

---

# **em1dtm Documentation**

*Release 5.0*

**UBC-GIF**

**Aug 11, 2018**



---

# Contents

---

<b>1</b>	<b>EM1DTM package overview</b>	<b>3</b>
1.1	Description . . . . .	3
1.2	Program library content . . . . .	4
1.3	Licensing . . . . .	4
1.4	Installing . . . . .	4
<b>2</b>	<b>Background theory</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	Forward Modeling . . . . .	9
2.3	Inversion Methodologies . . . . .	21
<b>3</b>	<b>Elements of the program</b>	<b>29</b>
3.1	EM1DFM Main Input File . . . . .	29
3.2	EM1DFM Supporting Files . . . . .	34
3.3	EM1DFM Output Files . . . . .	41
3.4	EM1DFMFWD Program . . . . .	49
<b>4</b>	<b>Running the programs</b>	<b>51</b>
4.1	Command Prompt/Terminal Syntax . . . . .	51
4.2	Command Prompt/Terminal Outputs . . . . .	52
<b>5</b>	<b>Examples</b>	<b>53</b>
5.1	Example 1: Single Sounding . . . . .	53
5.2	Example 2: Multiple Soundings . . . . .	53
<b>6</b>	<b>References</b>	<b>57</b>



EM1DTM is a program library for carrying out forward modeling and inversion of time-domain electromagnetic data for a 1d layered Earth. The contents of this manual are as follows:



## EM1DTM package overview

## 1.1 Description

EM1DTM is a Fortran-based, multi-platform program library for carrying out the one-dimensional inversion of time-domain, small loop, electromagnetic data acquired to determine the spatial variation of the electrical conductivity and/or magnetic susceptibility of the subsurface. The name “EM1DTM” derives from: electromagnetics (“EM”), one-dimensional models (“1D”), time-domain observations (“T”), and magnetic (dipole) sources and receivers (“M”).

The observations are values of voltage (i.e., dB/dt) or magnetic eld. Receiver coils can be oriented in the x-, y- or z-directions, and they can be at any position relative to the transmitter loop. The transmitter loop can have any number of sides (greater than 2), and can be at any height above the ground surface. It is assumed to be horizontal. The transmitter current waveform can be a step o, a linear ramp turn-o, or a general waveform that is provided in discretized form. Observations can be for any time after the step or ramp turn o, or any time after or during a discretized waveform. All the observations (in any combination) that are provided for a particular transmitter loop constitute a “sounding”, and are used to construct the one- dimensional model for that sounding. Measurement uncertainties can be in the same units as the observations or as relative uncertainties in percent.

The product: an electrical conductivity model. The Earth models are composed of layers of uniform conductivity with xed interface depths. The value of the conductivity in each layer is sought by the inversion. Multiple soundings can be handled in a single run of the program. Each sounding is inverted independently for a one-dimensional model under the sounding location, with the sequence of one-dimensional models written out. These can be viewed directly as a composite two-dimensional image using the graphical user interface, or converted to a format which is suitable for viewing as a three- dimensional image using MeshTools3D. For any sounding location, the mathematical representation that EM1DTM uses to model the Earth varies only with depth. In particular, the representation comprises

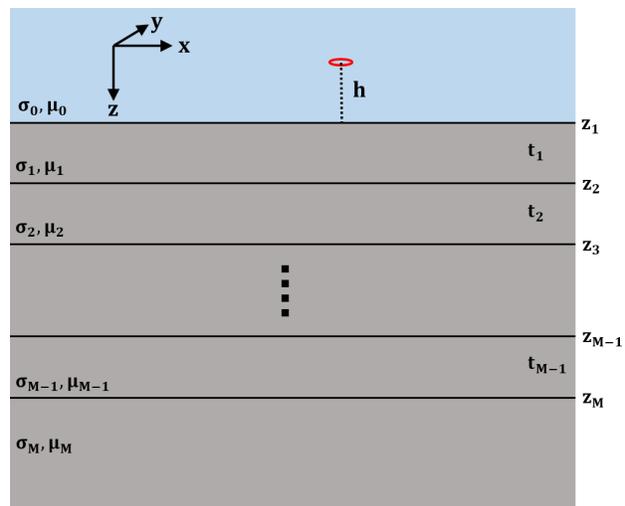


Fig. 1.1: Layered 1D model describing the Earth for each sounding. 3

many uniform “infinite” horizontal layers. For a complete mathematical treatment of the forward and inversion problem, see *Background Theory*.

General measures for both the measure of data mist and the measure of the amount of model structure:

- Huber M-measure for data mist, and
- Ekblom p-measure for model structure, allow for a whole suite of variations, from the traditional sum-of-squares measures, to more robust measures which can ignore outliers in the observations and which can generate piecewise-constant models.

The initial research underlying this program library was funded principally by the “IMAGE” consortium, of which the following companies were participants: AGIP, Anglo American, Billiton, Cominco, Falconbridge, INCO, MIM, Muskox Minerals, Newmont, Placer Dome and Rio Tinto, and from the Natural Sciences and Engineering Research Council of Canada (NSERC).

## 1.2 Program library content

This package consists of two programs:

- **EM1DTM**: carries out the inversion of small loop, time-domain EM data assuming a layered Earth model
- **EM1DTMFWD**: a stand-alone program for forward modeling the time-domain response assuming a layered Earth model

## 1.3 Licensing

A **constrained educational version** of the program is available with the **IAG** package (please visit [UBC-GIF website](#) for details). The educational version is fully functional so that users can learn how to carry out effective and efficient 3D inversions of magnetic data. **However, RESEARCH OR COMMERCIAL USE IS NOT POSSIBLE because the educational version only allows a limited number of data and model cells.**

Licensing for an unconstrained academic version is available - see the [Licensing policy document](#).

**NOTE:** All academic licenses will be **time-limited to one year**. You can re-apply after that time. This ensures that everyone is using the most recent versions of codes.

Licensing for commercial use is managed by third party distributors. Details are in the [Licensing policy document](#).

## 1.4 Installing

There is no automatic installer currently available for the . Please follow the following steps in order to use the software:

1. Extract all files provided from the given zip-based archive and place them all together in a new folder such as
2. Add this directory as new path to your environment variables.

Two additional notes about installation:

- Do not store anything in the “bin” directory other than executable applications and Graphical User Interface applications (GUIs).
- A Message Pass Interface (MPI) version is available for Linux upon and the installation instructions will accompany the code.



### 2.1 Introduction

The EM1DTM and EM1DTMFWD programs are designed to interpret time-domain, small loop, electromagnetic data over a 1D layered Earth. These programs model the Earth's time-domain electromagnetic response due to a small inductive loop source which carries a sinusoidal time-varying current. The data are the secondary magnetic field which results from currents and magnetization induced in the Earth.

#### 2.1.1 Details regarding the source and receiver

EM1DTM and EM1DTMFWD assume that the transmitter loop is horizontal. The programs also assume that each receiver loop is sufficiently small and that they may be considered point receivers; i.e. the spatial variation in magnetic flux through the receiver loop is negligible.

#### 2.1.2 Details regarding the domain

EM1DTM and EM1DTMFWD model the Earth's response for measurements above a stack of uniform, horizontal layers. **The coordinate system used for describing the Earth models has  $z$  as positive downwards, with the surface of the Earth at  $z=0$ .** The  $z$ -coordinates of the source and receiver locations (which must be above the surface) are therefore negative. Each layer is defined by a thickness and an electrical conductivity. It is the physical properties of the layers that are obtained during the inversion of all measurements gathered at a single location, with the depths to the layer interfaces remaining fixed. If measurements made at multiple locations are being interpreted, the corresponding one-dimensional models are juxtaposed to create a two-dimensional image of the subsurface.

All measurements that are to be inverted for a single one-dimensional model must be grouped together as a "sounding". A sounding can be considered a distinct collection of TEM measurements (receivers and times) for a given transmitter. Each different sounding is inverted for a separate one-dimensional model.

The electrical conductivity of Earth materials varies over several orders of magnitude, making it more natural to invert for the logarithms of the layer conductivities rather than the conductivities themselves. This also ensures that conductivities in the constructed model are positive.

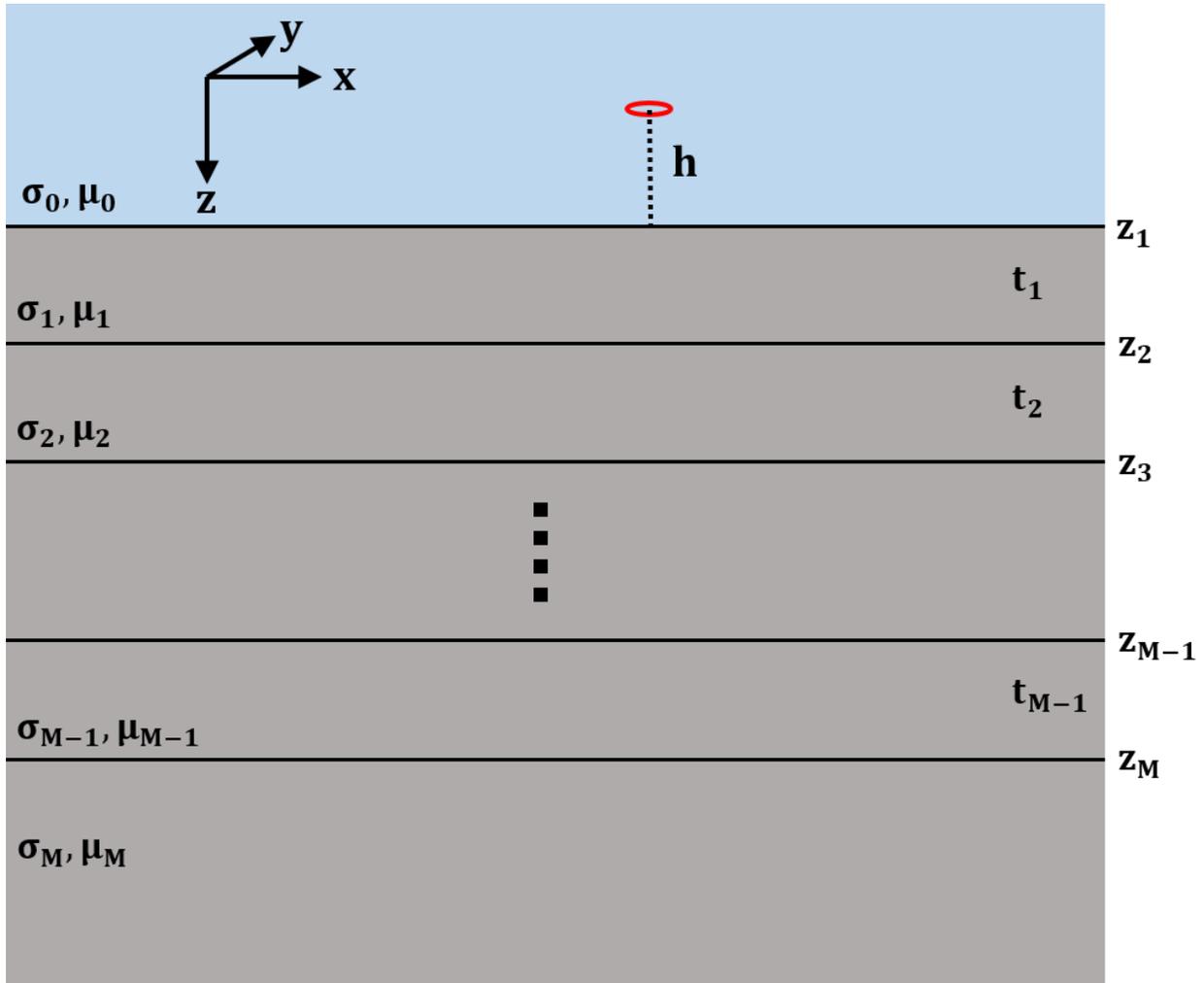


Fig. 2.1: Layered 1D model describing the Earth for each sounding.



Fig. 2.2: Two distinct groupings of transmitters and receivers (soundings) at the same location (left). Different soundings used to map lateral variation in the Earth (right). [Click to enlarge](#).

### 2.1.3 Details regarding the data

For a horizontal loop source, EM1DTM and EM1DTMFWD can handle any combination of:

- times for the transmitter current
- separations and heights for the transmitter and receiver(s)
- B-field or time varying (dBdt) response
- x, y and z-components of the response

Programs EM1DTM and EM1DTMFWD accept observations in two different forms: values of the secondary magnetic field H-field in A/m and values of the total H-field in A/m. If the transmitter and receiver have the same orientation, the x-, y- and z-components of the secondary field are normalized by the x-, y- and z-components of the free-space field respectively. If the transmitter and receiver have different orientations, the secondary field is normalized by the magnitude of the free-space field.

## 2.2 Forward Modeling

The method used to compute the magnetic field values for a particular source- receiver arrangement over a layered Earth model is the matrix propagation approach described in Farquharson ([Farquharson2003]). The method uses the z-component of the Schelkunoff F-potential ([Ward1987]): layered Earth model is the matrix propagation approach described in Farquharson & Oldenburg (1996) and Farquharson, Oldenburg & Routh (2003). Computations are done in the frequency domain, then the elds transformed to the time domain.

This section covers the aspects of the forward-modelling procedure for the three components of the magnetic field above a horizontally-layered Earth model for a horizontal, many-sided transmitter loop also above the surface of the model. The field for the transmitter loop is computed by the superposition of the fields due to horizontal electric dipoles (see eqs.~4.134–4.152 [Ward1987]). Because the loop is closed, the contributions from the ends of the electric dipole are ignored, and the superposition is carried out only over the **TE**-mode component. This TE-mode only involves the z-component of the Schelkunoff **F**-potential, just as for program **EM1DFM**.

The Schelkuno F-potential is dened as follows:

$$\mathbf{E} = -\nabla \times \mathbf{F}$$

$$\mathbf{H} = -\sigma \mathbf{F} + \frac{1}{i\omega\mu} \nabla(\nabla \cdot \mathbf{F})$$

where **E** and **H** are the electric and mangetic fields,  $\sigma$  and  $\mu$  are the conductivity and permability of the uniform region to which the above queation refer, and a time-dependence of  $e^{i\omega t}$  has been assumed.

In the  $j^{th}$  layer ( $j > 0$ ), with conductivity  $\sigma_j$ , and permeability  $\mu_j$ , the z-component of the Schelkuno potential satisses the equation (assuming the quasi-static approximation, and the permeability of the layer is equal to that of free space  $\mu_0$ ):

$$\nabla^2 F_j - i\omega\mu_0\sigma_j F_j = 0$$

The propagation of  $F$  through the stack of layers therefore happens in exactly the same way, and so is not repeated here (see **EM1DFM Forward modeling** section).

---

#### Note:

#### Assumptions

- $e^{i\omega t}$  time-dependence (just as in [Ward1987])
- quasi-static approximation throughout

- $z$  positive downwards
- air halfspace ( $\sigma = 0$ ) for  $z < 0$ , piecewise constant model ( $\sigma > 0$ ) of  $N$  layers for  $z \geq 0$ ,  $N^{th}$  layer being the basement (i.e. ~homogeneous) halfspace
- magnetic permeability everywhere equal to that of free space.

From the propagation of  $F$  through the layers gives the following expression for the kernel of the Hankel transform,  $\tilde{F}$ , in the air halfspace ( $z < 0$ ):

$$\tilde{F}_0 = D_0^S \left( e^{-u_0 z} + \frac{P_{21}}{P_{11}} e^{u_0 z} \right) \quad (2.1)$$

(same as [here in EM1DFM](#)).

