MTZ3Dinv

A Program Library for Inversion of Magnetotelluric and Magnetovariational Data over 3D Structures

VERSION 2011

Developed under the IMAGE consortium research project

INVERSION AND MODELLING OF APPLIED GEOPHYSICAL ELECTROMAGNETIC DATA

UBC–Geophysical Inversion Facility
Department of Earth and Ocean Sciences
University of British Columbia
Vancouver, British Columbia

UBC – Geophysical Inversion Facility
January 2011

Table of Contents

1. Introduction to MTZ3Dinv

MTZ3Dinv is a program library for inverting magnetotelluric and/or Magnetovariational data to recover 3D models of conductivity in an Earth model that is discretized using an irregular mesh of rectangular cells. The data to be inverted can be either the elements of the impedance tensor, magnetivariational transfer functions (Tippers) or the associated apparent resistivities and phases. The inversion program can be run from the command line under LINUX, or MS-Windows or, in the MS Windows environment, using a graphical user interface (GUI). The UBC-GIF utility MeshTools3D is used to examine resulting 3D conductivity models.

As of 2010, the code has been updated to work with transfer function data, along with impedance MT data and furthermore, the code has been parallelized with open MP and MPI. MPI parallelization was used for variety of frequency data and will be discussed below in greater detail, while open MP parallelization is used for two orthogonal source polarization directions for each frequency. Hence, for the frequency-based parallelization, the number of processors defined under MPI settings can not exceed the number of frequencies described in the data file. Each of these frequency-oriented processes will be further split up in two source polarization threads using the open MP environment.

This document provides details regarding the use of MTZ3Dinv. The bulk of this manual provides details on file structures and parameters for controlling the calculations. Other related documentation includes:

- Papers published in the scientific literature.
- A description of the differences in the theoretical backgrounds of MTZ3Dinv and its sister program H3Dinv (file name `mt3dinv-overview.pdf`).
- A manual for the Graphical User Interface (GUI), provided in HTML format (mt-gui.html).
- A summary of the log file produced by the inversion as a summary of how calculations proceeded, also in HTML format (`mt3dinv-log.html`).
- A summary of the data, inputs, log, and resulting model for a simple synthetic case involving a single conductive block in a small mesh.

Experienced users of inversion understand that fine tuning the parameters concerned with computational accuracy can affect the efficiency of convergence. In principle, one wants to compute all quantities as accurately as possible and solve the matrix systems exactly. Unfortunately that can lead to prohibitively large computational costs and so strategies that reduce the computations, and yet do not compromise the final model, are sought. For this reason there are two levels in which MTZ3Dinv can be run. The first uses all default parameters. In the second, the user can adjust tolerance, maximum number of iterations, etc. to gain computational efficiency. In order to adjust these parameters in a meaningful way, the user needs to understand the basic structure of the code and the parameters that control the calculations. Therefore it is important to read the overview of background theory (in the PDF documents mt3dinv-overview.pdf and eh3dinv-overview.pdf), and to use the manual that follows to understand exactly what each parameter does.

2. Files used by the MTZ3Dinv Program Library

2.1 Introduction

The inversion program requires input files, as well as parameters, in order to run. Some files are used by a number of programs. Before detailing the procedures for running MTZ3Dinv, the necessary files are described.

2.2 General Files for MTZ3Dinv Programs

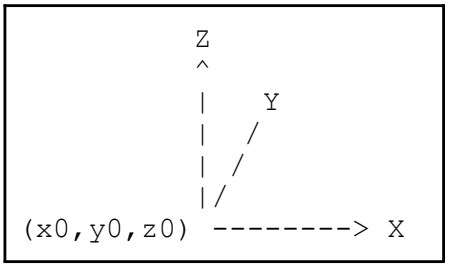
The general files which contain data or models or parameters or information about the inversion are:

1. mesh.txt
2. data.txt
3. weight.txt
4. dobs.txt
5. dpred.txt
6. mt3dinv.inp
7. mt3dinv.out
8. bicg.txt
9. mt3dinv.log
10. model.con
11. topo.dat

2.2.1. File: mesh.txt

This file contains the 3D mesh which defines the model region. It has the following format:

```
nx  ny  nz
x0 y0 z0
dx_1  dx_2  ...  dx_nx
dy_1  dy_2  ...  dy_ny
dz_1 dz_2 ... dz_nz
```



Coordinate axes for UBC-GIF meshes.

Parameters are defined as follows:

`nx,ny,nz` - number of cells in the X, Y, and Z directions;
`x0,y0,z0` - coordinates of the top south west corner of the mesh;
`dx,dy,dz` - cell widths.

Mesh design is a critical step. Stable inversion is more likely if adjacent cells do not change size radically. For example, padding cells around a zone with uniform cells should increase in size by a factor of around 1.5. See the simple block example below. The mesh created by the Graphical User Interface is described in the GUI manual.

