite the original publications of WSINV3DMT & WSFWD3DMT as in the conditions of use depending on which part of the code I modify. See list of publications for conditions of use.
3. Once finished, **I will send the complete modified code, with documentation of all modifications made, to the developer**. **I can ONLY send the modified codes or subroutines to others**. I will not distribute the whole package without explicit permission by the original developer. Before granting permission, any 3rd party users will be required to register with this website first.
4. Publications resulting from use of the modified code must cite the original papers describing WSINV3DMT & WSFWD3DMT, not just more recent publications describing the modifications.
5. The original developer has the right to modify or add to the code in any way.

2. SYSTEM CONFIGURATION AND CODE COMPILATION

2.1 Know your RAM size

Before using WSINV3DMT program, you should know your computer specification. The most important factor is the size of RAM, e.g., 256 MByte, 512 MByte or 1 GByte. Currently, new PC can have a memory up to 4 GByte with 1 GByte RAM for each memory slot, and new workstation can have a memory even upto 16 (or 32) GByte of RAM. **Size of RAM limits the size of data and model parameters for running WSINV3DMT.**

2.2 Know the sizes of data and model parameters

- The data parameter size N is the total size of your dataset, which generally the product of the number of periods N_p, the number of stations N_s, and the number of responses N_r (maximum is 8 for all impedances, and 4 for only off-diagonal impedances),

  i.e. , $$N = N_p N_s N_r .$$

- The model parameter size M is the total number of discretization blocks, which is a product of number of discretization in x (north-south) M_x, number of discretization in y (east-west) M_y, and number of discretization in z (vertical) M_z,

  i.e., $$M = M_x M_y M_z .$$

At present WSINV3DMT requires the same grid be used for the model parameterization and numerical model solutions.

2.3 Limitations of WSINV3DMT *must read

WSINV3DMT program requires RAM storage at least 1.2 × (8N^2 + 8NM) to store the N × N representer matrix and N × M sensitivity matrix. The program requires extra memory for other computations. Total storage is dominated by these two matrices. The factor 1.2 which accounts for this is not exact, but is generally an average approximation. This factor ranges between 1.-1.4 (or larger) depending on the size N and M.

For example, if you have 50 sites with 16 periods and 8 responses, i.e., N = 50*16*8 = 6400, and you discretize the model into 35 × 35 × 30, i.e. M = 36750, the estimated needed memory is 1.2 * (8*6400*6400 + 8*6400*36750) = 2.6 GByte.

Before you compile the program, make sure that your computer's RAM size is larger than this minimum requirement. If not, there are several options.

- Adding more physical RAM to your computer system.
- Reduce N by using the fact that the MT data is redundant.
  ***TIP-1**: Running with at least two-three periods per decade may be enough to construct a reasonable model.
  ***TIP-2**: Select a subset of stations that uniformly cover the whole region.
  ***TIP-3**: Get rid of contaminated data sites and/or frequencies.
- Reduce M.
  ***TIP-4**: The grid discretization in the area outside of interest can be crude, while inside is finer.

2.4 Before compilation *must read

You must edit "para.h" file to suit with your computer system. Using an editor (such as vi, emac, notepad, etc.), you edit or adjust the numbers following these variables ONLY. Do not change any other parameters.

Variable name :		Explanation
NSMAX	:	Maximum number of station
NPMAX	:	Maximum number of period (frequency)
NRMAX	:	Maximum number of response
	:	(8 for all impedance tensor, real and imaginary of Z_{xx}, Z_{xy}, Z_{yx} and Z_{yy}, and 4 for real and imaginary of Z_{xy} and Z_{yx})
XMAX	:	Maximum number of horizontal block in x-comp. (north-south)
YMAX	:	Maximum number of horizontal block in y-comp. (east-west)
ZMAX	:	Maximum number of vertical block in z-comp. (down-up)

Next, edit “Makefile” to suit with your computer system.

2.5 Compilation

For UNIX system

After adjusting any parameters, to compile the program, type “make” at the command line. If your setting fits the system, you would obtain “wsinv3dmt” executable file. Otherwise, you may obtain an error message, like this,

Array “sens” at <^> is too large to handle.

The program has been successfully test on Sun workstation and cygwin system.

For Other systems Please consult your compiler manual.