For a horizontal  $x$ -directed electric dipole at a height  $h$  (i.e.,  $z = -h$ ,  $h > 0$ ) above the surface of the layered Earth, the downward-decaying part of the primary solution of  $\tilde{F}$  (and the only downward-decaying part of the solution in the air halfspace) at the surface of the Earth ( $z = 0$ ) is given by

$$D_0^S = -\frac{i\omega\mu_0}{2u_0} \frac{ik_y}{k_x^2 + k_y^2} e^{-u_0 h} \quad (2.2)$$

from [Ward1987], equation (4.137). Substituting (2.2) into (2.1) gives

$$\tilde{F}_0 = -\frac{i\omega\mu_0}{2u_0} \frac{ik_y}{k_x^2 + k_y^2} \left( e^{-u_0(z+h)} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right). \quad (2.3)$$

Generalizing this expression for  $z$  above ( $z < -h$ ) as well as below the source ( $z > -h$ ):

$$\tilde{F}_0 = -\frac{i\omega\mu_0}{2u_0} \frac{ik_y}{k_x^2 + k_y^2} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right). \quad (2.4)$$

Applying the inverse Fourier transform to (2.4) gives

$$F_0(x, y, z, \omega) = -\frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{i\omega\mu_0}{2u_0} \frac{ik_y}{k_x^2 + k_y^2} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right) e^{i(k_x x + k_y y)} dk_x dk_y \quad (2.5)$$

(equation (4.139) of [Ward1987]). Using the identity

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{F}(k_x^2 + k_y^2) dk_x dk_y = 2\pi \int_0^{\infty} \tilde{F}(\lambda) \lambda J_0(\lambda r) d\lambda, \quad (2.6)$$

([Ward1987], equation (2.10)) where  $\lambda^2 = k_x^2 + k_y^2$  and  $r^2 = x^2 + y^2$ , (2.5) can be rewritten as

$$F_0(x, y, z, \omega) = -\frac{1}{2\pi} \frac{\partial}{\partial y} \int_0^{\infty} \frac{i\omega\mu_0}{2u_0} \frac{1}{\lambda^2} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right) \lambda J_0(\lambda r) d\lambda, \quad (2.7)$$

$$= -\frac{i\omega\mu_0}{4\pi} \frac{\partial}{\partial y} \int_0^{\infty} \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \frac{1}{\lambda^2} J_0(\lambda r) d\lambda, \quad (2.8)$$

$$= \frac{i\omega\mu_0}{4\pi} \frac{y}{r} \int_0^{\infty} \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \frac{1}{\lambda} J_1(\lambda r) d\lambda \quad (2.9)$$

since

$$\frac{\partial J_0(\lambda r)}{\partial y} = -\lambda \frac{y}{r} J_1(\lambda r) \quad (2.10)$$

([Ward1987], equation 4.44 (almost)).

The  $H$ -field in the air halfspace can be obtained from (2.9) (or (2.8)) by using equation (1.130) of [Ward1987]:

$$H_x = \frac{1}{i\omega\mu_0} \frac{\partial^2 F_0}{\partial x \partial z}, \quad (2.11)$$

$$H_y = \frac{1}{i\omega\mu_0} \frac{\partial^2 F_0}{\partial y \partial z}, \quad (2.12)$$

$$H_z = \frac{1}{i\omega\mu_0} \left( \frac{\partial^2}{\partial z^2} + \kappa_0^2 \right) F_0 \quad (2.13)$$

$$= \frac{1}{i\omega\mu_0} \frac{\partial^2 F_0}{\partial z^2}. \quad (2.14)$$

since  $\kappa_0^2 = 0$ . Applying (2.11) to (2.9) gives

$$H_x(x, y, z, \omega) = \frac{1}{4\pi} \frac{\partial}{\partial x} \frac{y}{r} \int_0^\infty \left( \pm e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) J_1(\lambda r) d\lambda. \quad (2.15)$$

(The plus/minus is to do with whether or not the observation location is above or below the source. In the program, it perhaps it is only the secondary fields that are computed using the above expressions: the primary field, which corresponds to the first term in each Hankel transform kernel above is computed using its for in  $(x, y, z)$ -space.) When the above expression for a horizontal electric dipole is integrated along a wire all that is left is the effects of the endpoints. These will cancel when integrating around the closed loop. So as far as the part of  $H_x$  that contributes to the file due to a closed loop:

$$H_x(x, y, z, \omega) = 0. \quad (2.16)$$

For the  $y$ -component of the H-field, first consider differentiating the expression for  $F_0$  in (2.5) with respect to  $y$ :

$$\frac{\partial F_0}{\partial y} = -\frac{1}{4\pi^2} \frac{\partial}{\partial y} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{i\omega\mu_0}{2u_0} \frac{ik_y}{k_x^2 + k_y^2} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right) e^{i(k_x x + k_y y)} dk_x dk_y, \quad (2.17)$$

$$= \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{i\omega\mu_0}{2u_0} \frac{k_y^2}{k_x^2 + k_y^2} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right) e^{i(k_x x + k_y y)} dk_x dk_y, \quad (2.18)$$

$$= \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{i\omega\mu_0}{2u_0} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right) e^{i(k_x x + k_y y)} dk_x dk_y$$

$$- \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{i\omega\mu_0}{2u_0} \frac{k_x^2}{k_x^2 + k_y^2} \left( e^{-u_0|z+h|} + \frac{P_{21}}{P_{11}} e^{u_0(z-h)} \right) e^{i(k_x x + k_y y)} dk_x dk_y, \quad (2.19)$$

since

$$\frac{k_y^2}{k_x^2 + k_y^2} = 1 - \frac{k_x^2}{k_x^2 + k_y^2}. \quad (2.20)$$

Converting the  $k_x^2$  into derivatives with respect to  $x$ , and converting the two-dimensional Fourier transforms to Hankel transforms gives

$$\frac{\partial F_0}{\partial y} = \frac{i\omega\mu_0}{4\pi} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) J_0(\lambda r) d\lambda$$

$$+ \frac{i\omega\mu_0}{4\pi} \frac{\partial^2}{\partial x^2} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \frac{1}{\lambda^2} J_0(\lambda r) d\lambda, \quad (2.21)$$

$$= \frac{i\omega\mu_0}{4\pi} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) J_0(\lambda r) d\lambda$$

$$- \frac{i\omega\mu_0}{4\pi} \frac{\partial}{\partial x} \frac{x}{r} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \frac{1}{\lambda} J_1(\lambda r) d\lambda, \quad (2.22)$$

using equations (4.144) and (4.117) of [Ward1987]. Differentiating (2.22) with respect to  $z$  and scaling by  $i\omega\mu_0$  (see (2.12)) gives

$$\begin{aligned} H_y(x, y, z, \omega) &= \frac{1}{4\pi} \int_0^\infty \left( \pm e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \lambda J_0(\lambda r) d\lambda \\ &\quad - \frac{1}{4\pi} \frac{\partial}{\partial x} \frac{x}{r} \int_0^\infty \left( \pm e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) J_1(\lambda r) d\lambda \end{aligned} \quad (2.23)$$

(equation 4.150 of [Ward1987]). The second integral in the above expression only contributes at the ends of the dipole. So the only part of  $H_y$  required to compute the field due to the closed loop is

$$H_y(x, y, z, \omega) = \frac{1}{4\pi} \int_0^\infty \left( \pm e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \lambda J_0(\lambda r) d\lambda. \quad (2.24)$$

Finally, applying (2.14) to (2.9) gives the  $z$ -component of the H-field:

$$H_z(x, y, z, \omega) = \frac{1}{4\pi} \frac{y}{r} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \lambda J_1(\lambda r) d\lambda \quad (2.25)$$

(equation (4.152) of [Ward1987]).

(2.24) and (2.25) are for the total H-field ( $H_x = 0$  from (2.16) for an  $x$ -directed electric dipole excluding the effects at the end-points, that is, the wholespace field up in the air plus the field due to currents induced in the layered Earth. In (2.24) and (2.25), the first part of the kernel of the Hankel transform corresponds to the primary wholespace field:

$$H_y(x, y, z, \omega) = \frac{1}{4\pi} \int_0^\infty \pm e^{-\lambda|z+h|} \lambda J_0(\lambda r) d\lambda, \quad (2.26)$$

$$= \frac{1}{4\pi} \frac{\partial}{\partial z} \int_0^\infty e^{-\lambda|z+h|} J_0(\lambda r) d\lambda, \quad (2.27)$$

and

$$H_z(x, y, z, \omega) = \frac{1}{4\pi} \frac{y}{r} \int_0^\infty e^{-\lambda|z+h|} \lambda J_1(\lambda r) d\lambda \quad (2.28)$$

$$= -\frac{1}{4\pi} \frac{y}{r} \frac{\partial}{\partial r} \int_0^\infty e^{-\lambda|z+h|} J_0(\lambda r) d\lambda. \quad (2.29)$$

From [Ward1987] equation (4.53), the integral in the above two expressions can be done:

$$\int_0^\infty e^{-\lambda|z+h|} J_0(\lambda r) d\lambda = \frac{1}{(r^2 + (z+h)^2)^{1/2}}. \quad (2.30)$$

So

$$H_y(x, y, z, \omega) = \frac{1}{4\pi} \frac{\partial}{\partial z} \frac{1}{(r^2 + (z+h)^2)^{1/2}}, \quad (2.31)$$

$$= -\frac{1}{4\pi} \frac{z}{(r^2 + (z+h)^2)^{3/2}} \quad (2.32)$$

(equation (2.42) of [Ward1987] for  $\sigma = 0$ ), and

$$H_z(x, y, z, \omega) = -\frac{1}{4\pi} \frac{y}{r} \frac{\partial}{\partial r} \frac{1}{(r^2 + (z+h)^2)^{1/2}}, \quad (2.33)$$

$$= \frac{1}{4\pi} \frac{y}{r} \frac{r}{(r^2 + (z+h)^2)^{3/2}}, \quad (2.34)$$

$$= \frac{1}{4\pi} \frac{y}{(r^2 + (z+h)^2)^{3/2}} \quad (2.35)$$

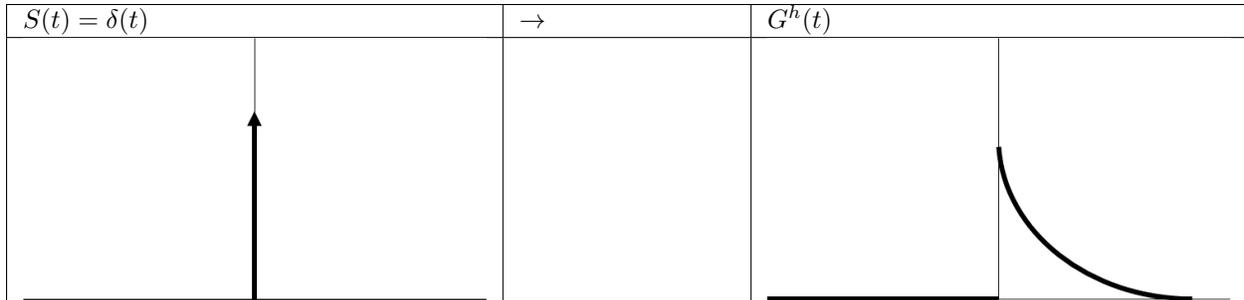
(equation (2.42) of [Ward1987] for  $\sigma = 0$ ).

### 2.2.1 Frequency- to time-domain transformation – Part I

The solution for the H-field in the frequency domain for a delta-function source in time (and hence a flat, constant, real source term in the frequency domain) is, for example,

$$H_z(x, y, z, \omega) = \frac{1}{4\pi} \frac{y}{r} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{P_{21}}{P_{11}} e^{\lambda(z-h)} \right) \lambda J_1(\lambda r) d\lambda.$$

Doing the inverse Fourier transform of these kinds of expressions does not encounter any subtleties, and gives an H-field as a function of time that, schematically, looks like:



This is the basic *response* that program EM1DTM computes. Notation of  $G^h(t)$  because this is the Green's function for convolution with the transmitter current waveform  $S(t)$  to give the H-field:

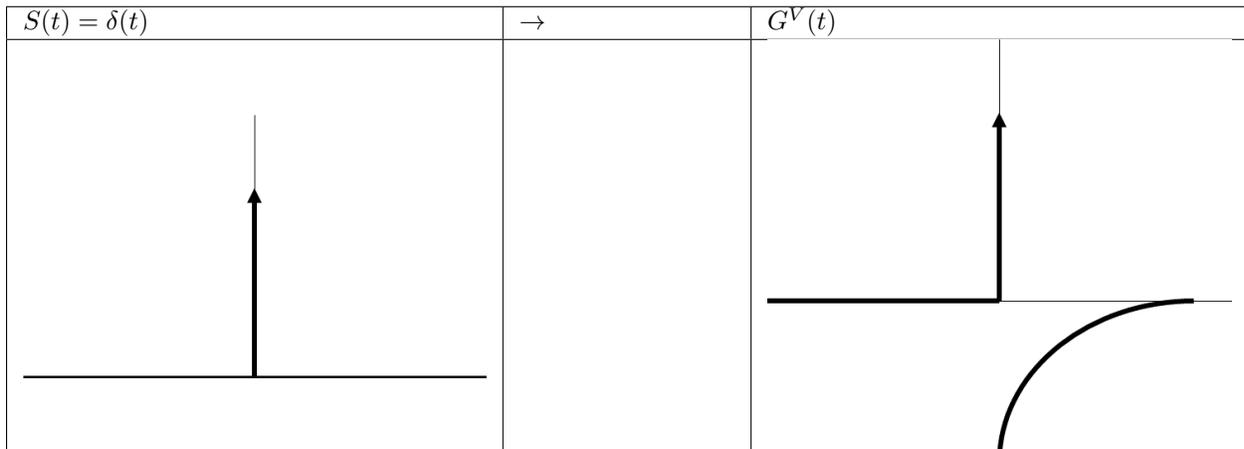
$$h(t) = \int_{t'=-\infty}^\infty G^h(t-t') S(t') dt'. \tag{2.36}$$

The H-field for the delta-function source, that is,  $G^h$  certainly exists for  $t > 0$ . Also, it is certainly zero for  $t < 0$ . And at  $t = 0$ , it certainly is not infinite (not physical). Let's re-describe the function  $G^h$  (shown in the diagram above) as

$$G^h(t) = X(t) \tilde{G}^h(t), \tag{2.37}$$

where  $\tilde{G}^h(t)$  is equal to  $G^h$  for  $t > 0$ ,  $\tilde{G}^h(0) = \lim_{t \rightarrow 0^+} G^h$ , and does anything it wants for  $t < 0$ . And  $X(t)$  is the Heaviside function. This moves all issues about what is happening at  $t = 0$  into the Heaviside function.

For measurements of voltage, the Green's function (*impulse response*) that is required is the time derivative of  $G^h$  (and for all  $t$ , not just  $t > 0$ ). Schematically:



In terms of math:

$$V(t) = \int_{t'=-\infty}^\infty G^V(t-t') S(t') dt'. \tag{2.38}$$

Let's take the time derivative of (2.37) to get the full expression for  $G^V$ :

$$\begin{aligned}
 G^V(t) &= \frac{dG^h}{dt}, \\
 &= \frac{d}{dt}(X \tilde{G}^h) \\
 &= X \frac{d\tilde{G}^h}{dt} + \delta \tilde{G}^h,
 \end{aligned} \tag{2.39}$$

where  $\delta$  is the delta function. Now, this is not a time derivative that should be happening numerically. So, given the basic  $G^h(t)$  and some representation of the transmitter current waveform  $S(t)$ , program EM1DTM currently uses the re-arrangement of (2.38) given by the substitution of (2.39) into (2.38) followed by some integration by parts:

$$\begin{aligned}
 V(t) &= \int_{t'=-\infty}^{\infty} \left\{ X(t-t') \frac{d\tilde{G}^h}{dt'}(t-t') + \delta(t-t') \tilde{G}^h(t-t') \right\} S(t') dt', \\
 &= \tilde{G}^h(0) S(t) + \int_{t'=-\infty}^t \frac{d\tilde{G}^h}{dt'}(t-t') S(t') dt',
 \end{aligned} \tag{2.40}$$

where the Heaviside function has been used to restrict the limits of the integration. Now doing the integration by parts:

$$\begin{aligned}
 V(t) &= \tilde{G}^h(0) S(t) + \left[ \tilde{G}^h(t-t') S(t') \right]_{t'=-\infty}^t - \int_{t'=-\infty}^t \tilde{G}^h(t-t') \frac{dS}{dt'}(t') dt' \\
 &= \tilde{G}^h(0) S(t) + \tilde{G}^h(0) S(t) - \int_{t'=-\infty}^t \tilde{G}^h(t-t') \frac{dS}{dt'}(t') dt'.
 \end{aligned} \tag{2.41}$$

Which looks as though it has the *expected* additional non-convolution-integral term.