2.2.2. File: data.txt

The file containing measurements that are to be inverted is complicated because it is versatile. It has been updated from the previous version of the code. In addition to magnetotellouric sounding data it is now possible to invert magnetovariational data (MV), including airborne Tipper surveys (ZTEM), where the airborne coil is only measuring the vertical component of the secondary magnetic field, while the Tipper transfer functions are calculated, taken the horizontal fields measured at the base station on the surface (see the theoretical background for details). It is now also possible to accommodate a combined data set (MT and ZTEM or MT and MV).

Note that line numbers must NOT be included - they are shown in the figure to help with the explanations for each line given after the figure. Note that in the data file, any lines starting with "!" are treated as comment lines and are ignored.

MT data alone

```
1) DATATYPE MTZ
2) IGNORE -i

3) f1     ! frequency
4) nr     ! # of receivers
5) x  y  z  rZxx er iZxx er rZxy er iZxy er rZyx er iZyx er rZyy er iZyy er
6) 5  5  0    1  .1   2  .2   3  .3   4  .4   -0 -0  -0  -0   6  .6  6  .6 (values for illustration only)
7) 6  6  0    1  .1   2  .2   3  .3   4  .4   -0 -0  -0  -0   6  .6  6  .6
ETC.

3) f2     ! frequency
4) nr     ! # of recievers
5) x  y  z  rZxx er iZxx er rZxy er iZxy er rZyx er iZyx er rZyy er iZyy er
6) 5  5  0    1  .1   2  .2   3  .3   4  .4   -0 -0  -0  -0   6  .6  6  .6
7) 6  6  0    1  .1   2  .2   3  .3   4  .4   -0 -0  -0  -0   6  .6  6  .6
ETC.

REPEAT FOR MORE FREQUENCIES AS NECESSARY
```

MV/ZTEM data alone

```
1) DATATYPE MTT
2) IGNORE -i

3) f1     ! frequency
4) nr     ! # of receivers
5) xb yb zb iiiiiiii   !xbase ybase zbase number of channels (always 8)
5) x  y  z  rTX erTX iTX eiTX rTY erTY iTY eiTY
6) 5  5  0    1  .1   2  .2    3  .3    4   .4                (values for illustration only)
7) 6  6  0    1  .1   2  .2    3  .3    4   .4
ETC.

3) f2     ! frequency
4) nr     ! # of receivers
5) xb yb zb iiiiiiii   !xbase ybase zbase number of channels (always 8)
5) x  y  z  rTX erTX iTX eiTX rTY erTY iTY eiTY
6) 5  5  0    1  .1   2  .2    3  .3    4   .4
7) 6  6  0    1  .1   2  .2    3  .3    4   .4
ETC

REPEAT FOR MORE FREQUENCIES AS NECESSARY
```

Joint MT/ZTEM data file

```
1) DATATYPE MTB
2) IGNORE -i

3) f1     ! MT frequency
4) nr     ! # of receivers
5) xb yb zb iiiiiiiiiiiiiiiiiiiiiiii   !xbase ybase zbase (ii..) - number of data channels (always 24)
6) x  y  z  rZxx er iZxx er rZxy er iZxy er rZyx er iZyx er rZyy er iZyy er   ZTEM dummied out (8 columns)
7) 5  5  0    1  .1   2  .2   3  .3   4  .4   -0 -0  -0  -0   6  .6  6  .6    i i i i i i i i
8) 6  6  0    1  .1   2  .2   3  .3   4  .4   -0 -0  -0  -0   6  .6  6  .6    i i i i i i i i
ETC.
1) f2     ! ZTEM frequency
2) nr     ! # of recievers
xb yb zb  ! xbase ybase zbase
3) x  y  z    MT data (dummied out)16 columns rTX erTX iTX eiTX rTY erTY iTY eiTY
4) 5  5  0    i i i i i i i i i i i i i i i i  1  .1    2  .2    3   .3   4  .4
5) 6  6  0    i i i i i i i i i i i i i i i i  1  .1    2  .2    3   .3   4  .4
ETC.


REPEAT FOR MORE FREQUENCIES AS NECESSARY
```

Data type

Lines 1 - 2 specify how the remainder of the file will be read. Input data can be provided in four different ways:

1. As real and imaginary parts of the elements of the impedance tensor (MTZ).
2. As apparent resistivities and phases (MTR).
3. As Real and Imaginary parts of tipper functions in units of absolute ratio (MTT).
4. As joint dataset of tipper matrix elements and impedance tensor elements (MTB)

The type of input data has to be specified, using the first line of the file as follows. Data must then be given with components in column format (as shown below) starting after the number of receivers is specified.

```
DATATYPE MTZ
rZxx iZxx rZxy iZxy rZyx iZyx rZyy iZyy

DATATYPE MTR
Zxx-rho Zxx-ph Zxy-rho Zxy-ph Zyx-rho Zyx-ph Zyy-rho Zyy-ph

DATATYPE MTT
reTX iTX rTY iTY

DATATYPE MTB
rZxx iZxx rZxy iZxy rZyx iZyx rZyy iZyy rTX iTX rTY iTY
```

Note: The last data type MTB is only applicable in provided format, when ZTEM frequency is identical to MT frequency. If the frequencies do not match, then they have to be listed separately, however the number of columns in the data file should still be sufficient to accommodate all possible data. The unused columns should be dummied out, using the dummy value, specified in line 2 of the data file.