2.6 Testing your compilation

To test whether “wsinv3dmt” is working on your machine, typing “wsinv3dmt” at the command line. The program should ask for the startup file. If not, you may receive the error message. **Tip:** Check there may be other people running under your system and consuming your memory (depending on the computer system!)

Goto directory /TEST3DMT/RUNTEST, there is a “startup” file. Run “../../wsinv3dmt” with the default “startup” file. The data and model files are located in /TEST3DMT main directory. After the run the output will be generated in this directory. The old run from my system is saved under directory /OLDRUNTEST in this directory. You can compare the result of your run with the results from the old tests. If they are different, send me email to wsiripun@yahoo.com

NOTE: this test (and actual inversion) may be slow for some systems. For example, on a 1 Giga Hertz machine, it may take at least a day to finish 5 iterations on the test example case.

3. INPUTS

There are several input files that need to be prepared before running WSINV3DMT.

3.1 Startup file

The Startup file lists all of the input files and several parameters used by the program.

Example of Startup file

```
DATA_FILE               ../blk3d.data
OUTPUT_FILE             blk3d
INITIAL_MODEL_FILE      ../init3d
PRIOR_MODEL_FILE        default
CONTROL_MODEL_INDEX     default
TARGET_RMS              1.
MAX_NO_ITERATION        5
MODEL_LENGTH_SCALE      default
LAGRANGE_INFO           default
ERROR_TOL_LEVEL         default
```

Description of startup file

There are 10 lines and 10 keywords. All keywords must be in this order. The easiest way is to copy the "startup_file" from the example, and change the input parameters after the keywords.

Line 1: DATA_FILE *filename* (less than 70 characters)
DATA_FILE requires filename (including path if appropriate) of the data file. A full description of the data file is given in section 3.2.

Line 2: OUTPUT_FILE *filename*
OUTPUT_FILE requires filename used to generate output filenames. More detail is given in output section (section 4.)

Line 3: INITIAL_MODEL_FILE *filename* (less than 70 characters)

INITIAL_MODEL_FILE requires filename (including its path) of the initial model. A full description of the initial model is given in section 3.3.

Line 4: PRIOR_MODEL_FILE *filename*/default

PRIOR_MODEL_FILE requires filename (including its path) of the prior or base model. This is the $\mathbf{m_0}$ term in equation (1). If setting as "default", the prior model is the same as the initial model.

Line 5: CONTROL_MODEL_INDEX *filename*/default

CONTROL_MODEL_INDEX requires filename (including its path) of the control model index file. This index file is required when prior information, such as ocean, is known and need to be fixed in the inversion. If no prior information, "default" should be used, in which case all model parameters are free to change.

Line 6: TARGET_RMS *real*

TARGET_RMS requires *real* number. This is the desired normalize root mean square (RMS) misfit to the data defined as $\|\mathbf{C_d}^{-1}(\mathbf{d-F[m]})\|/N$. The goal of the inversion is to find the minimum norm model subject to this RMS. After completing Phase I (i.e., reaching this desired RMS), the inversion starts Phase II by keeping the misfit at this desired level but trying to decrease the norm of the model $\|(\mathbf{m-m_0})^T\mathbf{C_m}^{-1}(\mathbf{m-m0})\|$. A typical value for this parameter is 1.

Line 7: MAX_NO_ITERATION *integer*

MAX_NO_ITERATION requires *integer* value. This indicates the maximum number of iteration (outer loop) for inversion. The program will stop if the number of iteration exceeds this value, even thought the desired RMS has not reached. If 0 (zero) is input here, the program will ONLY do the initial forward modeling step for the given initial model.

Line 8: MODEL_LENGTH_SCALE default/ *integer real real real*

MODEL_LENGHT_SCALE requires 4 parameters; 1 integer and 3 reals. These parameters control characteristic of model covariance ($\mathbf{C_m}$) which is obtained from solving a diffusion equation. The de-correlation scale for the diffusion equation in all directions varies in space, with length scales proportional to $\sqrt{(4\delta\tau)}$ of the local grid resolution, where δ (between 0 and 1) and τ are given by users. For more detail on the model covariance, please refer to Siripunvaraporn et al. (2005) and Siripunvaraporn and Egbert (2000). If set as "default", the program will automatically use pre-assigned values, i.e. 5 0.1 0.1 0.1.

- The first integer indicates the time step. The default is 5. Higher value requires longer run time and generates smoother model. Small value requires less cpu time and less smoother model.
- The next 3 real values indicate δ (between 0 and 1) in x-, y- and z- directions.