However, perhaps there should be an additional minus sign in going from AA--4 to the one before (2.40) because the derivative has changed from  $d/dt$  to  $d/dt'$ . But perhaps not.

## 2.2.2 Frequency- to time-domain transformation

The Fourier transform that was applied to Maxwell's equations to get the frequency-domain equations was (see [Ward1987], equation (1.1))

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt,$$

and the corresponding inverse transform is

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega.$$

For the frequency domain computations, it is assumed that the source term is the same for all frequencies. In other words, a flat spectrum, which corresponds to a delta-function time-dependence of the source.

Consider at the moment a causal signal, that is, one for which  $f(t) = 0$  for  $t < 0$ . The Fourier transform of this signal is then

$$\begin{aligned}
 F(\omega) &= \int_0^{\infty} f(t) e^{-i\omega t} dt \\
 &= \int_0^{\infty} f(t) \cos \omega t dt - i \int_0^{\infty} f(t) \sin \omega t dt.
 \end{aligned}$$

Note that because of the dependence of the real part of  $F(\omega)$  on  $\cos \omega t$  and of the imaginary part on  $\sin \omega t$ , the real part of  $F(\omega)$  is even and the imaginary part of  $F(\omega)$  is odd. Hence,  $f(t)$  can be obtained from either the real or

imaginary part of its Fourier transform via the inverse cosine or sine transform:

$$f(t) = \frac{2}{\pi} \int_0^{\infty} \operatorname{Re} F(\omega) \cos \omega t \, d\omega, \quad \text{or}$$

$$f(t) = -\frac{2}{\pi} \int_0^{\infty} \operatorname{Im} F(\omega) \sin \omega t \, d\omega$$

(For factor of  $2/\pi$  see, for example, Arfken.)

Now consider that we've computed the H-field in the frequency domain for a uniform source spectrum. Then from the above expressions, the time-domain H-field for a  $t^{\text{th}}$  positive delta-function} source time-dependence is

$$h_{\delta+}(t) = \frac{2}{\pi} \int_0^{\infty} \operatorname{Re} H(\omega) \cos \omega t \, d\omega, \quad \text{or}$$

$$h_{\delta+}(t) = -\frac{2}{\pi} \int_0^{\infty} \operatorname{Im} H(\omega) \sin \omega t \, d\omega$$

where  $H(\omega)$  is the frequency-domain H-field for the uniform source spectrum. For a  $t^{\text{th}}$  negative delta-function} source:

$$h_{\delta-}(t) = -\frac{2}{\pi} \int_0^{\infty} \operatorname{Re} H(\omega) \cos \omega t \, d\omega, \quad \text{or}$$

$$h_{\delta-}(t) = \frac{2}{\pi} \int_0^{\infty} \operatorname{Im} H(\omega) \sin \omega t \, d\omega.$$

The negative delta-function source dependence is the derivative with respect to time of a step turn-off source dependence. Hence, the  $t^{\text{th}}$  derivative} of the time-domain H-field due to a  $t^{\text{th}}$  step turn-off} is also given by the above expressions:

$$\frac{\partial h_s}{\partial t}(t) = -\frac{2}{\pi} \int_0^{\infty} \operatorname{Re} H(\omega) \cos \omega t \, d\omega, \quad \text{or}$$

$$\frac{\partial h_s}{\partial t}(t) = \frac{2}{\pi} \int_0^{\infty} \operatorname{Im} H(\omega) \sin \omega t \, d\omega.$$

Integrating the above two expressions gives the H-field for a  $t^{\text{th}}$  step turn-off} source:

$$h_s(t) = h(0) - \frac{2}{\pi} \int_0^{\infty} \operatorname{Re} H(\omega) \frac{1}{\omega} \sin \omega t \, d\omega, \quad \text{or}$$

$$h_s(t) = -\frac{2}{\pi} \int_0^{\infty} \operatorname{Im} H(\omega) \frac{1}{\omega} \cos \omega t \, d\omega.$$

(See also Newman, Hohmann and Anderson, and Kaufman and Keller for all this.)

Thinking in terms of the time-domain inhomogeneous differential equation:

$$L h_{\delta-} = \delta_-$$

$$\Rightarrow L h_{\delta-} = \frac{\partial}{\partial t} H_0$$

$$\Rightarrow L \frac{\partial h_s}{\partial t} = \frac{\partial}{\partial t} H_0.$$

Fake / equivalent world			Real World	
	$\& h(t)$	$\Leftrightarrow$		$\& \frac{\partial h}{\partial t}(t)$
	$\& h(t)$	$\Leftrightarrow$		$\& \frac{\partial h}{\partial t}(t)$

Top left is what we know (flat frequency spectrum for the source and sine transform of the imaginary part of the field), and top right is what we're after. Also, bottom right is obtained from top left by convolution with the box-car, and bottom right is what we're considering it to be. Note that there should really be some minus signs in the above diagram.

Fake / equivalent world			Real World	
	$\& \int^t h(t') dt'$	$\Leftrightarrow$		$\& h(t)$
	$\& \int^t h(t') dt'$	$\Leftrightarrow$		$\& h(t)$

Again, top left is what we know (flat frequency spectrum for the source and sine transform of the imaginary part of the field), and top right is what we're after. Also, bottom right is obtained from top left by convolution with the box-car, and bottom right is what we're considering it to be. Note that there should really be some minus signs in the above diagram.

Fake / equivalent world			Real World	
	$\& h(t)$	$\Leftrightarrow$		$\& \frac{\partial h}{\partial t}(t)$
	$\& h(t)$	$\Leftrightarrow$		$\& \frac{\partial h}{\partial t}(t)$

Top left is what we have, and right is what we're thinking it is. Bottom left is the convolution with a discretized half-

sine, and bottom right is what we're considering it to be: the time-derivative of the H-field for a half-sine waveform.

### 2.2.3 Integration of cubic splined function

The time-domain voltage or magnetic field ends up being known at a number of discrete, logarithmically/exponentially-spaced times as a result of Anderson's cosine/sine digital transform. This time-domain function is cubic splined in terms of the logarithms of the times. Hence, between any two discrete times, the time-domain function is approximated by the cubic spline

$$y(h) = y_0 + q_1 h + q_2 h^2 + q_3 h^3,$$

(see routines **RSPLN** and **RSPLE**) where  $h = \log x - \log t_i$ ,  $x$  is the time at which the function  $y$  is required,  $t_i$  is the  $i^{\text{th}}$  time at which  $y$  is known ( $t_i \leq x \leq t_{i+1}$ ),  $y_0 = y(\log t_i)$ , and  $q_1, q_2$  and  $q_3$  are the spline coefficients. The required integral is

$$\begin{aligned} \int_{x=a}^b y(\log x) dx &= \int_{\log x = \log a}^{\log b} y(\log x) x d(\log x) \\ &= \int_{\log x = \log a}^{\log b} y(\log x) e^{\log x} d(\log x) \\ &= \int_{h = \log a - \log t_i}^{\log b - \log t_i} y(h) e^{(h + \log t_i)} dh \\ &= t_i \int_{h = \log a - \log t_i}^{\log b - \log t_i} y(h) e^h dh. \end{aligned}$$

Substituting the polynomial expression for  $y(h)$  into the above integral and worrying about each term individually gives:

$$\int y_0 e^h dh = y_0 e^h,$$

$$\int q_1 h e^h dh = q_1 e^h (h - 1)$$

(G and R 2.322.1),

$$\int q_2 h^2 e^h dh = q_2 e^h (h^2 - 2h + 2)$$

(G and R 2.322.2), and

$$\int q_3 h^3 e^h dh = q_3 e^h (h^3 - 3h^2 + 6h - 6)$$

(G and R 2.322.3). Hence, summing the integrals above,

$$\begin{aligned} \int_{x=a}^b y(\log x) dx &= t_i y_0 \left( \frac{b}{t_i} - \frac{a}{t_i} \right) \\ &+ t_i q_1 \left( \frac{b}{t_i} (\log b - \log t_i - 1) - \frac{a}{t_i} (\log a - \log t_i - 1) \right) \\ &+ t_i q_2 \left( \frac{b}{t_i} ((\log b - \log t_i)^2 - 2(\log b - \log t_i) + 2) - \right. \\ &\quad \left. \frac{a}{t_i} ((\log a - \log t_i)^2 - 2(\log a - \log t_i) + 2) \right) \\ &+ t_i q_3 \left( \frac{b}{t_i} ((\log b - \log t_i)^3 - 3(\log b - \log t_i)^2 + 6(\log b - \log t_i) - 6) - \right. \\ &\quad \left. \frac{a}{t_i} ((\log a - \log t_i)^3 - 3(\log a - \log t_i)^2 + 6(\log a - \log t_i) - 6) \right). \end{aligned}$$

The original plan was to treat a discretised transmitter current waveform as a piecewise linear function (<sup>th</sup> i.e.), straight line segments between the provided sampled points), which meant that the response coming out of Anderson's filtering routine was convolved with the piecewise constant time-derivative of the transmitter current waveform to give voltages. This proved to be not good enough for on-time calculations (the step-y nature of the approximation of the time derivative of the transmitter current waveform could be seen in the computed voltages). So it was decided to cubic spline the transmitter current waveform, which gives a piecewise quadratic approximation to the time derivative of the waveform. And so the convolution of the stuff coming out of Anderson's routine is now with a constant, a linear time term and a quadratic term. The involved integral above is still required, along with:

$$\begin{aligned}
 \int_{x=a}^b x y(\log x) dx &= \int_{\log x = \log a}^{\log b} y(\log x) x^2 d(\log x) \\
 &= \int_{\log x = \log a}^{\log b} y(\log x) e^{2 \log x} d(\log x) \\
 &= \int_{h = \log a - \log t_i}^{\log b - \log t_i} y(h) e^{(2h + 2 \log t_i)} dh \\
 &= t_i^2 \int_{h = \log a - \log t_i}^{\log b - \log t_i} y(h) e^{2h} dh
 \end{aligned}$$

Using the integrals above for the various powers of  $h$  times  $e^h$ , the relevant integrals for the various parts of the cubic spline representation of  $y(h)$  are:

$$\begin{aligned}
 \int y_0 e^{2h} dh &= y_0 \frac{1}{2} e^{2h}, \\
 \int q_1 h e^{2h} dh &= q_1 \frac{1}{4} e^{2h} (2h - 1), \\
 \int q_2 h^2 e^{2h} dh &= q_2 \frac{1}{8} e^{2h} (4h^2 - 4h + 2), \\
 \int q_3 h^3 e^{2h} dh &= q_3 \frac{1}{16} e^{2h} (8h^3 - 12h^2 + 12h - 6).
 \end{aligned}$$

The limits for the integral are  $h = \log a - \log t_i$  and  $h = \log b - \log t_i$ . The term  $e^{2h}$  becomes:

$$\begin{aligned}
 e^{2(\log X - \log t_i)} &= \{e^{(\log X - \log t_i)}\}^2 \\
 &= \left\{ \frac{e^{\log X}}{e^{\log t_i}} \right\}^2 \\
 &= \left( \frac{X}{t_i} \right)^2 \\
 &= \frac{X^2}{t_i^2}
 \end{aligned}$$

where  $X$  is either  $a$  or  $b$ . Hence,

$$\begin{aligned} \int_{x=a}^b x y(\log x) dx &= t_i^2 y_0 \left( \frac{b^2}{t_i^2} - \frac{a^2}{t_i^2} \right) \\ &+ t_i^2 q_1 \frac{1}{4} \left( \frac{b^2}{t_i^2} (2 \log b - 2 \log t_i - 1) - \frac{a^2}{t_i^2} (2 \log a - 2 \log t_i - 1) \right) \\ &+ t_i^2 q_2 \frac{1}{8} \left( \frac{b^2}{t_i^2} (4(\log b - \log t_i)^2 - 4(\log b - \log t_i) + 2) - \right. \\ &\quad \left. \frac{a^2}{t_i^2} (4(\log a - \log t_i)^2 - 4(\log a - \log t_i) + 2) \right) \\ &+ t_i^2 q_3 \frac{1}{16} \left( \frac{b^2}{t_i^2} (8(\log b - \log t_i)^3 - 12(\log b - \log t_i)^2 + 12(\log b - \log t_i) - 6) - \right. \\ &\quad \left. \frac{a^2}{t_i^2} (8(\log a - \log t_i)^3 - 12(\log a - \log t_i)^2 + 12(\log a - \log t_i) - 6) \right). \end{aligned}$$

And

$$\begin{aligned} \int_{x=a}^b x^2 y(\log x) dx &= \int_{\log x = \log a}^{\log b} y(\log x) x^3 d(\log x) \\ &= \int_{\log x = \log a}^{\log b} y(\log x) e^{3 \log x} d(\log x) \\ &= \int_{h = \log a - \log t_i}^{\log b - \log t_i} y(h) e^{(3h + 3 \log t_i)} dh \\ &= t_i^3 \int_{h = \log a - \log t_i}^{\log b - \log t_i} y(h) e^{3h} dh. \end{aligned}$$

And

$$\begin{aligned} \int y_0 e^{3h} dh &= y_0 \frac{1}{3} e^{3h}, \\ \int q_1 h e^{3h} dh &= q_1 \frac{1}{9} e^{3h} (3h - 1), \\ \int q_2 h^2 e^{3h} dh &= q_2 \frac{1}{27} e^{3h} (9h^2 - 6h + 2), \\ \int q_3 h^3 e^{3h} dh &= q_3 \frac{1}{81} e^{3h} (27h^3 - 27h^2 + 18h - 6). \end{aligned}$$

Hence,

$$\begin{aligned} \int_{x=a}^b x^2 y(\log x) dx &= t_i^3 y_0 \left( \frac{b^3}{t_i^3} - \frac{a^3}{t_i^3} \right) \\ &+ t_i^3 q_1 \frac{1}{9} \left( \frac{b^3}{t_i^3} (3 \log b - 3 \log t_i - 1) - \frac{a^3}{t_i^3} (3 \log a - 3 \log t_i - 1) \right) \\ &+ t_i^3 q_2 \frac{1}{27} \left( \frac{b^3}{t_i^3} (9(\log b - \log t_i)^2 - 6(\log b - \log t_i) + 2) - \right. \\ &\quad \left. \frac{a^3}{t_i^3} (9(\log a - \log t_i)^2 - 6(\log a - \log t_i) + 2) \right) \\ &+ t_i^3 q_3 \frac{1}{81} \left( \frac{b^3}{t_i^3} (27(\log b - \log t_i)^3 - 27(\log b - \log t_i)^2 + 18(\log b - \log t_i) - 6) - \right. \\ &\quad \left. \frac{a^3}{t_i^3} (27(\log a - \log t_i)^3 - 27(\log a - \log t_i)^2 + 18(\log a - \log t_i) - 6) \right). \end{aligned}$$

bigskip In the previous two integrals of the product of  $x$  and  $x^2$  with the function splined in terms of  $\log x$ , the  $x$  and  $x^2$  should really be  $(B - x)$  and  $(B - x)^2$ , where  $B$  is the end of the relevant interval of the splined transmitter current waveform (because it's convolution that's happening):

$$I_1 = \int_{x=a}^b (B - x) y(\log x) dx,$$

and

$$I_2 = \int_{x=a}^b (B - x)^2 y(\log x) dx.$$

Also, it was not really  $x$  and  $x^2$  in those integrals because these terms are coming from the cubic splining of the transmitter current waveform, which means that in each interval between discretization points, it should be  $(x - A)$  and  $(x - A)^2$  that are involved, where  $A$  is the start of the relevant interval for the transmitter current waveform. Because

$$\begin{aligned} \int_{x=a}^b (x - A) y(\log x) dx &= -A \int_{x=a}^b y(\log x) dx + \int_{x=a}^b x y(\log x) dx, \quad \text{and} \\ \int_{x=a}^b (x - A)^2 y(\log x) dx &= A^2 \int_{x=a}^b y(\log x) dx - 2A \int_{x=a}^b x y(\log x) dx + \int_{x=a}^b x^2 y(\log x) dx \end{aligned}$$

and

$$\begin{aligned} \int_{x=a}^b (B - x) y(\log x) dx &= B \int_{x=a}^b y(\log x) dx - \int_{x=a}^b x y(\log x) dx, \quad \text{and} \\ \int_{x=a}^b (B - x)^2 y(\log x) dx &= B^2 \int_{x=a}^b y(\log x) dx - 2B \int_{x=a}^b x y(\log x) dx + \int_{x=a}^b x^2 y(\log x) dx \end{aligned}$$

then

$$\begin{aligned} I_1 &= (B - A) \int_{x=a}^b y(\log x) dx - \int_{x=a}^b (x - A) y(\log x) dx, \quad \text{and} \\ I_2 &= (B - A)^2 \int_{x=a}^b y(\log x) dx - 2(B - A) \int_{x=a}^b (x - A) y(\log x) dx + \int_{x=a}^b (x - A)^2 y(\log x) dx. \end{aligned}$$

## 2.2.4 Computing Sensitivities

The inverse problem of determining the conductivity and/or susceptibility of the Earth from electromagnetic measurements is nonlinear. Program EM1DTM uses an iterative procedure to solve this problem. At each iteration the linearized approximation of the full nonlinear problem is solved. This requires the Jacobian matrix for the sensitivities,  $\mathbf{J} = (\mathbf{J}^\sigma, \mathbf{J}^\kappa)$  where:

$$\begin{aligned} J_{ij}^\sigma &= \frac{\partial d_i}{\partial \log \sigma_j} \\ J_{ij}^\kappa &= \frac{\partial d_i}{\partial \kappa_j} \end{aligned} \tag{2.42}$$

in which  $d_i$  is the  $i^{\text{th}}$  observation, and  $\sigma_j$  and  $\kappa_j$  are the conductivity and susceptibility of the  $j^{\text{th}}$  layer.