Ignore

Line 2 specifies a character string that you use to signify no data in a column. In the example above, the character string "-0" is used (a single character could be used too). If not applicable, this line can be omitted.

Frequency

Line 3 is the frequency at which the next set of data were collected, specified as a real number in units of Hertz.

Number of receivers

Line 4: After the frequency, the number of receivers with data is specified as an integer.

Line 5 (This is a special line only present in ZTEM or joint ZTEM/MT data files). It describes the ZTEM base station location, which should be within the extent of the mesh. If the actual base station is too far away, it is acceptable to arbitrary edit the location.

Data

Line 6 (in case of classic MTZ or MTR data file, line 5) is included above for demonstration only. DO NOT INCLUDE THIS IN A REAL DATA SET.

Following the number of receivers, data values at each receiver location are provided on a single line using columns as specified above under "Data type".

Once all data gathered at one frequency have been specified, data gathered at a new frequency can be listed by repeating the procedure and appending the new frequency data to the previous frequency data.

NOTES:

1. Do not include a line with the column labels (line 5).
2. As many frequencies as desired can be included.
3. See the simple buried conductive block example for examples of field formats.

2.2.3 File: `weight.txt`

User supplied values of any depth weighting to be applied. The format of this file is the same as that of a model file. In other words, the weighting value for each cell is specified on one line in the order described for the "`model.con`" file.

The weighting matrix is generally used to implement a depth weighting which may be needed when there are few source locations or few frequencies; that is, when the data do not adequately constrain the distribution of conductivity with depth. In this circumstance, the model recovered without any weighting may extend to the surface. The weighting matrix can be used to find solutions with alternative distributions of conductivity with depth. The methodology is similar to that employed when inverting surface gravity or magnetic data, so the
weights in the matrix can be inversely proportional to some power of the depth of the associated cell. Only relative values matter, and regions where cells have smaller weights will have more structure. For example, assigning weights of surface cells to a value of unity, and deeper cells to values between 0 and 1 will force structure to occur at greater depths. Weights for inactive cells, including the air, do not play a role in the solution and so they can be assigned any number.

2.2.4 Files: `dobs.txt`

(This file is made by the program.) The observed data are extracted by the program from the data input file and saved to the `dobs.txt` file. The format is same as the data portion of the data input file, but without the header.

2.2.5 Files: `dpred.txt`

(This file is made by the program.) The predicted data, calculated by forward modelling the data that would be observed over the Earth model resulting from the inversion, is saved in the dpred.txt file in the same format is the dobs.txt file. This file is updated each time predicted data are computed for a new model in the inversion process.

2.2.5a File: `dpred_j.txt`

The predicted data for the model produced for the jth value of beta.

2.2.6 File: `mt3dinv.inp`

This is the control file containing all references to necessary files and inversion parameters. It is described in detail below in section 3.3.1.

2.2.7 File: mt3dinv.out

A summary of the inversion algorithm's progress. This file basically gathers summary lines of information from the log file (mt3dinv.log) so that values can be assessed together and plotted using a graphing tool if desired. The eight columns are identified with a title row. Values can be understood best by correlating them with the description of the log file (section 3.3.4 below).

2.2.8 File: bicg.txt

This output file contains a summary of BiCGSTAB calculations that are done when solving the forward and adjoint problems. It is usually used for diagnostics, to see how well systems are being solved. For example:

```
          Frequency      #i Residual
  eh      1.00000E+01    3  1.97019E-12
  eh      1.00000E+01    3  2.55530E-12
        1  1.00000E+01    1  6.51573E-10
        1  1.00000E+01    1  6.42418E-10
        1  1.00000E+01    1  7.28805E-09
        1  1.00000E+01    1  7.19741E-09
        1  1.00000E+01    1  6.51573E-10
        1  1.00000E+01    1  6.42418E-10
  adj   1  1.00000E+01    4  7.45414E-11
  adj   1  1.00000E+01    4  7.34665E-11
        1  1.00000E+01    1  2.67982E-09
        1  1.00000E+01    1 2.51723E-09
```

```
The 5 columns (left to right) are:
Label, frequency index, value of frequency, number
of
iterations, value of residual.
Label "eh" means the 4 numerical columns refer to
primary
field computations.
Label "adj" means the 4 numerical columns refer to
the
adjoint solution.
```

2.2.9 File: mt3dinv.log

Progress of the inversion program as it works through all the necessary iterations is recorded in the mt3dinv.log file. This file is used to assess the inversion process, and to determine if the algorithm proceeded in an acceptable manner. Details are provided in section 3.3.4.