Line 9: LAGRANGE_INFO default/*real real*

LAGRANGE_INFO requires 2 real values or "default". In Occam's style inversion, the program will automatically search for λ to minimize the model norm subject to the desired RMS misfit. For more detail on Lagrange multiplier, please refer to Siripunvaraporn et al. (2005) and Siripunvaraporn and Egbert (2000). The first value indicates the starting λ. The second indicates the step size of λ in log10 scale. The "default" are 1. and 0.5, respectively.

Line 10: ERROR_TOL_LEVEL default/ *real real real integer*

ERROR_TOL_LEVEL requires 3 real values and 1 integer or "default". It gives tolerances for solving forward problems for responses and for sensitivity calculation. The last integer is the maximum number of iteration for the 3-D forward solver. The first tolerance level is used when the forward problem is solved for the final model of each iteration. The second tolerance level is used when the forward problem is solved while searching for optimum λ. The third tolerance level is used when the forward problem is solved to obtain the sensitivities. See Siripunvaraporn et al. (2005) for more detail. The defaults are 10^{-7}, 10^{-7}, 10^{-4} and 2000, respectively.

3.2 Data file

Example of data input file

```
36 5 8                                                  % first line
Station_Location: N-S                                   % second part
-24500. … -24500.
…
24500. …  24500.
Station_Location: E-W
-24500. … 24500.
…
-24500. … 24500.
DATA_Period: 1.                                         %impedance tensor data
0.2528E-05 -0.7146E-05 0.1979E-01 -0.1958E-01 -0.1979E-01 0.1958E-01 -0.2735E-05 0.6835E-05
…
DATA_Period: 3.
…
…
DATA_Period: 100.
…
ERROR_Period: 1.                                        % error bar
0.2528E-06 -0.7146E-06 0.1979E-02 -0.1958E-02 -0.1979E-02 0.1958E-02 -0.2735E-06 0.6835E-06
…
ERROR_Period: 3.
…
…
ERROR_Period: 100.
…
ERMAP_Period: 1.                                        % error scale map
1. 1. 1. 1. 1. 1. 1. 1.
…
…
ERMAP_Period: 100.
....
```

Description of data file

There are 5 sections in data file. All parameters must be in this order with no empty lines in between.

1. The first lines of the data file requires 3 integers for number of stations, number of periods and number of responses. All of these values should not exceed the

parameters set in "para.h". Only 4 and 8 responses are supported corresponding to the full impedance and the off-diagonal impedance, respectively.

2. The second section is the 2nd and 3rd lines. They are the station locations in x- (North-South) and y- (East-West) directions.

- The station locations (in meters) must be listed after the keyword <Station_Location: N-S> and <Station_Location: E-W>.
- Note that, for this version of the program, **station locations must be located ONLY at the middle of the grid at the surface**. Therefore, users should design the grid to match with the station locations.
    - *Tip*: In some cases, a station location can be moved a small distance from its actual location without distorting results.
- The origin in the x- and y-direction of model file is at the middle of the model domain. In plane view coordinate, it increases from bottom to top in x-, from left to right in y-, and from the surface down in z-.

3. Next section (3rd) is the impedance tensor data.

- Starting from the first to last period, the period value must be entered after the <DATA_Period:> keyword.
- Next line is the impedance tensor of the first station. Each line for each station, repeated till the last station.
- If the number of responses is 8, each line must consists of *REAL*(Z_{xx}), *IMAG*(Z_{xx}), *REAL*(Z_{xy}), *IMAG*(Z_{xy}), *REAL*(Z_{yx}), *IMAG*(Z_{yx}), *REAL*(Z_{yy}) and *IMAG*(Z_{yy}), in this order.
- If the number of responses is 4, each line must consists of *REAL*(Z_{xy}), *IMAG*(Z_{xy}), *REAL*(Z_{yx}), and *IMAG*(Z_{yx}), in this order.
- The time dependence used here is $e^{-i\omega t}$. **Note that this might differ from the phase converter used in some processing codes.**
- The unit of impedance Z is ohm. **Make sure you use the same unit. (see next page)**

Warning: Most data processing codes generate impedance in the unit of **mv/km/nT**. Here, the impedance is in the unit of **ohm** or **[V/m]/[A/m]**. The conversion between these two units is given below:

$$1 \text{ Ohm} = 1 \text{ [V/m]/[A/m]} = 10^4/4\pi \text{ mV/km/nT} = 796 \text{ mV/km/nT}$$

The difference of the unit comes from using different definition of Z.