The algorithm for computing the sensitivities is obtained by differentiating the expressions for the H-fields with respect to the model parameters ([Farquharson2003]). For example, the sensitivity with respect to  $m_j$  (either the conductivity or susceptibility of the  $j^{\text{th}}$  layer) of the z-component of the H-field for a z-directed magnetic dipole source is given by:

$$\frac{\partial H_z}{\partial m_j}(x, y, z, \omega) = \frac{1}{4\pi} \int_0^\infty \left( e^{-\lambda|z+h|} + \frac{\partial}{\partial m_j} \left[ \frac{P_{21}}{P_{11}} \right] e^{\lambda(z-h)} \right) \lambda^2 J_0(\lambda r) d\lambda \tag{2.42}$$

The derivative of the coefficient is simply:

$$\frac{\partial}{\partial m_j} \left[ \frac{P_{21}}{P_{11}} \right] = \frac{\partial P_{21}}{\partial m_j} \frac{1}{P_{11}} - \frac{\partial P_{11}}{\partial m_j} \frac{P_{21}}{P_{11}^2}$$

where  $P_{11}$  and  $P_{21}$  are elements of the propagation matrix  $\mathbf{P}$ . The derivative of  $\mathbf{P}$  with respect to  $m_j$  (for  $1 \leq j \leq M - 1$ ) is

$$\frac{\partial \mathbf{P}}{\partial m_j} = \mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{j-1} \left( \frac{\partial \mathbf{M}_j}{\partial m_j} \mathbf{M}_{j+1} + \mathbf{M}_j \frac{\partial \mathbf{M}_{j+1}}{\partial m_j} \right) \mathbf{M}_{j+2} \dots \mathbf{M}_M$$

The sensitivities with respect to the conductivity and susceptibility of the basement halfspace are given by

$$\frac{\partial \mathbf{P}}{\partial m_M} = \mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{M-1} \frac{\partial \mathbf{M}_M}{\partial m_M}$$

The derivatives of the individual layer matrices with respect to the conductivities and susceptibilities are straightforward to derive, and are not given here.

Just as for the forward modelling, the Hankel transform in eq. (2.42), and those in the corresponding expressions for the sensitivities of the other observations, are computed using the digital filtering routine of Anderson ([Anderson1982]).

The partial propagation matrices

$$\mathbf{P}_k = \mathbf{M}_1 \prod_{j=2}^k \mathbf{M}_j, \quad k = 2, \dots, M$$

are computed during the forward modelling, and saved for re-use during the sensitivity computations. This sensitivity-equation approach therefore has the efficiency of an adjoint-equation approach.

## 2.3 Inversion Methodologies

In program EM1DTM, there are four different inversion algorithms. They all have the same *general formulation*, but differ in their treatment of the trade-off parameter (see *fixed trade-off*, *discrepancy principle*, *GCV* and *L-curve criterion*).

### 2.3.1 General formulation

The aim of each inversion algorithm is to construct the simplest model that adequately reproduces the observations. This is achieved by posing the inverse problem as an optimization problem in which we recover the model that minimizes the objective function:

$$\Phi = \phi_d + \beta \phi_m \tag{2.43}$$

The two components of this objective function are as follows.  $\phi_d$  is the data misfit:

$$\phi_d = M_d (\mathbf{W}_d (\mathbf{d} - \mathbf{d}^{\text{obs}}))$$

where  $\mathbf{d}^{\text{obs}}$  is the vector containing the  $N$  observations,  $\mathbf{d}$  is the forward-modelled data and  $M_d(\mathbf{x})$  is some measure of the *length* of a vector  $\mathbf{x}$ . It is assumed that the noise in the observations is Gaussian and uncorrelated, and that the estimated standard deviation of the noise in the  $i^{\text{th}}$  observation is of the form  $s_0 \hat{s}_i$ , where  $\hat{s}_i$  indicates the amount of noise in the  $i^{\text{th}}$  observation relative to that in the others, and is a scale factor that specifies the total amount of noise in the set of observations. The matrix  $\mathbf{W}_d$  is therefore given by:

$$\mathbf{W}_d = \text{diag}\{1/(s_0 \hat{s}_1), \dots, 1/(s_0 \hat{s}_N)\}$$



where

$$\rho_E(x) = (x^2 + \epsilon^2)^{p/2} \quad (2.48)$$

The parameter  $p$  can be used to vary the behaviour of this measure. For example, with  $p = 2$ , this measure behaves like the sum-of-squares measure, and a model constructed using this will have the smeared-out, fuzzy appearance that is characteristic of the sum-of-squares measure. In contrast, for  $p = 1$ , this measure does not bias against large jumps in conductivity from one layer to the next, and will result in a piecewise-constant, blocky model. The parameter  $\epsilon$  is a small number, considerably smaller than the average size of an element of the vector. Its use is to avoid the numerical difficulties for zero-valued elements when  $p < 2$  from which the true  $l_p$ -norm users. In program EM1DTM, the values of  $p$  can be different for the smallest and largest components of  $\phi_m$ .

### 2.3.3 General Algorithm

As mentioned in the *computing sensitivities* section, the inverse problem considered here is nonlinear. It is solved using an iterative procedure. At the  $n^{\text{th}}$  iteration, the actual objective function being minimized is:

$$\Phi^n = \phi_d^n + \beta^n \phi_m^n \quad (2.49)$$

In the data misfit  $\phi_d^n$ , the forward-modelled data  $\mathbf{d}^n$  are the data for the model that is sought at the current iteration. These data are approximated by:

$$\mathbf{d}^n = \mathbf{d}^{n-1} + \mathbf{J}^{\sigma, n-1} \delta \mathbf{m} \quad (2.50)$$

where  $\delta \mathbf{m} = \mathbf{m}^n - \mathbf{m}^{n-1}$  and  $\mathbf{J}^{n-1}$  is the Jacobian matrix given by (2.42) and evaluated for the model from the previous iteration. At the  $n^{\text{th}}$  iteration, the problem to be solved is that of finding the change,  $(\delta \mathbf{m}, \delta \mathbf{m}^\kappa)$  to the model which minimizes the objective function  $\Phi^n$ . Differentiating eq. (2.49) with respect to the components of  $\delta \mathbf{m}$  and  $\delta \mathbf{m}^\kappa$ , and equating the resulting expressions to zero, gives the system of equations to be solved. This is slightly more involved now that  $\phi_d$  and  $\phi_m$  comprise the Huber's  $M$ -measure (2.45) and Eklblom's  $l_p$ -like measure, rather than the usual sum-of-squares measures. Specifically, the derivative of (2.46) gives:

$$\frac{\partial M_d}{\partial \delta m_k}(\mathbf{x}) = \sum_{i=1}^N \rho'_H(x_i) \frac{\partial x_i}{\partial \delta m_k}$$

where  $x = W_d(\mathbf{d}^{n-1} + \mathbf{J}^{n-1} \delta \mathbf{m} - \mathbf{d}^{\text{obs}})$ . The vector of the derivatives with respect to the perturbations of the model parameters in all the layers can be written in matrix notation as:

The linear system of equations to be solved for  $\delta \mathbf{m}$  is therefore:

$$\left[ \mathbf{J}^{n-1 T} \mathbf{W}_d^T \mathbf{W}_d \mathbf{J}^{n-1} + \beta^n \sum_{i=1}^2 \mathbf{W}_i^T \mathbf{W}_i + \frac{\gamma^n}{2} \hat{\mathbf{X}}^{n-1 T} \hat{\mathbf{X}}^{n-1} \right] \delta \mathbf{m} = \mathbf{J}^{n-1 T} \mathbf{W}_d^{n-1} \mathbf{W}_d (\mathbf{d}^{\text{obs}} - \mathbf{d}^{n-1}) + \beta^n \sum_{i=1}^2 \mathbf{W}_i^T \mathbf{W}_i (\mathbf{m}_i^{\text{ref}} - \mathbf{m}^{n-1}) + \frac{\gamma^n}{2} \hat{\mathbf{X}}^{n-1 T} \hat{\mathbf{X}}^{n-1} \mathbf{m}^{n-1} \quad (2.51)$$

where  $T$  denotes the transpose and:

$$\begin{aligned} \mathbf{J}^{n-1} &= (\mathbf{J}^{\sigma, n-1} \mathbf{J}^{\kappa, n-1}) \\ \mathbf{W}_1 &= \begin{bmatrix} \sqrt{\alpha_s^\sigma} \mathbf{W}_s^\sigma & 0 \\ 0 & \sqrt{\alpha_s^\kappa} \mathbf{W}_s^\kappa \end{bmatrix} \\ \mathbf{W}_2 &= \begin{bmatrix} \sqrt{\alpha_z^\sigma} \mathbf{W}_z^\sigma & 0 \\ 0 & \sqrt{\alpha_z^\kappa} \mathbf{W}_z^\kappa \end{bmatrix} \\ \mathbf{m}_1^{\text{ref}} &= (\mathbf{m}_s^{\sigma, \text{ref} T} \mathbf{m}_s^{\kappa, \text{ref} T})^T \\ \mathbf{m}_2^{\text{ref}} &= (\mathbf{m}_z^{\sigma, \text{ref} T} \mathbf{m}_z^{\kappa, \text{ref} T})^T \\ \hat{\mathbf{X}}^{n-1} &= (0 (\mathbf{X}^{n-1})^{-1}) \end{aligned}$$

where  $\hat{\mathbf{X}}^{n-1} = \text{diag}\{m_1^{\kappa, n-1}, \dots, m_M^{\kappa, n-1}\}$ . The solution to eq. (2.51) is equivalent to the least-squares solution of:

$$\begin{bmatrix} \mathbf{W}_d \mathbf{J}^{n-1} \\ \sqrt{\beta^n} \mathbf{W}_1 \\ \sqrt{\beta^n} \mathbf{W}_2 \\ \sqrt{\gamma^n/2} \hat{\mathbf{X}}^{n-1} \end{bmatrix} \delta \mathbf{m} = \begin{bmatrix} \mathbf{W}_d (\mathbf{d}^{\text{obs}} - \mathbf{d}^{n-1}) \\ \sqrt{\beta^n} \mathbf{W}_1 (\mathbf{m}_1^{\text{ref}} - \mathbf{m}^{n-1}) \\ \sqrt{\beta^n} \mathbf{W}_2 (\mathbf{m}^{\text{ref}} - \mathbf{m}^{n-1}) \\ \sqrt{\gamma^n/2} \hat{\mathbf{X}}^{n-1} \mathbf{m}^{n-1} \end{bmatrix} \quad (2.52)$$

Once the step  $\delta \mathbf{m}$  has been determined by the solution of eq. (2.51) or eq. (2.52), the new model is given by:

$$\mathbf{m}^n = \mathbf{m}^{n-1} + \nu \delta \mathbf{m} \quad (2.53)$$

There are two conditions on the step length  $\nu$ . First, if positivity of the layer susceptibilities is being enforced:

$$\nu \delta \kappa_j > -\kappa_j^{n-1} \quad (2.54)$$

must hold for all  $j = 1, \dots, M$ . Secondly, the objective function must be decreased by the addition of the step to the model:

$$\phi_d^n + \beta^n \phi_m^n - \gamma^n \phi_{LB}^n < \phi_d^{n-1} + \beta^n \phi_m^{n-1} - \gamma^n \phi_{LB}^{n-1} \quad (2.55)$$

where  $\phi_d^n$  is now the misfit computed using the full forward modelling for the new model  $\mathbf{m}^n$ . To determine  $\mathbf{m}^n$ , a step length ( $\nu$ ) of either 1 or the maximum value for which eq. (2.54) is true (whichever is greater) is tried. If eq. (2.55) is true for the step length, it is accepted. If eq. (2.55) is not true,  $\nu$  is decreased by factors of 2 until it is true.

### 2.3.4 Algorithm 1: fixed trade-off parameter

The trade-off parameter,  $\beta$ , remains fixed at its user-supplied value throughout the inversion. The least-squares solution of eq. (2.52) is used. This is computed using the subroutine ‘‘LSQR’’ of Paige & Saunders ([Paige1982]). If the desired value of  $\beta$  is known, this is the fastest of the four inversion algorithms as it does not involve a line search over trial values of  $\beta$  at each iteration. If the appropriate value of  $\beta$  is not known, it can be found using this algorithm by trail-and-error. This may or may not be time-consuming.

### 2.3.5 Algorithm 2: discrepancy principle

If a complete description of the noise in a set of observations is available - that is, both  $s_0$  and  $\hat{s}_i$  ( $i = 1, \dots, N$ ) are known - the expectation of the misfit,  $E(\phi_d)$ , is equal to the number of observations  $N$ . Algorithm 2 therefore attempts to choose the trade-off parameter so that the misfit for the final model is equal to a target value of  $chifac \times N$ . If the noise in the observations is well known,  $chifac$  should equal 1. However,  $chifac$  can be adjusted by the user to give a target misfit appropriate for a particular data-set. If a misfit as small as the target value cannot be achieved, the algorithm searches for the smallest possible misfit.

Experience has shown that choosing the trade-off parameter at early iterations in this way can lead to excessive structure in the model, and that removing this structure once the target (or minimum) misfit has been attained can require a significant number of additional iterations. A restriction is therefore placed on the greatest-allowed decrease in the misfit at any iteration, thus allowing structure to be slowly but steadily introduced into the model. In program EM1DTM, the target misfit at the  $n^{\text{th}}$  iteration is given by:

$$\phi_d^{n, \text{tar}} = \max(mfac \times \phi_d^{n-1}, chifac \times N) \quad (2.56)$$

where the user-supplied factor  $mfac$  is such that  $0.1 \leq mfac \leq 0.5$ .

The step  $\delta \mathbf{m}$  is found from the solution of eq. (2.52) using subroutine LSQR of Paige & Saunders ([Paige1982]). The line search at each iteration moves along the  $\phi_d$  versus  $\log \beta$  curve until either the target misfit,  $\phi_d^{n, \text{tar}}$ , is bracketed, in which case a bisection search is used to converge to the target, or the minimum misfit ( $> \phi_d^{n-1}$ ) is bracketed, in which

case a golden section search (for example, Press et al., 1986) is used to converge to the minimum. The starting value of  $\beta$  for each line search is  $\beta^{n-1}$ . For the first iteration, the  $\beta (= \beta_0)$  for the line search is given by  $N/\phi_m(\mathbf{m}^\dagger)$ , where  $\mathbf{m}^\dagger$  contains typical values of conductivity and/or susceptibility. Specifically,  $\mathbf{m}^\dagger$  is a model whose top  $M/5$  layers have a conductivity of 0.02 S/m and susceptibility of 0.02 SI units, and whose remaining layers have a conductivity of 0.01 S/m and susceptibility of 0 SI units. Also, the reference models used in the computation of  $\phi_m(\mathbf{m}^\dagger)$  are homogeneous halfspaces of 0.01 S/m and 0 SI units. The line search is efficient, but does involve the full forward modelling to compute the misfit for each trial value of  $\beta$ .