2.2.10 File: inv.con

This file contains the cell values of the conductivity model resulting from the inversion. This file is in the standard
UBC-GIF 3D model format, with one cell value per line as follows:

$cond_{1,1,1}$
$cond_{1,1,2}$
:
$cond_{1,1,NV}$
$cond_{1,2,1}$
:
$cond_{i,j,k}$
:
$cond_{NN,NE,NV}$

where $cond_{i,j,k}$ is the conductivity at location [i j k], and [i j k] = [1 1 1] is defined as the cell at the top-south-west corner of the model. The total number of lines in this file should equal NN x NE x NV, where NN is the number of cells in the North direction, NE is the number of cells in the East direction, and NV is the number of cells in the vertical direction. The lines must be ordered so that k changes the quickest (from 1 to NV), followed by j (from 1 to NE), then followed by i (from 1 to NN). If the surface topography (topo.dat) file is supplied, the values above the surface will be ignored in the inversion. These

values should be assigned 1.0 x 10^{-8} to avoid confusion with the other model elements. This file is updated each time a new model is computed in the inversion process.

2.2.10a File: `inv_j.con`

The conductivity model produced for the jth value of beta.

2.2.11 File: `topo.dat`

This optional file is used to define the surface topography of the 3D model by specifying the elevation at different locations. If `topo.dat` is not supplied, the surface will be treated as being flat. topo.dat has the following structure:

```
! comment
!
npt
E1 N1 elev1
E2 N2 elev2
:
Enpt Nnpt elevnpt
```

`! comments` – top lines beginning with ! are comments.
`npt` – number of points.
`Ei, Ni, elevi` – Easting, Northing and elevation of the i^{th} point. The elevation in this file and Zo in the mesh file must be specified relative to a common reference.

The lines in the `topo.dat` file can be in any order as long as the total number is equal to npt. The topographic data need not be supplied on a regular grid. UBC-GIF programs assume a set of scattered points for generality and they use triangulation-based interpolation to determine the surface elevation above each column of cells. To ensure the accurate discretization of the topography, it is important that the topographic data be supplied over the entire area above the model and that the supplied elevation data points are not too sparse.

Example of `topo.dat` file:

```
!! topographic data
4
0.0 0.0 50.0
0.0 1000.0 50.0
1000.0 0.0 -50.0
1000.0 1000.0 -50.0
```

NOTE: Although the cells above the topographic surface are removed from the model, they must still be included in the model file, as if they are a part of the model. For input model files, these cells can be assigned any value.

The recovered model produced by inversion programs also includes the cells that are excluded from the model, but these cells will have a value of 1.0 x 10^{-8} to identify them.

3. Running MTZ3Dinv Main Programs

3.1 Introduction

The MTZ3Dinv library consists of programs for forward modelling and inverting MT data, and a Graphical User interface (GUI) for setting up inversions. For optimal performance it is best to run the program using the Message Pass Interface (MPI, discussed below).

1. `mt3dfwd.exe` (section 3.2 below)
2. `ZTEM_MTZ3Dinv_64.exe` (section 3.3 below)
3. `mt3dgui.exe` (section 3.4 below)

3.2 MTZ3Dfwd

MTZ3Dfwd is a magnetotelluric forward-modelling program for 3D conductivity models. The operation of MTZ3Dfwd is very similar to that of its controlled-source counterpart EH3D. The reader is therefore referred to the documentation for EH3D.

MTZ3Dfwd uses the same total-field mode of solution as the routine within MTZ3Dinv that computes the primary fields. (The remainder of the forward solutions, and all the adjoint solutions, within MTZ3Dinv use the primary-secondary mode of solution.) The 3D conductivity model is required in standard UBC-GIF 3D model format. The appropriate boundary conditions are determined within MTZ3Dfwd using a 2D E-polarization forward-modelling routine. This implicitly assumes that the 3D model becomes sufficiently 2D as one approaches each vertical boundary of the mesh. In order to compute the MT impedances, the electric and magnetic fields are required for two different polarizations of the inducing field (see eq. 8 in "Theoretical Background for the Magnetotelluric Inversion Program MTZ3Dinv"). Polarization 1 is for a unit, x-directed H-field at the very top of the mesh, and polarization 2 is for a unit, y-directed H-field at the top of the mesh.

There are a number of possible outputs from MTZ3Dfwd:

- the real & imaginary parts of the elements of the impedance tensor at a list of observation locations whose coordinates are supplied in a separate input file;
- the apparent resistivities & phases at a list of observation locations;
- the real & imaginary parts of the components of the E- & H-fields for the two polarizations interpolated to a list of observation locations;
- the real & imaginary parts of the components of the E- & H-fields for the two polarizations everywhere on the mesh.

3.2.1 MTZ3Dfwd input file format

The operation of MTZ3Dfwd is controlled by a series of parameters that are provided to the program via an input file that must be called `mt3dfwd.inp`. Note that file names and paths should not contain spaces.