Z = E/H : [V/m]/[A/m] = Ohm Z = E/B : [mV/km/nT]

4. Next section (4^{th}) is the error bar of the impedance tensor. The format is the same as the data section. The keyword is <ERROR_Period:>. The period value must be followed this keyword, and in the same order as the period used in the data section.

5. Last section (5^{th}) is the error scale map. The format is the same as the data and error sections. The error scale map is used to increase or decrease the error bar by directly multiplying it with the error bar. If set to 1., the given error bars remain unchanged. This feature is good if users want to exclude bad data at some sites and periods, the error scale map at that station and period should be set to very large number, such as 999.

3.3 Initial model

Example of initial model file

```
#Initial model                          % Title
21 28 11 2                              % first line: Nx Ny Nz and Nr (number of resistivity index)
20000. … 20000.                         % block size in x-
20000. … 20000.                         % block size in y-
500. 1000. … 40000.                     % block size in z-
50. 10.                                 % resistivity values
1 5                                     % beginning and end layer
1 1 1 1 1 … 1 1 1 1 1 1                 % resistivity index
…
1 1 1 1 1 … 1 1 1 1 1 1
6 11                                    % beginning and end layer  (repeated till end layer = Nz)
2 2 2 2 2 … 2 2 2 2 2 2                 % resistivity index
….
2 2 2 2 2 … 2 2 2 2 2 2
```

Description of initial model file

There are 5 sections in the initial model file. All parameters must be in the given order.

1. The first section is the first line of the file and it is the title of the model.

2. The second line of the initial model file requires 4 integers for number of blocks in x-, y- and z- directions, respectively. All of these values should not exceed the parameters set in "para.h" file. The last integer indicates the number of resistivity indices used in this file. (for detail on format, see resistivity format end of this section)
   - If the last integer is set to 0, the input resistivity model is not in index format, but in real format.
   - If set to 1, the input resistivity is in half-space format. If greater than 1, the input resistivity is in index format.
   - The last number can not exceed 9 for this current version.

3. The 3rd section gives block sizes in x-, y- and z-, respectively (in meters). The origin in x- and y-direction of model file is at the center of model domain and increasing from bottom to top (south to north) in x-, and increasing from left to right (west to east) in y-direction, and from the surface down in z-direction.
   - **Tip:** the summation of block size in x- should be larger than the length of the profile in x-direction, similarly for y-direction.
   - **Make sure your model is designed correctly, and your station locations match correctly with the model block size you designed here!**

4. The 4th line is the resistivity values corresponding to each resistivity index. If the last integer of the first line is N, then there must be N resistivity values. Each value represents one resistivity. If the last integer of 1st line is 0, this line must be omitted.

5. The 5th line is the input resistivity model.
   - If the last integer of 1st line is 0, the actual resistivity values are specified in *real* format and read with the following code.

```
DO iz = 1,Nzb                    % from 1st layer to last layer
DO iy = 1,Ny                     % from left (west) to right (east)
DO ix = Nx,1,-1                  % from top (north) to bottom (south)
READ(10,*) resis….
ENDDO
ENDDO
ENDO
```

   - If the last integer of the 1st line is 1, then the resistivity value is half-space. The program will insert the real resistivity value from section #4 into the model, so this section can be omitted.

   - If the last integer of the 1st line is greater than 1, then the resistivity value is listed in *integer* index format for each layer from the surface down. The user must specify the beginning and end layer that use the same resistivity index. Then follow by the resistivity index. The total number of layer and the last layer must be equal to Nz (number of block in z-direction) specifying in the first line.

Example of resistivity index:

```
1 5                          ← from 1st layer to 5th layer
1 1 1 1 1 ... 1 1 1 1 1 1    ← Northernmost (x-axis) and west to east (y-axis)
...
1 1 1 1 1 ... 1 1 1 1 1 1    ← Southernmost (x-axis) and west to east (y-axis)
6 11                         ← from 6th layer to last layer
2 2 2 2 2 ... 2 2 2 2 2 2    ← repeat above
....
2 2 2 2 2 ... 2 2 2 2 2 2
```

3.4 Prior model file

The prior model is $\mathbf{m_0}$ in equation (1). The inversion program will search the model **m** around this prior model. The "default" prior model file is the same as the initial model file. The format of the prior model is the same as the format of the initial model file. However, the prior model could be different from the initial model.