### 2.3.6 Algorithm 3: GCV criterion

If only the relative amount of noise in the observations is known - that is,  $\hat{s}_i (i = 1, \dots, N)$  is known but not  $s_0$  - the appropriate target value for the misfit cannot be determined, and hence Algorithm 2 is not the most suitable. The generalized cross-validation (GCV) method provides a means of estimating, during the course of an inversion, a value of the trade-off parameter that results in an appropriate fit to the observations, and in so doing, effectively estimating the level of noise,  $s_0$ , in the observations (see, for example, [Wahba1990]; [Hansen1998]).

The GCV method is based on the following argument ([Wahba1990]; [Haber1997]; [Haber2000]). Consider inverting all but the first observation using a trial value of  $\beta$ , and then computing the individual misfit between the first observation and the first forward-modelled datum for the model produced by the inversion. This can be repeated leaving out all the other observations in turn, inverting the retained observations using the same value of  $\beta$ , and computing the misfit between the observation left out and the corresponding forward-modelled datum. The best value of  $\beta$  can then be defined as the one which gives the smallest sum of all the individual misfits. For a linear problem, this corresponds to minimizing the GCV function. For a nonlinear problem, the GCV method can be applied to the linearized problem being solved at each iteration ([Haber1997]; [Haber2000]; [Li2003]; [Farquharson2000]). From eq. (2.51), the GCV function for the  $n^{th}$  iteration is given by:

$$GCV(\beta) = \frac{\|\mathbf{W}_d \hat{\mathbf{d}} - \mathbf{W}_d \mathbf{J}^{n-1} \mathbf{M}^{-1} (\mathbf{J}^{n-1 T} \mathbf{W}_d T \mathbf{W}_d \hat{\mathbf{d}} + \mathbf{r})\|^2}{[\text{trace}(\mathbf{I} - \mathbf{W}_d \mathbf{J}^{n-1} \mathbf{M}^{-1} \mathbf{J}^{n-1 T} \mathbf{W}_d T)]^2} \quad (2.57)$$

where

$$\begin{aligned} \mathbf{M}(\beta) &= \left[ \mathbf{J}^{n-1 T} \mathbf{W}_d T \mathbf{W}_d \mathbf{J}^{n-1} + \beta^n \sum_{i=1}^2 \mathbf{W}_i T \mathbf{W}_i + \frac{\gamma^n}{2} \hat{\mathbf{X}}^{n-1 T} \hat{\mathbf{X}}^{n-1} \right] \\ \mathbf{r} &= \beta^n \sum_{i=1}^2 \mathbf{W}_i T \mathbf{W}_i (\mathbf{m}_i^{\text{ref}} - \mathbf{m}^{n-1}) + \frac{\gamma^n}{2} \hat{\mathbf{X}}^{n-1 T} \hat{\mathbf{X}}^{n-1} \mathbf{m}^{n-1} \end{aligned}$$

and  $\hat{\mathbf{d}} - \mathbf{d}^{\text{obs}} - \mathbf{d}^{n-1}$ . If  $\beta^*$  is the value of the trade-off parameter that minimizes eq. (2.57) at the  $n^{th}$  iteration, the actual value of  $\beta$  used to compute the new model is given by:

$$\beta_n = \max(\beta^*, \text{bfac} \times \beta^{n-1}) \quad (2.58)$$

where the user-supplied factor *bfac* is such that  $0.01 < \text{bfac} < 0.5$ . As for Algorithm 2, this limit on the allowed decrease in the trade-off parameter prevents unnecessary structure being introduced into the model at early iterations. The inverse of the matrix  $\mathbf{M}$  required in eq. (2.57), and the solution to eq. (2.51) given this inverse, is computed using the Cholesky factorization routines from LAPACK ([Anderson1999]). The line search at each iteration moves along the curve of the GCV function versus the logarithm of the trade-off parameter until the minimum is bracketed (or  $\text{bfac} \times \beta^{n-1}$  reached), and then a golden section search (e.g., Press et al., 1986) is used to converge to the minimum. The starting value of  $\beta$  in the line search is  $\beta^{n-1}$  ( $\beta^0$  is estimated in the same way as for Algorithm 2). This is an efficient search, even with the inversion of the matrix  $\mathbf{M}$ .

### 2.3.7 Algorithm 4: L-curve criterion

As for the *GCV-based method*, the L-curve method provides a means of estimating an appropriate value of the trade-off parameter if only  $\hat{s}_i, i = 1, \dots, N$  are known and not  $s_0$ . For a linear inverse problem, if the data misfit  $\phi_d$  is plotted

against the model norm  $\phi_m$  for all reasonable values of the trade-off parameter  $\beta$ , the resulting curve tends to have a characteristic “L”-shape, especially when plotted on logarithmic axes (see, for example, [Hansen1998]). The corner of this L-curve corresponds to roughly equal emphasis on the misfit and model norm during the inversion. Moving along the L-curve away from the corner is associated with a progressively smaller decrease in the misfit for large increases in the model norm, or a progressively smaller decrease in the model norm for large increases in the misfit. The value of  $\beta$  at the point of maximum curvature on the L-curve is therefore the most appropriate, according to this criterion.

For a nonlinear problem, the L-curve criterion can be applied to the linearized inverse problem at each iteration (Li & Oldenburg, 1999; Farquharson & Oldenburg, 2000). In this situation, the L-curve is defined using the linearized misfit, which uses the approximation given in eq. (2.50) for the forward-modelled data. The curvature of the L-curve is computed using the formula (Hansen, 1998):

$$C(\beta) = \frac{\zeta' \eta'' - \zeta'' \eta'}{((\zeta')^2 + (\eta')^2)^{3/2}} \quad (2.59)$$

where  $\zeta = \log \phi_d^{lin}$  and  $\eta = \log \phi_m$ . The prime denotes differentiation with respect to  $\log \beta$ . As for both Algorithms 2 & 3, a restriction is imposed on how quickly the trade-off parameter can be decreased from one iteration to the next. The actual value of  $\beta$  chosen for use at the  $n^{th}$  iteration is given by eq. (2.58), where  $\beta^*$  now corresponds to the value of  $\beta$  at the point of maximum curvature on the L-curve.

Experience has shown that the L-curve for the inverse problem considered here does not always have a sharp, distinct corner. The associated slow variation of the curvature with  $\beta$  can make the numerical differentiation required to evaluate eq. (2.59) prone to numerical noise. The line search along the L-curve used in program EM1DTM to find the point of maximum curvature is therefore designed to be robust (rather than efficient). The L-curve is sampled at equally-spaced values of  $\log \beta$ , and long differences are used in the evaluation of eq. (2.59) to introduce some smoothing. A parabola is fit through the point from the equally-spaced sampling with the maximum value of curvature and its two nearest neighbours. The value of  $\beta$  at the maximum of this parabola is taken as  $\beta^*$ . In addition, it is sometimes found that, for the range of values of  $\beta$  that are tried, the maximum value of the curvature of the L-curve on logarithmic axes is negative. In this case, the curvature of the L-curve on linear axes is investigated to find a maximum. As for Algorithms 1 & 2, the least-squares solution to eq. (2.52) is used, and is computed using subroutine LSQR of Paige & Saunders ([Paige1982]).

### 2.3.8 Relative weighting within the model norm

The four coefficients in the model norm (see eq. (2.44)) are ultimately the responsibility of the user. Larger values of  $\alpha_s^\sigma$  relative to  $\alpha_z^\sigma$  result in constructed conductivity models that are closer to the supplied reference model. Smaller values of  $\alpha_s^\sigma$  and  $\alpha_z^\sigma$  result in flatter conductivity models. Likewise for the coefficients related to susceptibilities.

If both conductivity and susceptibility are active in the inversion, the relative size of  $\alpha_s^\sigma$  &  $\alpha_z^\sigma$  to  $\alpha_s^\kappa$  &  $\alpha_z^\kappa$  is also required. Program EM1DTM includes a simple means of calculating a default value for this relative balance. Using the layer thicknesses, weighting matrices  $\mathbf{W}_s^\sigma$ ,  $\mathbf{W}_z^\sigma$ ,  $\mathbf{W}_s^\kappa$  &  $\mathbf{W}_z^\kappa$ , and user-supplied weighting of the smallest and flattest parts of the conductivity and susceptibility components of the model norm (see *acs*, *acz*, *ass* & *asz* in the input file description, line 5), the following two quantities are computed for a test model  $\mathbf{m}^*$ :

$$\begin{aligned} \phi_m^\sigma &= acs \|\mathbf{W}_s^\sigma (\mathbf{m}^* - \mathbf{m}_s^{\sigma,ref})\|^2 + acz \|\mathbf{W}_z^\sigma (\mathbf{m}^* - \mathbf{m}_z^{\sigma,ref})\|^2 \\ \phi_m^\kappa &= ass \|\mathbf{W}_s^\kappa (\mathbf{m}^* - \mathbf{m}_s^{\kappa,ref})\|^2 + asz \|\mathbf{W}_z^\kappa (\mathbf{m}^* - \mathbf{m}_z^{\kappa,ref})\|^2 \end{aligned}$$

The conductivity and susceptibility of the top  $N/5$  layers in the test model are 0.02 S/m and 0.02 SI units respectively, and the conductivity and susceptibility of the remaining layers are 0.01 S/m and 0 SI units. The coefficients of the model norm used in the inversion are then  $\alpha_s^\sigma = acs$ ,  $\alpha_z^\sigma = acz$ ,  $\alpha_s^\kappa = A^s \times ass$  &  $\alpha_z^\kappa = A^d \times asz$  where  $A^s \phi_m^\sigma / \phi_m^\kappa$ . It has been found that a balance between the conductivity and susceptibility portions of the model norm computed in this way is adequate as an initial guess. However, the balance usually requires modification by the user to obtain the best susceptibility model. (The conductivity model tends to be insensitive to this balance.) If anything, the default balance will suppress the constructed susceptibility model.

### 2.3.9 Positive susceptibility

ProgramEM1DTM can perform an unconstrained inversion for susceptibilities (along with the conductivities) as well as invert for values of susceptibility that are constrained to be positive. Following Li & Oldenburg ([Li2003]), the positivity constraint is implemented by incorporating a logarithmic barrier term in the objective function (see eqs. (2.43) & `barrier_cond`). For the initial iteration, the coefficient of the logarithmic barrier term is chosen so that this term is of equal important to the rest of the objective function:

$$\gamma^0 = \frac{\phi_d^0 + \beta^0 \phi_m^0}{-\phi_{LB}^0}$$

At subsequent iterations, the coefficient is reduced according to the formula:

$$\gamma^n = (1 - \min(\nu^{n-1}, 0.925))\gamma^{n-1}$$

where  $\nu^{n-1}$  is the step length used at the previous iteration. As mentioned at the end of the *general formulation*, when positivity is being enforced, the step length at any particular iteration must satisfy eq. (2.54).

### 2.3.10 Convergence criteria

To determine when an inversion algorithm has converged, the following criteria are used ([Gill1981]):

$$\begin{aligned} \Phi^{n-1} - \Phi^n &< \tau(1 + \Phi^n) \\ \|\mathbf{m}^{n-1} - \mathbf{m}\| &< \sqrt{\tau}(1 + \|\mathbf{m}^n\|) \end{aligned} \quad (2.60)$$

where  $\tau$  is a user-specified parameter. The algorithm is considered to have converged when both of the above equations are satisfied. The default value of  $\tau$  is 0.01.

In case the algorithm happens directly upon the minimum, an additional condition is tested:

$$\|\mathbf{g}^n\| \leq \epsilon \quad (2.61)$$

where  $\epsilon$  is a small number close to zero, and where the gradient,  $\mathbf{g}^n$ , at the  $n^{th}$  iteration is given by:

$$\mathbf{g}^n = -2\mathbf{J}^{nT} \mathbf{W}_d^T \mathbf{W}_d (\mathbf{d}^{\text{obs}} - \mathbf{d}^n) - 2\beta^n \sum_{i=1}^2 \mathbf{W}_i^T \mathbf{W}_i (\mathbf{m}_i^{\text{ref}} - \mathbf{m}^{n-1}) - \gamma^n \hat{\mathbf{X}}^{n2T} \mathbf{m}^n$$



---

## Elements of the program

---

The program library consists of the programs:

1. **em1dfm**: performs the inversion
2. **em1dfm fwd**: stand-alone forward modeling program

Each of the above programs requires correctly formatted input files and a specified set of parameters in order to run. These are explained in the following sections:

### 3.1 EM1DFM Main Input File

#### 3.1.1 Line Descriptions

The main input file for the em1dfm code sets up all aspects of the inversion algorithm. This includes setting: the observation file, the inversion algorithm type, what starting and reference models are used, and specifying inversion parameters. **This file must be given the name “em1dtm.in”**. The structure of the file **em1dfm.in** is described below. The *supporting files* and their structures are provided on the following page.

**NOTE:**

- some lines may be omitted depending on the inversion type chosen (which mean the number of lines may change!!!)
- for all file names and their paths, please avoid spaces as Fortran 90 cannot handle spaces (so **do not** put files in C:Program Files on a Windows machine)
- Apart from the root name (which must be a maximum of 20 characters with no spaces), all files specified within the input file must not contain spaces and be no longer than 99 characters

Line #	Parameter	Description
1	<i>rootname</i>	root for name for output files
2	<i>obsfname</i>	name of file containing the field observations
3	<i>mtype</i>	flag for type of model used in the inversion
3a	<i>stconfname</i>	starting conductivity model
3b	<i>stsusfname</i>	starting susceptibility model
3c	<i>rsconfname</i>	reference (smallest) or background conductivity model
3d	<i>rssusfname</i>	reference (smallest) or background susceptibility model
3e	<i>rzconfname</i>	reference (flattest) conductivity model
3f	<i>rzsusfname</i>	reference (flattest) susceptibility model
4	<i>weightname</i>	additional model weights
5	<i>alphas</i>	coefficients for model norm components
6	<i>iatype</i>	type of inversion algorithm
7	<i>iapara(s)</i>	additional inversion algorithm parameter(s)
8	<i>maxniter</i>	maximum number of iterations in an inversion
9	<i>logstretch</i>	stretch factor for logarithmic barrier term
10	<i>numconv</i>	small number for convergence tests
11	<i>hankelevel</i>	number of explicit evaluations of Hankel transform kernels
12	<i>outflg</i>	flag indicating amount of output

- **Line 1 - rootname:** “rootname” is the root for the names of all output files. This string must contain no more than 20 characters! For example, one might use the root name “testinv”. The filenames of all output files from the inversion would therefore begin with “testinv”.
- **Line 2 - obsfname:** “obsfname” is the name of the file containing the field observations. An example of the formatting of the observation file can be found [here](#).
- **Line 3 - mtype:** “mtype” indicates the type of model being recovered in the inversion. It is specified using flag values of 1, 2, 3 or 4. The choice made here affects what is required for the remaining lines in the input file; especially the starting and reference models required. It also impacts the total number of lines contained in the input file. Please check all parameter lines very carefully. The types of model which can be recovered from the inversion are:
  - mtype = 1 implies just conductivity is active in the inversion
  - mtype = 2 implies just susceptibility (with positivity constrained by means of a logarithmic barrier term) is active in the inversion
  - mtype = 3 implies both conductivity and susceptibility are active with susceptibility constrained to be positive
  - mtype = 4 implies both conductivity and susceptibility are active but with no constraints on the susceptibility
- **Line 3a - stconfname:** “stconfname” sets the starting conductivity model for the inversion. If active in the inversion, the name of the file containing the starting conductivity model is entered. The starting conductivity model can be set as the best-fitting halfspace by entering only the layer thicknesses in the file and omitting the conductivities column (i.e. a layers only file).
  - Required if mtype = 1, 3 or 4
  - Omitted if mtype = 2
  - *Formatting for stconfname files*
- **Line 3b - stsusfname:** “stsusfname” sets the starting susceptibility model for the inversion. If active in the inversion, several inputs types can be used to specify the starting susceptibility model.