The input file is a text file with 10 lines containing parameters as follows (the "|" symbol means "or"). Each line is described in detail below.

```
1) frequency
2) mesh file
3) conductivity file
```

```
4) susceptibility file | null
5) locations file
6) solver parameters: droptol, tol, iter
7) 4 (preconditioner type)
8) 0|1 calculate fields
9) 0|1 output all fields
10)0|1 output data at certain locations
```

Input File Line 1:

Source frequency in units of Hz. (MTZ3Dfwd considers only one frequency at a time: results for multiple frequencies require multiple runs of the program.)

Input File Line 2:

The name of the file that defines the 3D mesh. This mesh file has the following format (the standard UBC-GIF 3D mesh file format):

```
nx
ny
nz
x0
y0
z0
dx_1 dx_2 ... dx_nx
dy_1 dy_2 ... dy_ny
dz_1 dz_2 ... dz_nz
```

Parameters are defined as follows:

`nx, ny, nz` - number of cells in the X, Y, and Z directions;
`x0, y0, z0` - coordinates of the top south west corner of the mesh;
`dx, dy, dz` - cell widths in the z-, y- & z-directions.

Mesh design is a critical step. There are guidelines in "Practical aspects of running EH3D" in the IMAGE Annual Report for 2001.

Input File Line 3:

The name of the file containing the conductivity for all cells in the model. This model file is in the standard UBC-GIF 3D model format. MTZ3Dfwd considers only real conductivities, and expects values in units of S/m. For a homogeneous halfspace model, a value in units of S/m can be entered on this line. All cells below z=0 will then be set to this value. The cells above z=0 will be set to 1.0E-08 S/m.

Input File Line 4:

The name of a file containing magnetic susceptibility in SI units for all cells in the model. It has the same format as that of the file identified on Line C.

For a homogeneous halfspace model, a value in SI units can be entered. All cells below z=0 will then be set to this value. The cells above z=0 will be set to 0 (SI units). If "null" is entered, all cells will be set to 0 (SI units).

Input File Line 5:

The name of a file containing the x-, y- & z-coordinates of the locations at which the impedances,

apparent resistivities & phases, and/or fields are desired. Linear interpolation is used to obtained the fields at the required locations from the values on the mesh. Impedances (and apparent resistivities & phases) are calculated from the interpolated values of the fields. The file has the format:

```
x
1
y
1
z
1
x
2
y
2
z
2
x
3
y
3
z
3
```

where (`x1,y1,z1`) are the coordinates of the first observation location, (`x2,y2,z2`) are the coordinates of the second location, etc.

Input File Line 6:

Parameters specifying convergence criteria:

`Droptol`	- sets the threshold for dropping small terms in the ILU factorization;
`tol`	- tolerance for convergence;
`iter`	- maximum number of iterations to perform.

Input File Line 7:

Preconditioner type. Was hardcoded in the previous version,l can be 0 or 4. 0 corresponds to ssor preconditioner, 4 corresponds to blugs preconditioner. In general ssor preconditioner uses less memory, but converges slower (is recommended for older computers and large problems). Blugs is faster to converge, but uses more memory.

Input File Line 8:

Flag that indicates whether a complete forward-modelling should be performed, or whether only impedances and/or apparent resistivities & phases are required, given previously computed E- & H-fields.

`1` - Calculate E- and H-fields everywhere;

`0` - read E and H fields from files: e1.dat, e2.dat, h1.dat, h2.dat.

Input File Line 9:

Flag that indicates whether or not the E- & H-fields everywhere on the mesh are required as output.

`1` - Output files containing E- and H-fields everywhere: e1.dat, e2.dat, h1.dat, h2.dat;

`0` - do not output.

Input File Line 10:

Flag that indicates whether or not data is desired at a list of observation locations.
1 - Output data. The following three files are created:

`MT_fields.txt` - E- and H-fields (real and imaginary, both polarizations). `MT_impedance_ri.txt` - real and imaginary Z. `MT_impedance_rho_ph.txt` - apparent resistivity and phase of Z.

0 - do not output data.

3.2.2 MTZ3Dfwd outputs

There are two main types of output from MTZ3Dfwd: values of the E- & H-fields everywhere on the mesh, and values of the fields interpolated to a series of observation locations and their associated impedances and apparent resistivities & phases.

If line 9 of `mt3dfwd.inp` contains 1 (see above), the E- & H-fields everywhere on the mesh for the two source polarizations are output (in binary format) in files `e1.dat, e2.dat, h1.dat, h2.dat`. These files are in the format required by the program dat2modMT (which is the MT equivalent of program dat2mod: see IMAGE Annual Report for 2001) that converts the fields into a format that can be viewed using MeshTools3D.

If line 10 of `mt3dfwd.inp` contains 1 (see above), the E- & H-fields are linearly interpolated to the list of observation locations provided in the file given on line E of mt3dfwd.inp. The interpolated field values are written out to the file `MT_fields.txt,` and the associated elements of the impedance tensor and the apparent resistivities and phases are written out to files `MT_impedance_ri.txt` and `MT_impedance_rho_ph.txt` respectively.