3.5 Control model index

If users want to include "known" features in the model, such as the ocean, this control model index file is important. Otherwise the defaults should be used, i.e. all cells of the model are free to change.

In the current version, the only control allowed is freezing part of the model. To freeze part of the model, such as ocean, the user must provide the control model index file. The format of the control model index is almost the same as the initial model file, except that there is no block size. Also, the index format is the same as for the resistivity index, but only 0 and 1 are allowed. One (1) means cell is fixed and resistivity of this cell can not be changed, while 0 (or anything) means cell is free to change.

Example of control model index file.

```
21 28 11                ← Nx, Ny and Nz
1 11                    ← 1st layer to last layer (z-dir)
0 0 0 0 0 ... 0 0 0 0 0 ← Northernmost (x-axis) from west to east (y-axis)
...
0 0 0 0 0 ... 0 0 0 0 0 ← Southernmost (x-axis) from west to east (y-axis)
```

4. OUTPUTS

There are several output files generated by the program: data and error files, and for each iteration and for each λ (this is useful for some cases), the model and response files. In addition, the log file is also generated.

- The data and error files display the data and error used in the inversion. The data and error files use the output "filename" given in the "startup" file and end with a suffix ".data", and ".error", respectively. The format of the data and error output files are the same as the data input files described in section 3.2, except that they are shorter only the "data" section in the data output file, and "error" section in the error output file.

  Example of data and error filename:
  prism3d.data for data file and prism3d.error for error file.

- The other files are the model files and response files for each iteration and for each λ.
  - The final models and responses for each iteration will end with a suffix "_model.XX" and "_resp.XX", respectively, after the given "filename", where "XX" is the number of the iteration.
  - For each iteration, various values of λ are used to search for the minimum RMS misfit in Occam's Phase I and to search for minimum norm in Phase II (see Siripunvaraporn et al., 2005 for more detail). The program will also generate the model and response files for each λ used, and end with a suffix "_model.XX_YY" and "_resp.XX_YY", respectively, where "YY" is the integer index indicating the order of λ used. The format of the model file is the same as the input initial model file with "real" format. The first line of the model file display is the information about the RMS misfit of the model and the value of λ used in the title. The format of the response file is the same as the data output file. The end of the file shows information about RMS misfit and the value of λ used. Because the format of the output are the same as input, it can then be used as an input file for later inversion.

Example of filename:

prism3d_model.01_01, prism3d_model.01_02, …, prism3d_model.01_07 and prism3d_resp.01_01, prism3d_resp.01_02, …, prism3d_resp.01_07 are model and response files after each λ used of the 1st iteration. The best model which generate smallest RMS will be used for the next iteration.

prism3d_model.01 and prism3d_resp.01 are model and response files after 1st iteration. This model generate the smallest RMS after several trial of λ.

- The program writes various information used for the run to the log file. The log file ends with ".log" after the given filename.

5. ACKNOWLEDGEMENT

This work has been supported by Thailand Research Fund (PDF/37/2543; RSA4780021) to WS, and partly by a JSPS postdoctoral fellowships (P02053) and MEXT, Japan (02053) to WS and MU, and by Department of Energy (DOE-FG0302ER15318) to GE.

6. REFERENCES

- Siripunvaraporn W., G. Egbert, Y. Lenbury and M. Uyeshima, 2005a, Three-Dimensional Magnetotelluric: Data Space Method, *Physics of the Earth and Planetary Interiors*, 150, 3-14.
- Siripunvaraporn W., G. Egbert and M. Uyeshima, 2005b, Interpretation of 2-D Magnetotelluric Profile Data with 3-D Inversion: Synthetic Examples, *Geophys. Jour. Inter.* , 160, 804-814.
- Siripunvaraporn W., G. Egbert and Y. Lenbury, 2002, Numerical Accuracy of Magnetotelluric Modeling: A Comparison of Finite Difference Approximation, *Earth Planets Space*, 54, 721-725.
- Siripunvaraporn W. and G. Egbert, 2000, An efficient data-subspace inversion method for 2-D magnetotelluric data, Geophysics, 65, 791-803.

Thank you very much.

Weerachai Siripunvaraporn, Gary Egbert and Makoto Uyeshima

Copyright © 2006 Weerachai Siripunvaraporn