- Omitted if  $mtype = 1$ .
- For  $mtype=2$ , provide the name of a *model file* **or** a layers-only file (in which case the best-fitting halfspace is used as the starting model).
- For  $mtype=3$  or 4, provide the name of a *model file*, **or** a numerical value for the halfspace susceptibility (since layer thicknesses are known from the conductivity file), **or** “DEFAULT” if the best-fitting halfspace is to be used as the starting model.
- *Formatting for stsusfname files*
- **Line 3c - rsconfname:** “rsconfname” sets the reference conductivity model for the smallness term in the inversion. If active in the inversion, several inputs types can be used to specify this model.
  - Required if  $mtype = 2$ , or if  $mtype = 1, 3$  or 4 with  $acs > 0$
  - Enter “NONE” if not required
  - rsconfname can be entered as the name of a *model file*, **or** as a specified value for a halfspace **or** as “DEFAULT” to set as the best-fitting halfspace.
  - *Formatting for rsconfname* if file is used
- **Line 3d - rssusfname:** “rssusfname” sets the reference susceptibility model for the smallness term in the inversion. If active in the inversion, several inputs types can be used to specify this model.
  - Required if  $mtype = 1$ , or if  $mtype = 2, 3$  or 4 with  $ass > 0$
  - Enter “NONE” if not required
  - rssusfname can be entered as the name of a *model file*, **or** as a specified value for a halfspace **or** as “DEFAULT” to set as the best-fitting halfspace.
  - *Formatting for rssusfname* if file is used
- **Line 3e - rzconfname:** “rzconfname” sets the reference conductivity model for the flatness term in the inversion. If active in the inversion, several inputs types can be used to specify this model.
  - Optional if  $mtype = 1, 3$  or 4
  - Enter “NONE” if you do not want a reference conductivity model in the flatness term
  - rsconfname can be entered as the name of a *model file*, **or** as a specified value for a halfspace **or** as “DEFAULT” to set as the best-fitting halfspace.
  - *Formatting for rzconfname* if file is used
- **Line 3f - rzsusfname:** “rzsusfname” sets the reference susceptibility model for the flatness term in the inversion. If active in the inversion, several inputs types can be used to specify this model.
  - Optional if  $mtype = 2, 3$  or 4
  - Enter “NONE” if you do not want a reference susceptibility model in the flatness term
  - rsconfname can be entered as the name of a *model file*, **or** as a specified value for a halfspace **or** as “DEFAULT” to set as the best-fitting halfspace.
  - *Formatting for rzsusfname* if file is used
- **Line 4 - weightname:** “weightname” can be used weight the relative contributions of layer values and gradients towards the model objective function. There are two options for this functionality.
  - Use “NONE” to indicate that no additional user-supplied weights are to be provided for use in the model norm
  - Use the name of the file containing the user-specified weights. Formatting for weightname can be found [here](#)

- **Line 5 - alphas:** “alphas” control the relative weighting of the smallness and flatness terms for the conductivity and susceptibility towards the model objective function. In the *theory*, the alphas are represented by *acs*, *ass*, *acz* and *asz*:
  - if *mtype* = 1, only values for the two parameters *acs* and *acz* are entered
  - if *mtype* = 2, only values for the two parameters *ass* and *asz* are entered
  - if *mtype* = 3 or 4 enter either:
    - \* the string “DEFAULT” and all four parameters *acs*, *acz*, *ass* and *asz* are required, **OR**
    - \* the six parameters *Ac*, *As*, *acs*, *acz*, *ass* and *asz*, where the value of *Ac* is  $A^c$  in the expression for the model norm below, the value of *As* is  $A^s$ , the value of *acs* is  $\alpha_s^c$ , the value of *acz* is  $\alpha_z^c$ , the value of *ass* is  $\alpha_s^s$ , and the value of *asz* is  $\alpha_z^s$ .
- **Line 6 - iatype:** “iatype” indicates the type of inversion algorithm to be used. Each algorithm computes the trade-off parameter  $\beta$  in a different manner. Algorithms are explained in the *theory section*. Options are:
  - *iatype* = 1 implies a fixed, user-supplied value for the trade-off parameter
  - *iatype* = 2 implies that the trade-off parameter will be chosen by means of a line search so that a target misfit is achieved (or, if this is not possible, then the smallest misfit)
  - *iatype* = 3 implies the trade-off parameter will be chosen using the GCV criterion
  - *iatype* = 4 implies that the trade-off parameter will be chosen using the L-curve criterion
- **Line 7 - iapara:** “iapara” specifies the additional inversion parameters and depends on the choice for “iatype”. The entries required base on type are as follows:
  - if *iatype* = 1, the value of the trade-off parameter is used
  - if *iatype* = 2, the target misfit and greatest allowable decrease in misfit are entered and separated by a space. The target misfit is entered in terms of a chi factor (*chifac*), where the target misfit is the chi factor times the total number of observations for the sounding. The greatest allowable decrease in the misfit at any one iteration is represented by *mfac* (see eq. (2.56))
  - if *iatype* = 3 or 4, enter the greatest allowable decrease in the trade-off parameter at any one iteration in terms of *bfac* (see eq. (2.58))
- **Line 8 - maxnitters:** “maxnitters” is the maximum number of iterations to be carried out in the inversion
- **Line 9 - logstretch:** “logstretch” impacts the logarithmic barrier term for ensuring the recovered susceptibility contains strictly positive values. It is represented by *c* in eq. `barrier_cond`. This field can be entered as either:
  - “DEFAULT” can be entered to indicate a value of 1
  - some other value (a strictly positive real number) can be entered (only used if *mtype* = 2 or 3)

**Line 10: numconv**

“Small” number for convergence tests:

- either “DEFAULT” can be entered to indicate that the default value of 0.01 is to be used in the tests of convergence for an inversion, or,
- if another value is desired, it can be entered on this line;

**Line 11: hanklevel**

Number of explicit evaluations of Hankel transform kernels:

- either “DEFAULT” can be entered to indicate the kernel of the Hankel transforms is to be explicitly evaluated the default number of times (= 50), or,

- if there are concerns about the accuracy of the Hankel transform computations, a number greater than 50 can be entered on this line;

### Line 12: outflg

outflg is the flag indicating the amount of output from the program. (WARNING: it is highly recommended that outflg = 3 or 4 is NOT specified if there are more than a few soundings to be inverted in a single run.)

- outflg = 1 implies the output of a brief convergence / termination report for each sounding plus the final two-dimensional composite model (cond &/or susc) for all the soundings, and the corresponding forward-modelled data. If only one sounding is being considered the model(s) are output in one-dimensional format.
- outflg = 2 implies output as for outflg = 1 plus an iteration by iteration summary of the various components of the objective function.
- outflg = 3 implies output as for outflg = 2 plus the one-dimensional models and corresponding predicted data for each iteration for each sounding. The diagnostics file is also produced.
- outflg = 4 implies output as for outflg = 3 plus any line-search information from misfit, GCV function or L-curve curvature versus trade-off parameter. Also produced is a diagnostics file for the LSQR solution routine if it is used.

## 3.1.2 Examples

### Example 1:

Here, the inversion is set to strictly recover a conductivity model (*mtype* = 1). As a result, there is no line for the starting susceptibility model (*Line 3b*) or flattest susceptibility model (*Line 3f*). A background susceptibility model, however, is provided (*Line 3d*).

```

test      ! Root for output file names.
test.obs  ! Name of the observations file.
1         ! Model type -- just conductivity.
start.con ! Starting conductivity model file.
ref.con   ! Reference conductivity model file.
back.sus  ! Background susceptibility model file.
NONE      ! Reference (flattest) conductivity.
NONE      ! Additional model weights.
0. 1.     ! alpha s & alpha z.
1         ! Type of inversion algorithm -- fixed trade-off param.
10.       ! The value of the trade-off parameter.
15        ! Maximum number of iterations.
DEFAULT   ! Convergence test parameter.
DEFAULT   ! Number of evaluations of the Hankel transform kernel.
3         ! Amount of output.

```

### Example 2:

Here, the inversion recovers both a conductivity and a strictly positive susceptibility model (*mtype* = 3). Using a layers only file, the starting conductivity (*Line 3a*) is set as the best-fitting half-space. The starting susceptibility model (*Line*

3b) was set using a constant value. In this case, fields must be entered for the reference and flattest models for both the conductivity and susceptibility models. Although there are no flattest model, four alphas are required (*Line 5*).

```
hssmz           ! Root for output file names.
hssmz.obs       ! Name of the observations file.
3              ! Model type -- cond. & susc. (with posit.).
layers.dat      ! Starting conductivity model.
0.             ! Starting susceptibility model.
DEFAULT        ! Reference conductivity model.
0.             ! Reference susceptibility model.
NONE           ! Reference (flattest) conductivity.
NONE           ! Reference (flattest) susceptibility.
NONE           ! Additional model weights.
0.01 1. 0.01 1. ! Cond.-susc. balance, acs, acz, ass & asz.
3              ! Type of inversion algorithm -- GCV.
0.5            ! Max. decrease of trade-off param. at an iteration.
15             ! Maximum number of iterations.
DEFAULT        ! Coefficient in log barrier term.
DEFAULT        ! Convergence test parameter.
DEFAULT        ! Number of evaluations of the Hankel transform kernel.
1             ! Amount of output -- minimum.
```

## 3.2 EM1DFM Supporting Files

### 3.2.1 Observation File

This file contains the observations and all the survey parameters, with five lines including parameters as follows:

1. the number of soundings
2. the x- and y-coordinates of each sounding, and the number of frequencies per sounding
3. each frequency, and the number of transmitters for each frequency
4. the dipole moment of each transmitter, their z-coordinates and orientations, and the number of receivers for each
5. the last line contains all of:
  - the dipole moment of each receiver,
  - the transmitter-receiver separation in the x- and y-directions,
  - the z-coordinate and orientation of each receiver,
  - what type of normalization has been applied to the observations,
  - whether in-phase and/or quadrature data are present,
  - the actual observed field values for this receiver, and
  - the type of uncertainties and their values.

**The structure of the observations file is as follows.** Click on the letter labels for variable names, and for parameter details:

*nsounds*

*soundx\_a soundy\_a nfreqs\_a*

*freq\_a nt\_a*

*momt\_a zt\_a ot\_a nr\_a*

*momr\_a trx\_a try\_a zr\_a or\_a ontype\_a octype\_a obs\_a utype\_a uncert\_a*

- **nsounds** is the number of soundings. Each sounding refers to an independent horizontal location where the Earth may be considered as a layered model. Thus, the EM1DFM code inverts for an independent layered Earth model for each sounding. This is used to infer horizontal variability in the Earth's conductivity and/or susceptibility structure.
- **soundx\_a( $i_s$ )** is the x-coordinate of the  $i_s^{th}$  sounding
- **soundy\_a( $i_s$ )** is the y-coordinate of the  $i_s^{th}$  sounding
- **nfreqs\_a( $i_s$ )** is the number of frequencies for the  $i_s^{th}$  sounding
- **freq\_a( $i_f, i_s$ )** is the frequency (Hz) of the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **nt\_a( $i_f, i_s$ )** is the number of transmitters for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **momt\_a( $i_t, i_f, i_s$ )** is the dipole moment ( $\text{Am}^2$ ) of the  $i_t^{th}$  transmitter for the  $i_f^{th}$  for the  $i_s^{th}$  sounding
  - This number is used as a simple scaling within the program. If  $momt_a = 2$ , then the forward-modelled observations are twice what they would be if  $momt_a = 1$ .
- **zt\_a( $i_t, i_f, i_s$ )** is the z-coordinate (metres, negative upwards from zero on the Earth's surface) of the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **ot\_a( $i_t, i_f, i_s$ )** is the orientation of the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding (“x” for an x-directed dipole, “y” for a y-directed dipole, and “z” for a vertical (downward-directed) dipole)
- **nr\_a( $i_t, i_f, i_s$ )** is the number of receivers for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **momr\_a( $i_t, i_f, i_s$ )** is a scale factor for the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding which allows the incorporation of any necessary parameters of the receiver that might mean the observations are not simply point measurements of the H-field.
  - An example of necessary parameters could be coil area and/or number of turns and/or orientation (e.g.,  $momr_a = 1$  for an upward-pointing z-directed receiver dipole).
  - This number simply appears as a scale factor within the code (i.e. if  $momr_a = 2$ , then the forward-modelled observations are twice what they would be if  $momr_a = 1$ )
  - *NOTE:* Some common data formats (such as DIGEM coaxial data - not coplanar data) require  $momr_a = -1$  to make these data compatible with the normalization convention used by EM1DFM. The first example requires this type of normalization
- **trx\_a( $i_r, i_t, i_f, i_s$ )** is the transmitter-receiver separation (m) in the x-direction between the  $i_r^{th}$  receiver and the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **try\_a( $i_r, i_t, i_f, i_s$ )** is the transmitter-receiver separation (m) in the y-direction between the  $i_r^{th}$  receiver and the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **zr\_a( $i_r, i_t, i_f, i_s$ )** is the z-component (metres, negative upwards from zero on the Earth's surface) of the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding

- **or\_a(i<sub>r</sub>, i<sub>t</sub>, i<sub>f</sub>, i<sub>s</sub>)** is the orientation of the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
  - “x” for an x-directed dipole,
  - “y” for a y-directed dipole, and
  - “z” for a vertical (downward-directed) dipole
- **ontype\_a(i<sub>r</sub>, i<sub>t</sub>, i<sub>f</sub>, i<sub>s</sub>)** is the type of normalization of the data/datum for the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
  - *ontype\_a* = 1 indicates the data are values in ppm of the secondary magnetic field normalized by the free-space magnetic field
  - *ontype\_a* = 2 indicates the data are values in % of the secondary magnetic field normalized by the free-space magnetic field
  - *ontype\_a* = 3 indicates the data are values of the secondary H-field in A/m, and
  - *ontype\_a* = 4 indicates the data are values of the total H-field in A/m
- **octype\_a(i<sub>r</sub>, i<sub>t</sub>, i<sub>f</sub>, i<sub>s</sub>)** is the observation type for the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
  - *octype\_a* = “b” indicates both inphase and quadrature observations are present
  - *octype\_a* = “i” just the inphase observation is present
  - *octype\_a* = “q” just the quadrature datum
- **obs\_a(i<sub>r</sub>, i<sub>t</sub>, i<sub>f</sub>, i<sub>s</sub>)** is the pair of inphase and quadrature observations, or just the in-phase observation, or just the quadrature observation, for the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
- **utype\_a(i<sub>r</sub>, i<sub>t</sub>, i<sub>f</sub>, i<sub>s</sub>)** indicates the form in which the uncertainties are provided for the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding
  - *utype\_a* = “v” for absolute uncertainties in the same units as the observations, and
  - *utype\_a* = “p” percentage uncertainties
- **uncert\_a(i<sub>r</sub>, i<sub>t</sub>, i<sub>f</sub>, i<sub>s</sub>)** is the pair of uncertainties for the inphase and quadrature observations, or the uncertainty in just the inphase observation, or the uncertainty in just the quadrature observation, for the  $i_r^{th}$  receiver for the  $i_t^{th}$  transmitter for the  $i_f^{th}$  frequency for the  $i_s^{th}$  sounding

### Example for a single sounding

The observations file for a single sounding (that is, a single one-dimensional model) at x = 0 m, y = 0 m for an airborne-type configuration is shown below.