There are 27 columns in file `MT_fields.txt`:

Xr x-coordinate (m) of observation location,
Yr y-coordinate (m), Zr z-coordinate (m),
Ex1r real part of x-component of E-field (V/m) for polarization 1,
Ex1i imaginary part of x-component of E-field (V/m) for polarization 1,
Ey1 r real part of y-component of E-field (V/m) for polarization 1,
Ey1 i imaginary part of y-component of E-field (V/m) for polarization 1,
.
.
.
Hz2 i imaginary part of z-component of H-field (A/m) for polarization 2.

There are 11 columns in file `MT_impedance_ri.txt`:

Xr x-coordinate (m) of observation location,
Yr y-cooordinate (m),
Zr z-coordinate (m),
Zxxr real part of xx-element of the impedance tensor (V/A),
Zxxi imaginary part of xx-element of the impedance tensor (V/A),
Zxyr real part of xy-element of the impedance tensor (V/A),
.
.
.

Zyy i imaginary part of yy-element of the impedance tensor (V/A).

There are 11 columns in file `MT_impedance_rho_ph.txt`:

Xr x-coordinate (m) of observation location,
Yr y-coordinate (m), Zr z-coordinate (m),
Zxx rho apparent resistivity (Ohm.m) for xx-element of impedance tensor,
xx phase phase (degrees) for xx-element of impedance tensor,
Zxy rho apparent resistivity (Ohm.m) for xy-element of impedance tensor,
.
.
.
Zyy phase Phase (degrees) for yy-element of impedance tensor.

3.3 MTZ3Dinv

3.3.1 `MT3Dinv.inp` input file format

The operation of MTZ3Dinv is controlled by a series of parameters that are provided to the program via an input file that must be called `mt3dinv.inp`. In order to understand what MTZ3Dinv is doing and how results are
generated it is necessary to understand the meaning of each parameter. When running MTZ3Dinv using the
Windows - hosted GUI this input file is generated automatically. Note that file names and paths should not contain spaces.

The input file is a text file with either 11 or 16 lines containing parameters as follows (the "|" symbol means "or"). Each line is described in detail below.

```
1)  mesh file
2)  background conductivity    file | value
3)  data file
4)  initial model file | value
5)  reference model file | value
6)  TOPO_CONST value | TOPO_FILE file | MNZ active zone file
7)  BOUNDS_NONE | BOUNDS_CONST bl bu | BOUNDS_FILE bounds file
8)  NONE | weight file
9)  DEFAULT | beta_start   beta_end   beta_factor
10) alpha_s   alpha_x   alpha_y   alpha_z
11) CHANGE_MREF | NOT_CHANGE_MREF
12) SMOOTH_MOD |SMOOTH_MOD_DIFF
13) USE_LOG_COND | USE_RES
14) DEFAULT | chifact
15) 0 | methpar
16) tol_nl   mindm   nit
17) intol   max_linit
18) fortol   initol
19) max_it_bicg   droptol   droptol_WTW
```

Line 1: Mesh file
See section 2.2.1; File: mesh.txt.

Line 2: Background conductivity

A single value, or the name of a file containing the background conductivity for all cells in the model. (The background conductivity is the model for which the primary fields are computed during the set-up stage of MT3Dinv.)

This model file is in the same format as the final conductivity model file (model.con) and it must be compatible with the mesh file on Line 1. The file is stored as one column of numbers, one for each cell. The first number is the value of the top south-western cell. The values are ordered such that Z (depth) changes the fastest, followed by X(easting), followed by Y(northing).

For uniform initial conductivity, a single value in units of S/m can be entered. All active cells are set to this

value. The inactive cells will be set to 1.0×10^{-8} S/m. (See also topography, Line 6.)

Line 3: Data file

See section 2.2.2; File: data.txt.

Line 4: Initial conductivity

See paragraphs 2 and 3 under Line 2: Background conductivity.

Line 5: Reference conductivity

As per line 4. This is a value (units of S/m), or the name of a file containing the reference conductivity, for all cells in the model. The reference conductivity is also used as the background conductivity in the primary- secondary separation that is used in the forward modelling within MTZ3Dinv when operated with the GUI.

Line 6 : Topography or active cells specification

When topography is flat, enter a value in metres. Everything above 'value' will be air (1.0×10^{-8} S/m) .

When topography is in a file, enter a file name. The file (described above in section 2.2.11) is in the same format used in other UBC-GIF codes. Air cells will be assigned a value of 1.0×10^{-8} S/m.

Alternatively, indicate which cells are active and which are inactive in the inversion using the MNZ (or the so- called "active zone file") option. This file contains only 1's and 0's, stored in the same format as the conductivity model. A value of 0 indicates that the cell is inactive and a value of 1 indicates that the cell is active, meaning the inversion will adjust the cell's conductivity as it searches for an optimal model.