- There are three frequencies for the horizontal coplanar loop configuration (880, 7213, 55840 Hz)
  - a z-directed magnetic dipole transmitter (dipole moment = 1 Am<sup>2</sup>) and receiver (dipole moment = 1 Am<sup>2</sup>) for each frequency separated by 8.1 m, 8.1 m and 6.3 m respectively in the x-direction, 0 m in the y-direction
  - both transmitter and receiver for each frequency are at a height of 40 m above the Earth’s surface
- There are two frequencies for the coaxial loop configuration (5848 and 1082 Hz)

```

1
0. 0. 5
880. 1
1. -40. z 1
1. 8.1 0. -40. z 1 b 2.474 30.29 v 1.00 1.58
7213. 1
1. -40. z 1
1. 8.1 0. -40. z 1 b 75.52 203.4 v 3.94 9.92
55840. 1
1. -40. z 1
1. 6.3 0. -40. z 1 b 261.2 208.3 v 13.2 11.0
5848. 1
1. -40. x 1
-1. 8.1 0. -40. x 1 b 13.44 43.13 v 1.00 2.12
1082. 1
1. -40. x 1
-1. 8.1 0. -40. x 1 b 2.035 10.93 v 1.00 1.00

```

- an x-directed magnetic dipole transmitter (dipole moment = 1 A m<sup>2</sup>) and receiver (dipole moment = 1 A m<sup>2</sup>) for both frequencies separated by 8.1 m in the x-direction and 0 m in the y-direction;
- both transmitter and receiver for each frequency are at a height of 40 m above the Earth’s surface.
- The observations are values of the secondary magnetic field normalized by the free-space field and expressed in terms of parts-per-million (ppm).
- Both inphase and quadrature components of the field are supplied.
- The uncertainties are expressed in absolute terms in the same units as the observations (i.e., ppm).

#### Example for EM-31 data for two soundings

The observations file for EM31-type data for two soundings is shown below

```

2
60.0 0. 1
9800. 4
1 -1.0 z 1
-1. 3.66 0.0 -1.0 z 2 q -0.6613E+01 v 0.3307E+00
1 -1.0 y 1
-1. 3.66 0.0 -1.0 y 2 q -0.5286E+01 v 0.2643E+00
1 -0.05 z 1
-1. 3.66 0.0 -0.05 z 2 q 0.8821E+01 v 0.4410E+00
1 -0.05 y 1
-1. 3.66 0.0 -0.05 y 2 q 0.9712E+01 v 0.4856E+00
60.0 -10.0 1
9800. 4
1 -1.0 z 1
-1. 3.66 0.0 -1.0 z 2 q -0.6665E+01 v 0.3332E+00
1 -1.0 y 1
-1. 3.66 0.0 -1.0 y 2 q -0.4623E+01 v 0.2311E+00
1 -0.05 z 1
-1. 3.66 0.0 -0.05 z 2 q -0.9256E+01 v 0.4628E+00
1 -0.05 y 1
-1. 3.66 0.0 -0.05 y 2 q -0.8282E+01 v 0.4141E+00

```

- Sounding locations are at x = 60 m & y = 0 m and x = 60 m & y = 10 m.
- There is one frequency (9.6 kHz),
- There are four instrument positions:

- at waist height (1 m) and on the ground (0.05 m),
- held both normally (vertical transmitter and receiver coil axes) and on its side (horizontal coil axes).
- The transmitter and receiver coils are separated by 3.66 m in the x-direction. Only the quadrature part of the normalized secondary H-field (in %) is provided as data,
- The uncertainties are absolute in %.

### 3.2.2 Files for reference and starting models

#### Conductivity files

##### Required if `mtype = 1, 3 or 4` in the input file

This is the file containing the starting conductivity model for all soundings if conductivity is active in the inversion. The relevant quantities are the number of layers, and the thickness (m) and conductivity (S/m) of each layer. A dummy value for the thickness of the basement halfspace is required in this file, but nothing is ever done with it after it is read in. If conductivity is active in the inversion, it is from this file that the program gets the number of layers and their thicknesses, which then must be the same for all other models read in by the program. This file is therefore required for mode `mtype = 1, 3 or 4` (i.e. conductivity is active in the inversion). This file can also be a layers-only file to indicate that the best-fitting halfspace is to be used as the starting model.

The structure of this file is as follows (just as for all 1D model files). For a layers-only file the conductivity column is left blank:

```

nlayers
thicks_a(1)          con_a(1)
thicks_a(2)          con_a(2)
...
thicks_a(nlayers-1) con_a(nlayers-1)
0.                   con_a(nlayers)

```

Fig. 3.1: File structure for conductivity models

- `nlayers` is the number of layers in the model,
- `thicks_a(j)` is the thickness in metres of the `j`th layer and
- `con_a(j)` is the conductivity in S/m of the `j`th layer.

An example file for a conductivity model made up of 12 layers (including the basement halfspace) is shown below. The thicknesses of the first eleven layers increase from 4.7987 m to 135.76 m. This model is a homogeneous halfspace of 0.3 milliSeimens per m.

#### Susceptibility files

##### Required if `mtype = 2` in the input file

This file contains the starting susceptibility model for all soundings if susceptibility is active in the inversion. The file contains the number of layers, and the thickness (m) and susceptibility (SI units) of each layer. A dummy value for the thickness of the basement halfspace is required, but nothing is done with it after it is read in.

```

12
4.7987  0.30E-03
5.0994  0.30E-03
5.7584  0.30E-03
6.9101  0.30E-03
8.8117  0.30E-03
11.941  0.30E-03
17.194  0.30E-03
26.311  0.30E-03
42.785  0.30E-03
73.931  0.30E-03
135.76  0.30E-03
0.0     0.30E-03

```

Fig. 3.2: Conductivity file example

If only susceptibility is active (i.e., `mtype = 2`), the inversion program gets its information about the number of layers in the model and their thicknesses from this file. If both conductivity and susceptibility are active (i.e., `mtype = 3` or `4`), the program gets the number of layers and their thicknesses from the starting conductivity model - see above. All other models (e.g., reference models) read in must then have the same number of layers with exactly the same thicknesses.

The structure of this file is as follows (just as for all 1D model files). For a layers-only file the susceptibility column is left blank:

```

nlayers
thicks_a(1)      sus_a(1)
thicks_a(2)      sus_a(2)
...
thicks_a(nlayers-1) sus_a(nlayers-1)
0.              sus_a(nlayers)

```

Fig. 3.3: File structure for susceptibility models

- **nlayers** is the number of layers in the model,
- **thicks\_a(j)** is the thickness in metres of the *j*th layer and
- **sus\_a(j)** is the susceptibility in SI units of the *j*th layer.

## Other inputs for starting and reference models

### File for reference conductivity model (for smallest model component) (Optional)

The parameter describing the reference conductivity model for the **smallest** component of the model norm if one is required for the inversion, or describing the reference conductivity model if only susceptibility is active in the inversion. The parameter is either “NONE”, **OR** a file **OR** a value for a halfspace **OR** “default” if the best-fitting halfspace is to be used. It is required if  $mtype = 1, 3$  or  $4$  and  $acs > 0$ , or if  $mtype = 2$ . If a file is used, it must have the same format as the starting conductivity model file (see above), and it must have the same number of layers with exactly the same thicknesses as the starting conductivity and/or susceptibility model.

### File for reference susceptibility model (for smallest model component) (Optional)

The parameter describing the reference susceptibility model for the **smallest** component of the model norm if one is required for the inversion, or describing the reference susceptibility model if only conductivity is active in the inversion. That is, this model is required if  $mtype = 2, 3$  or  $4$  and  $ass > 0$ , or if  $mtype = 1$ . The parameter is either “NONE”, **OR** a file **OR** a value for a halfspace **OR** “default” if the best-fitting halfspace is to be used. If a file is used, it must be in the same format as all model files (see above), and must have the same number of layers with exactly the same thicknesses as the starting conductivity and/or susceptibility model.

### File for reference conductivity model (for flattest model component) (Optional - not available in the GUI)

The parameter describing the reference conductivity model for the **flattest** component of the model norm if one is required for the inversion. Whether or not this parameter is specified in “em1dfm.in” determines whether or not such a reference model plays a part in the inversion. The parameter is either “NONE”, **OR** a file **OR** a value for a halfspace **OR** “default” if the best-fitting halfspace is to be used. If a file is used, it must be in the same format as the starting conductivity model file (see section above), and must have the same number of layers with exactly the same thicknesses as the starting conductivity and/or susceptibility model.

### File for reference susceptibility model (for flattest model component) (Optional - not available in the GUI)

The parameter describing the reference susceptibility model for the **flattest** component of the model norm if one is required for the inversion. Whether or not this parameter is specified in “em1dfm.in” determines whether or not such a reference model plays a part in the inversion. The parameter is either “NONE”, **OR** a file **OR** a value for a halfspace **OR** “default” if the best-fitting halfspace is to be used. If a file is used, it must be in the same format as the starting susceptibility model file (see section above), and must have the same number of layers with exactly the same thicknesses as the starting conductivity and/or susceptibility model.

## 3.2.3 File for additional model-norm weights

The file containing the information about the additional weighting of the layers for some or all of the four possible components of the model norm: smallest and flattest components for conductivity and susceptibility.

- **The first line of this file must contain two (if  $mtype = 1$  or  $2$ ) or four (if  $mtype = 3$  or  $4$ ) integers (which can either have the value 0 or 1)**
  - e.g., “1 0” for  $mtype = 1$  implies that additional weights are supplied for use in the smallest component of the model norm but not the flattest component for only conductivity active in the inversion;
  - “1 0” for  $mtype = 2$  implies that additional weights are supplied for use in the smallest component of the model norm but not the flattest component for only susceptibility active in the inversion;
  - “1 0 1 0” for  $mtype = 3$  or  $4$  implies that additional weights are supplied for both the smallest component of the conductivity portion of the model norm and the smallest component of the susceptibility portion of the model norm, but not for the flattest components, when both conductivity and susceptibility are active in the inversion.
- The second line of this file must contain the number of layers in the model. The order of the four possibilities must be the same as shown below, with any set of weights that is not needed by the program simply omitted.

```

ics icz iss isz
nlayers
(uswcs_a(j),    j = 1, ..., nlayers )
(uswcs_a(j),    j = 1, ..., nlayers )
(uswss_a(j),    j = 1, ..., nlayers)
(uswcz_a(j),    j = 1, ..., nlayers-1)
(uswsz_a(j),    j = 1, ..., nlayers-1)

```

The parameters within the this file are described as follows:

- *ics*, *icz*, *iss* & *isz* are the four integers that indicate the presence of weights for the smallest and flattest components of the model norm for conductivity and the smallest and flattest components of the model norm for susceptibility (if *mtype* = 2, *ics* & *icz* are omitted),
- *nlayers* is the number of layers in the model;
- *uswcs\_a(j)* is the weight for the *j*th layer in the smallest component of the conductivity portion of the model norm;
- *uswss\_a(j)* is the weight for the *j*th layer in the smallest component of the susceptibility portion of the model norm;
- *uswcz\_a(j)* is the weight for the difference between the *j*th and (*j*+1)th layers in the flattest component of the conductivity component of the model norm; and
- *uswsz\_a(j)* is the weight for the difference between the *j*th and (*j*+1)th layers in the flattest component of the susceptibility component of the model norm.

The supplied weights must be greater than zero. A weight greater than one increases the weight relative to the default setting, and a weight less than one decreases the weight relative to the default setting

## 3.3 EM1DFM Output Files

### 3.3.1 Standard output

#### Always given

Reports on the progress of the inversion are written to the standard output (usually the screen). The amount of this output depends on the value of *outflag* (see line 12 of the input file). Any error messages generated by the program will also be written to the standard output, and also to the file “em1dfm.out”, which is described next.

As an example, we will look at the command line output for the *first input file example* and the *first observation file example*. The starting conductivity file is the one from the *conductivity model* section. The reference models were: conductivity =  $10^{-4}$  S/m and susceptibility = 0 SI units. Since *outflag* = 3, the following was written to the standard output:

In this case, we see the iteration-by-iteration variation of the components of the objective function: the misfit, the trade-off parameter, the model norm and the complete objective function. For exactly the same example, but for *outflag* = 1, the output gives only a summary of the final state of the inversion. Note that the final values of the components of the objective function are also printed out. Thus:

```
PROGRAM "EM1DFM" (v1.0).

Start at 17:34:57.344, 28/06/2000.

Sounding 1 (0.0,0.0).
Initial phid= 2080.0, beta= 10.000, phim= 0.0000, Phi= 2080.0.
Iteration 1: phid= 1168.7, beta= 10.000, phim= 0.19895, Phi= 1170.7.
Iteration 2: phid= 736.22, beta= 10.000, phim= 0.18341, Phi= 738.06.
Iteration 3: phid= 380.57, beta= 10.000, phim= 0.24765, Phi= 383.05.
Iteration 4: phid= 114.11, beta= 10.000, phim= 0.30980, Phi= 117.20.
Iteration 5: phid= 12.815, beta= 10.000, phim= 0.15531, Phi= 14.368.
Iteration 6: phid= 6.2389, beta= 10.000, phim= 0.18378, Phi= 8.0767.
Iteration 7: phid= 6.0415, beta= 10.000, phim= 0.13129, Phi= 7.3544.
Iteration 8: phid= 5.6881, beta= 10.000, phim= 0.15350, Phi= 7.2230.
Iteration 9: phid= 5.7826, beta= 10.000, phim= 0.14203, Phi= 7.2029.
Convergence.

Final conductivity model written to "test.con".
Predicted data written to "test.prd".

The End! [17:35:52.387, 28/06/2000]
```

```
PROGRAM "EM1DFM" (v1.0).

Start at 17:43:57.855, 28/06/2000.

Sounding 1 (0.0,0.0).
Convergence: n= 9, phid= 5.7826, beta= 10.000, phim= 0.14203, Phi= 7.2029.

Final conductivity model written to "test.con".
Predicted data written to "test.prd".

The End! [17:44:50.798, 28/06/2000]
```

### Description of possible status messages

- “*Convergence*”: Convergence according to the criteria given in (2.60) within the *background theory* section. That is, the relative change in the objective function from one iteration to the next and the relative change in the model norm are less than the convergence parameter on line 10 of the input parameter file. Possible parameters which may appear next to *convergence* are (see (2.55)):
  - *n* - The iteration number.
  - *phid* - The misfit.
  - *beta* - The trade-off parameter.
  - *phim* - The model norm.
  - *gamma* - The coefficient of the logarithmic barrier term.
  - *phiLB* - The logarithmic barrier term.
  - *Phi* - The whole objective function.
  - (*Phi*) - Previous objective function
- “*Convergence (small gradient)*”: Convergence according to the criterion given in Eq. (2.61) within the *background theory* section. That is, the gradient is essentially zero.
- “*Target misfit not attained convergence to minimum*”: For inversion algorithm 2 the target misfit could not be reached, but convergence to the minimum misfit has occurred.
- “*No suitable step found*”: No step length was found that decreased the objective function, even after being decreased by a factor of  $2^{**nmax}$ . See Eq. (2.53) in the *background theory* section.
- “*Max number of iterations done without convergence*”: The convergence criteria have not been satisfied in the specified maximum number of iterations.
- “*Starting susceptibilities reset because of small values*”: When positivity is being enforced, the starting values of the susceptibilities are bumped up to be at least 0.001 SI units.

### 3.3.2 Main output file

#### Always given and always called em1dfm.out

This file contains a copy of all the progress reports that are written to the *standard output*. The extent of these reports depends on the parameter *outflg*. In addition, a summary of the inputs read in by the program are printed at the top of this file.