When TOPO_CONST or TOPO_FILE is used, all inactive cells are set to 1.0×10^{-8}. When the MNZ is used, the model has inactive cells set to the value of the reference model.

Line 7: Bounds

A bounds file may be specified in line 7 of `MT3Dinv.inp`, with a particular conductivity model. The bounds are used to put constraints on the inversion. There are three options to set up the conductivity bounds

`BOUNDS_NONE`	(No bounds are imposed on the conductivity values)
`BOUNDS_CONST bl bu`	(Same bounds for all cells: bl – lower bound, bu – upper bound)
`BOUNDS_FILE`	(A bounds file to be used)

The bounds file has the same number of lines as the model file. It consists of 2 columns, containing the lower bounds in the first column and the upper bounds in the second column.

Line 8: Weighting

Enter NULL to have no weighting or enter a file stored in the same format as conductivity which contains the weighting for each cell. Weighting for inactive cells will be ignored. (See the weighting file description in section 2.2.3 above.)

Line 9: DEFAULT | beta_start beta_end beta_factor

This line specifies how the tradeoff parameter (Beta) is to be "cooled" or adjusted during the inversion. Enter DEFAULT to tell the inversion to work out it's own sequence of Betas. Default tradeoff parameters are calculated according to:

```
beta_start = ||Jr||**2 / ||Wr||**2 *
1000 beta_end = beta_start / 1.e+7
```

where `r` is a random vector.

```
beta_factor = 0.16681
```

It is recommended that initial inversions employ default values for Beta. For subsequent inversions, the user could then specify alternative Beta values based upon those found initially by default.

Line 10: Smoothing

These are the model objective function parameters α_S, α_X, α_Y, and α_Z. Guidelines describing the values and effects of these three parameters are given in the UBC-GIF tutorial on inversion of 2D DC resistivity and IP data. The comments on that page are also relevant for MT3Dinv.

Line 11: Changing the reference model

`CHANGE_MREF` the reference model will be changed to the recovered model after each beta iteration.
`NOT_CHANGE_MREF` the reference model will never be changed.

Line 12: Model norm calculation selection

`SMOOTH_MOD` model norm does not contain the reference model.
`SMOOTH_MOD_DIF` model norm contains the reference model.

Line 13: Type of model to use in the inversion.

`USE_LOG_COND` – specifies log conductivity

NOTE: this is the only option for MT3Dinv. Linear values can not be used.

Line 14: Inversion parameters, or chifact

Enter DEFAULT on line 14 to use the following parameters (defined below in Lines 15 - 19)

parameter	default value	corresponding input file line
chifact	1	14
methpar	0	15

tol_nl	0.01	16
mindm	0.001	16
nit	5	16
intol	0.01	17
max_linit	10	17
fortol	1.0×10^{-9}	18
inintol	1.0×10^{-8}	18
max_it_bicg	15	19
droptol	0.01	19
droptol_ W^TW	0.01	19

Otherwise, enter specific values for these parameters on Lines 14 - 19. These lines are not needed if "DEFAULT" is entered on Line 14.

If not using "DEFAULT", enter chifact such that target misfit is reached when:

$W_d|d^{obs}-d^{pred}|^2 <= chifact * N$

Line 15: methpar

This line is reserved for future developments. It must be set to 0. (Default = 0)

Line 16: tol_nl mindm nit

NOTE: Separate parameters with spaces.

tol_nl	Stop tolerance for the nonlinear solver (uc_inv) which finds the optimal model at a given (fixed) tradeoff parameters (Beta). It is typically 0.01 or smaller. Numerically, the model is considered optimal when it's gradient is close to zero; i.e.when $\|gc\|/gc_0$ < tol_nl, when gc_0 = initial $\|gc\|$.	Default = 0.01
mindm	Smallest model perturbation step size (*m). One of the program's quitting criteria is when the model perturbation max(|*m|)< mindm.	Default = 0.001
nit	Maximum number of iterations when adjusting the step size that updates the model at each value of Beta.	Default = 5

Line 14: intol max_linit

intol	The tolerance for the linear solver (ipcg) which finds the optimal model perturbation size. Typically 0.1 to 0.01.	Default = 0.01
max_linit	Maximum number of iterations for this linear solver (ipcg).	Default = 10

Line 15: fortol inintol

fortol	Stop tolerance for forward and adjoint calculations when evaluating the objective function and gradients. This should be small	Default = 1.0×10^{-9}

inintol	Stop tolerance for the forward and adjoint calculations inside the linear solver (ipcg). This tolerance can be larger than "fortol" to save time (typical 0.001 and lower).	Default = 1.0×10^{-8}

Line 16: max_it_bicg droptol droptol_W^TW

max_it_bicg	Maximum number of iterations in BiCGSTAB when performing the forward and adjoint calculations.	Default = 15
droptol	Drop tolerance for the ilu preconditioner for the A matrix.	Default = 0.01
droptol_WTW	Drop tolerance for the ilu preconditioner for the W^TW matrix. This is used when the algorithm is looking for optimal model step size, and in the IPCG solver.	Default = 0.01

3.3.2 MTZ3Dinv outputs

The following files contain output information from the inversion. Files are described under section 2.2, General
Files for MTZ3Dinv Programs above. Links to details are included in the table.

File name	Descriptio
inv.con	The inversion result (a conductivity model).
inv_*j*.con	The conductivity model for the jth value of beta.
mt3dinv.out	A column-format list of iteration statistics summarizing the progress of the inversion algorithm.
bicg.txt	A summary of BICGStab calculations
dobs.txt	Observed data (read directly from the input data file) in the same format as predicted data.
dpred.txt	Predicted data generated by forward modelling on the most recent (or final) model recovered by the inversion
dpred_*j*.txt	Predicted data for the model produced for the jth value of beta.
mt3dinv.log	The inversion log file.

3.3.3. MTZ3Dinv log file description

See file mt3dinv-log.html.

3.4 Running ZTEM_MTZ3Dinv_64 under Message Pass Interface (MPI)

For optimal performance of the inversion (ZTEM_MTZ3Dinv_64) it is best to use the Message Passing Interface (MPI), which allows running multiple computational devices in parallel,

including commodity clusters, hi-speed networks and multi-core processors on local computers. In order to install the MPI application library, download it from http://www.mcs.anl.gov/research/projects/mpich2/

Linux:

For commodity clusters operated under Linux system, the code can be run on any number of processors listed in a description ASCII file. The following is an example of such description file:

```
Compname01:nProc
Compname02:nProc
Compname03:nProc
```

The description file name is completely arbitrary. "Compname" is the network name of the computer to be used and "nProc" is the number of processors to be employed for the procedure. If the computer is on the local network and can be directly accessed, then no path is needed to be specified. The following is an example of a command line to be used under Linux operating system in order to start the forward simulation:

```
mpiexec -machinefile machines.txt -n 20 ./ZTEM_MTZ3Dinv_64
```

In this command line "-machinefile" calls for a description file "machines.txt" and "-n 20" indicates a total amount of processors to be used on all the machines listed in the description file.

Windows:

For single multi-core computer usage, under Windows operating system, the command line will look like this:

```
"C:\Program Files\MPICH2\bin\mpiexec.exe" -localonly 4 ZTEM_MTZ3Dinv_64
```

In this line "-localonly" limits the computation to only one machine, from which the command line is launched and "4" specifies the number of processors to be used.
For running the code on several computers or a network:

- Make sure each computer has the same version of MPI installed.
- The user running the program should have the same "UserID" and "password" on each computer (the user does not have to be logged on to every computer, but has to have a network account set up on each computer with same identification).
- Make your *"workdir"* and *"execdir"* folders shareable, and make sure that *"workdir"* provides full sharing (read and write).
- Make sure your shared folders are visible on each computer then place the input files and executables in sharable folders.
- The firewall in each computer (except for the host) should be turned off, or else the h3dtd program should be added to the exceptions list.

The command line to start the MPI job is:

```
mpiexec.exe -machinefile machines.txt -n 8 -priority 1 -dir
\\MYCOMP\share \\MYCOMP\share\exe\ZTEM_MTZ3Dinv_64.exe
```

In the above example:

`machines.txt` - file containing the names of the computers to use. Each computer name can be followed be :p which indicates the number of processes to start on that machine.
`-n 8` - total number of processes to start.
`-priority 1` - (optional) indicates that all jobs should be started at low priority.
`-dir \\MYCOMP\share` - sharable folder that should be visible on all computers.
`\\MYCOMP\share\exe\ZTEM_MTZ3Dinv_64.exe` - full path of the executable. Other computers must be able to see it.

For convenience, it is recommended to set up batch files (*.bat) for running `ZTEM_MTZ3Dinv_64.exe` on a local network. Two examples of such batch files are provided with the documentation. The file *"run_ZTEM_MTZ3Dinv_64_local_mpi.bat"* is to be used for running `ZTEM_MTZ3Dinv_64.exe` on local workstation alone and the file *"run_ZTEM_MTZ3Dinv_64_mpi.bat*" is to be used for running the code on the local network.

In editing the file: *"run_ZTEM_MTZ3Dinv_64_local_mpi.bat"* the editable text includes the path to the ZTEM_MTZ3Dinv_64.exe file, which should be set to your "execdir" location. The editable parts of the *"run_ZTEM_MTZ3Dinv_64_mpi.bat"* include the following:

- machinefile.txt
- number of processors
- path to "workdir"
- path to "execdir"

The file *"machinefile.txt"* used for executing the code under local network has the same format as the example shown above for Linux environment and should be located in the *"workdir"*. Please note that for running the code on multiple network computers, the total number of processors should be specified, equal to the sum of all processors on all computers listed in *machinefile.txt*.