For the example provided in the *standard output section*, the file “em1dfm.out” is:

In the case that *outflg* was set to 1, the main output file would be:

### 3.3.3 Final model(s)

#### Always given

If only a single sounding is being inverted:

- the final one-dimensional conductivity model (if conductivity is active in the inversion, i.e., *mtype* = 1, 3 or 4) will be written to the file “rootname.con”. It has the same format as the input one-dimensional conductivity models (see *files for reference and starting models*),

```
PROGRAM "EM1DFM" (v1.0).

Run started at 11:57:59.223, 29/06/2000.

Reading inputs . . .
Filename root:           "test".
Number of soundings:     1.
Observations file:      "test.obs".
Model type:              1 (conductivity).
Starting conductivity file: "start.con".
Ref. (small.) cond. file: "ref.con".
Background susc. file:   "back.sus".
Ref. (flat.) cond. file: NONE.
Additional model weights: NONE.
alpha s:                 0.00.
alpha z:                 1.00.
Inversion algorithm:     1 (fixed trade-off parameter).
Supplied trade-off parameter: 10.00.
Max. number of iterations: 15.
Convergence param. (default): 0.100E-01.
No. of lambdas (default): 50.
Level of output:         3.
Finished reading inputs.

Sounding 1 (0.0,0.0).
Initial phid= 2080.0, beta= 10.000, phim= 0.0000, Phi= 2080.0.
Iteration 1: phid= 1168.7, beta= 10.000, phim= 0.19895, Phi= 1170.7.
Iteration 2: phid= 736.22, beta= 10.000, phim= 0.18341, Phi= 738.06.
Iteration 3: phid= 380.57, beta= 10.000, phim= 0.24765, Phi= 383.05.
Iteration 4: phid= 114.11, beta= 10.000, phim= 0.30980, Phi= 117.20.
Iteration 5: phid= 12.815, beta= 10.000, phim= 0.15531, Phi= 14.368.
Iteration 6: phid= 6.2389, beta= 10.000, phim= 0.18378, Phi= 8.0767.
Iteration 7: phid= 6.0415, beta= 10.000, phim= 0.13129, Phi= 7.3544.
Iteration 8: phid= 5.6881, beta= 10.000, phim= 0.15350, Phi= 7.2230.
Iteration 9: phid= 5.7826, beta= 10.000, phim= 0.14203, Phi= 7.2029.
Convergence.

Conductivity model: "test.con".
Predicted data:     "test.prd".
Diagnostics:        "test 001.dgns".

The End! [11:59:26.530, 29/06/2000]
```

```
PROGRAM "EM1DFM" (v1.0).
Run started at 17:43:57.893, 28/06/2000.
Reading inputs ...

Filename root:           "test".
Number of soundings:     1.
Observations file:      "test.obs".
Model type:             1 (conductivity).
Starting conductivity file: "start.con".
Ref. (small.) cond. file: "ref.con".
Background susc. file:  "back.sus".
Ref. (flat.) cond. file: NONE.
Additional model weights: NONE.
alpha s:                0.00.
alpha z:                1.00.
Inversion algorithm:    1 (fixed trade-off parameter).
Supplied trade-off parameter: 10.00.
Max. number of iterations: 15.
Convergence param. (default): 0.100E-01.
No. of lambdas (default): 50.
Level of output:        1.
Finished reading inputs.

Finished reading inputs.

Sounding 1 (0.0,0.0).
Convergence: n= 9, phid= 5.7826, beta= 10.000, phim= 0.14203, Phi= 7.2029.

Conductivity model: "test.con".
Predicted data:     "test.prd".

The End! [17:44:50.798, 28/06/2000]
```

- the final one-dimensional susceptibility model (if susceptibility is active in the inversion, i.e., *mtype* = 2, 3 or 4) will be written to the file “rootname.sus”. It has the same format as the input one-dimensional susceptibility models (see *files for reference and starting models*).

If two or more soundings are being inverted:

The final one-dimensional conductivity models for all soundings (if conductivity is active in the inversion) are written to the file “rootname\_con.mod”, and the final one-dimensional susceptibility models for all soundings (if susceptibility is active in the inversion) are written to the file “rootname\_sus.mod”. The structure of these files is as follows:

```
Number of layers:  nlayers
Layer thicknesses (m):  thicks_a(1) . . . thicks_a(nlayers-1)
Number of soundings:  nsounds
Sounding x- & y-coordinates, Conductivities (S/m) ...
soundx_a(1)  soundy_a(1) val_a(1,1) . . . val_a(nlayers,1)
. . .
soundx_a(nsounds)  soundy_a(nsounds) val_a(1,nsounds) . . . val_a(nlayers,nsounds)
```

- *nlayers* is the number of layers in the one-dimensional models for all soundings,
- *thicks\_a(j)*,  $j = 1, \dots, nlayers-1$ , are the thicknesses of the layers in the one-dimensional models,
- *nsounds* is the number of soundings,
- *soundx\_a(i)* and *soundy\_a(i)*,  $i = 1, \dots, nsounds$ , are the x- and y-coordinates of the soundings, and
- *val\_a(j,i)* is the value of the model (either conductivity in S/m or susceptibility in SI units) in the *j*th layer for the *i*th soundings.

If susceptibility is being written out, then “Conductivities (S/m)” on line 4 is replaced with “Susceptibilities (SI units)”. The final model(s) for each sounding are appended to this/these file(s) as soon as the inversion for each sounding has completed.

### 3.3.4 Final forward modeled data

#### Always given

The forward-modelled data for the final model is written to the file “rootname.prd”. The format for this file is the same as that for the input observations file (see *observation file*), but without the information about the uncertainties. The data for each sounding are appended to this file as soon as the inversion for each sounding has completed.

### 3.3.5 Final components of the objective function

#### Always given is *nsounds* > 1

If there are more than one sounding, the components of the objective function for the final model for each sounding are written out to the file “rootname\_phis.out”. All information for a sounding is written on one line in this file. The possible column headings are:

- *x* - The x-coordinate of the sounding.
- *y* - The y-coordinate of the sounding.
- *phid* - The misfit.
- *beta* - The trade-off parameter.
- *phim* - The model norm.

- *gamma* - The coefficient of the logarithmic barrier term.
- *phiLB* - The logarithmic barrier term.
- *Phi* - The whole objective function.
- *phim con* - The conductivity part of the model norm.
- *phim sus* - The susceptibility part of the model norm.

The values are appended to this file on completion of the inversion for each sounding.

### 3.3.6 Iteration-by-iteration one-dimensional models for each sounding

#### Only if `outflg >= 3`

If `outflg >= 3`, the one-dimensional conductivity and/or susceptibility model(s) obtained at each iteration in the inversion for each sounding are written out.

- The conductivity models are written to the files “rootname\_isound\_iter.con” where *isound* is the number of the sounding and *iter* is the number of the iteration (*iter* = 0 indicates the starting conductivity model). These files have the same format as the input conductivity files (see [files for reference and starting models](#)).
- The susceptibility models are written to the files “rootname\_isound\_iter.sus” where *isound* is the number of the sounding and *iter* is the number of the iteration (*iter* = 0 indicates the starting susceptibility model). These files have the same format as the input susceptibility files (see [files for reference and starting models](#)).
- If only conductivity is active in the inversion, the background susceptibility is written out to the file “rootname\_isound.sus”.
- If only susceptibility is active in the inversion, the background conductivity is written out to the file “rootname\_isound.con”.

### 3.3.7 Iteration-by-iteration forward-modelled data for each sounding

#### Only if `outflg >= 3`

If `outflg >= 3`, the forward-modelled data for each iteration for each sounding are written out to the files “rootname\_isound\_iter.dprd” where *isound* is the number of the sounding and *iter* is the number of the iteration (*iter* = 0 indicates the forward-modelled data for the starting model). The data are written out as they are ordered in the input observations file, but with none of the survey parameters. The observations (including their uncertainties) for each sounding are also written out in this format to the file “rootname\_isound.dobs”.

### 3.3.8 Diagnostics for each iteration for each inversion

#### Only if `outflg >= 3`

If `outflg >= 3`, the values of all the interesting quantities at each iteration for each sounding are written to the file(s) “rootname\_isound.dgns”, where *isound* is the number of the sounding. The possible quantities in these files are summarized in the following table. References to equations are found in the [inversion methodology section](#).

The same status message as that written to the standard output and `em1dfm.out` is also written to this file.

<code>tau</code>	The convergence parameter (see eqs. 68 & 69 ).
<code>epsilon</code>	The small number against which the norm of the gradient is compared (eq. 70)
<code>nnmax</code>	The max. number of times the step length can be halved.
<code>Iter</code>	Iteration number.
<code>normg</code>	The norm of the gradient.
<code>phid</code>	The misfit.
<code>beta</code>	The trade-off parameter.
<code>phim</code>	The model norm.
<code>gamma</code>	The coefficient of the logarithmic barrier term.
<code>phiLB</code>	The logarithmic barrier term.
<code>Phi</code>	The whole objective function.
<code>(Phi)</code>	Previous objective function: previous $\phi_d$ , $\phi_m$ & $\phi_{LB}$ and current $\beta$ & $gamma$ . See eq. (57).
<code>lambda</code>	The step length.
<code>normcdm</code>	The norm of the change in the model
<code>normm</code>	The norm of the model.

### 3.3.9 Misfit line search values

#### Only if `outflg = 4` and `iatype = 2`

If `outflg = 4`, and `iatype = 2` (i.e., a line search over the misfit is used at each iteration to choose the trade-off parameter - see *discrepancy principle*, the values of the trade-off parameter and the corresponding values of the misfit during each line search at each iteration of each inversion are written to the file “phidvsbeta”. There is just a single such file for a whole run of the program with information from each iteration separated by pairs of dashed lines. Note that the pairs of values of the trade-off parameter and misfit are written out in the order in which they are computed during the line search: they are not re-ordered.

### 3.3.10 GCV function line search values

#### Only if `outflg = 4` and `iatype = 3`

If `outflg = 4`, and `iatype = 3` (i.e., a line search over the GCV function is used at each iteration to choose the trade-off parameter - see *GCV criterion*, the values of the trade-off parameter and the corresponding values of the GCV function during each line search at each iteration of each inversion are written to the file “GCVvsbeta”. There is just a single such file for a whole run of the program with information from each iteration separated by pairs of dashed lines. Note that the pairs of values of the trade-off parameter and GCV function are written out in the order in which they are computed during the line search: they are not re-ordered.

### 3.3.11 L-curve line search diagnostics

#### Only if `outflg = 4` and `iatype = 4`

If `outflg = 4`, and `iatype = 4` (i.e., a line search over the curvature of the L-curve is used at each iteration to choose the trade-off parameter - see *L-curve criterion*, the values of the trade-off parameter and the corresponding values of the linearized misfit and the model norm during each line search at each iteration of each inversion are written to the file “phisvsbeta”, and the values of the trade-off parameter and the corresponding values of the curvature computed in log-log/linear space are written to the files “curlovsbeta”/“curlvlsbeta”. There is just a single version of each of these files for a whole run of the program with the information for each iteration separated from that for others by pairs of dashed lines.

### 3.3.12 Diagnostics for LSQR subroutine

#### Only if outflg = 4

If *outflg* = 4, diagnostics are written out from Saunder's LSQR subroutine to the file "lsqr.out". Warning: this file can become very large very quickly if there is more than one sounding.

## 3.4 EM1DFMFWD Program

### 3.4.1 Input file

The main input file for the em1dfmfwd code sets up all aspects of the forward modeling. This includes setting: the observation file and the physical property model(s). Unlike for the inversion code, the main input file for the EM1DFMFWD does not require a specified name. The structure of the main input file is described below. The supporting files and their structures were provided earlier.

```
test.obstype! Name of the observations-type file.
sigma.in    ! Conductivity model file.
susc_hs.in  ! Susceptibility model file
100         ! Number of evaluations of the Hankel transform kernel.
y           ! Is noise to be added?
5. 1. 1.   ! Percentage noise, threshold & seed.
```

### 3.4.2 Observation file

Observation files have the same format as those for the EM1DFM program. See [here](#). In this case, the uncertainties are no required.

### 3.4.3 Model file(s)

Formatting for conductivity and susceptibility model files is provided [here](#).

### 3.4.4 Output

The output from program EM1DFMFWD is a file called "em1dfmfwd.out". If noise has not been added, it is in the same format as the predicted data files output from program EM1DFM. If noise has been added, it is in the same format as the observations files for program EM1DFM with the standard deviations of the noise added to each data written out in place of *uncert\_a*. In addition, if noise has been added, the actual noise added to each datum and the total chi-squared sum of the noise are appended to the bottom of the file *em1dfmfwd.out* along with the noise-free observations.

Consider the following conductivity and susceptibility models:

The resulting output file "em1dfmfwd.out" would be:

```

3
20.0 0.005
30.0 0.005
0.0 0.0001
-----
3
20.0 0.0
30.0 0.0
0.0 0.0

```

```

1
0. 0. 5
880. 1
1. -40. z
1. 8.1 0. -40. z 1 b 2.474 30.29 v 1.00 1.58
7213. 1
1. -40. z
1. 8.1 0. -40. z 1 b 75.52 203.4 v 3.94 9.92
55840. 1
1. -40. z
1. 6.3 0. -40. z 1 b 261.2 208.3 v 13.2 11.0
5848. 1
1. -40. x
-1. 8.1 0. -40. x 1 b 13.44 43.13 v 1.00 2.12
1082. 1
1. -40. x
-1. 8.1 0. -40. x 1 b 2.035 10.93 v 1.00 1.00

Added noise, N(0,1) Added noise
inphase quadrature inphase quadrature
-8.85E-02 -0.804 -8.85E-02 -1.27
-0.835 0.5 -3.29 4.96
-0.187 -1.04 -2.46 -11.4
-1.27 0.361 -1.27 0.765
1.11 1.34 1.11 1.34

Total chi-squared noise added = 7.478

Noise-free data:

1
0. 0. 5
880. 1
1. -40. z
1. 8.1 0. -40. z 1 b 2.563 31.56
7213. 1
1. -40. z
1. 8.1 0. -40. z 1 b 78.81 198.4
55840. 1
1. -40. z
1. 6.3 0. -40. z 1 b 263.7 219.8
5848. 1
1. -40. x
-1. 8.1 0. -40. x 1 b 14.71 42.37
1082. 1
1. -40. x
-1. 8.1 0. -40. x 1 b 0.930 9.59

```

---

## Running the programs

---

The software package EM1DFM contains 2 codes:

- **EM1DFM:** carries out the inversion of small loop, frequency-domain EM data assuming a layered Earth model
- **EM1DFMFWD:** a stand-alone program for forward modeling the frequency-domain response assuming a layered Earth model

### 4.1 Command Prompt/Terminal Syntax

Both programs in the package can be executed within Windows or Linux environments. They can be run by typing the program name followed by the *main input file* name into the command prompt (Windows) or terminal (Linux); the program and main input file are separated by a space.

```
C:\Users\Sarah\Documents\Sandbox\em1dfm>em1dfmfwd.exe em1dfmfwd.in
EM1DFMfwd - Version 1.0
Developed by University of British Columbia
Geophysical Inversion Facility <UBC-GIF>
(C) Copyright 1992 - 2002, UBC-GIF,
Department of Earth and Ocean Sciences, UBC
http://www.geop.ubc.ca/ubcgif/
Licensed to:
For academic use within UBCGIF
```

For success, ensure the following:

- That the current directory in the command line is the same as the directory where the *main input file* and all *supporting files* are located
- That you have pathed the location of the EM1DFM and/or EM1DFMFWD to you environment variables (only if you want to avoid entering the entire filepath of the executable)

- That the formatting of the main input file and all supporting files are correct

## 4.2 Command Prompt/Terminal Outputs

Upon execution, outputs are displayed through the command prompt/terminal to indicate whether the program is running successfully. If there are errors, the output message will indicate the nature of the error. When running an inversion, the command prompt/terminal will display status messages. Status messages are described in the *Standard Output* section.

## 5.1 Example 1: Single Sounding

Here, we consider the forward modeling and inversion of DIGHEM-style data, collected over a 3-layered Earth model. Data were collected at 5 frequencies for a single sounding. The inversion was performed using algorithm 2 (*discrepancy principal*) and set to recover a conductivity model and a positive susceptibility model ( $mtype = 3$ ).

- [Download zip file for example 1](#), which contains all the files necessary to run the forward and inverse problems.

## 5.2 Example 2: Multiple Soundings

Here, we consider a line of small loop FEM data collected over two anomalous bodies. The westernmost body is resistive and highly susceptible while the easternmost body is conductive and moderately susceptible. The original data were produced synthetically with 3D FEM codes. The observation file contains data collected for 29 soundings at 10 frequencies. The inversion was performed using the algorithm 4 (*L-curve criterion*) and set to recover both a conductivity model and a positive susceptibility model ( $mtype = 3$ ). Since there are multiple soundings (29), the conductivity and susceptibility model files contain one physical property model for each sounding. As we can see, localized 1D inversion produce artifacts due to the 3D nature of the problem. However, the depth to each target and their horizontal locations are well-resolved.

- [Download zip file for example 2](#), which contains all the files necessary to run the inverse problems.

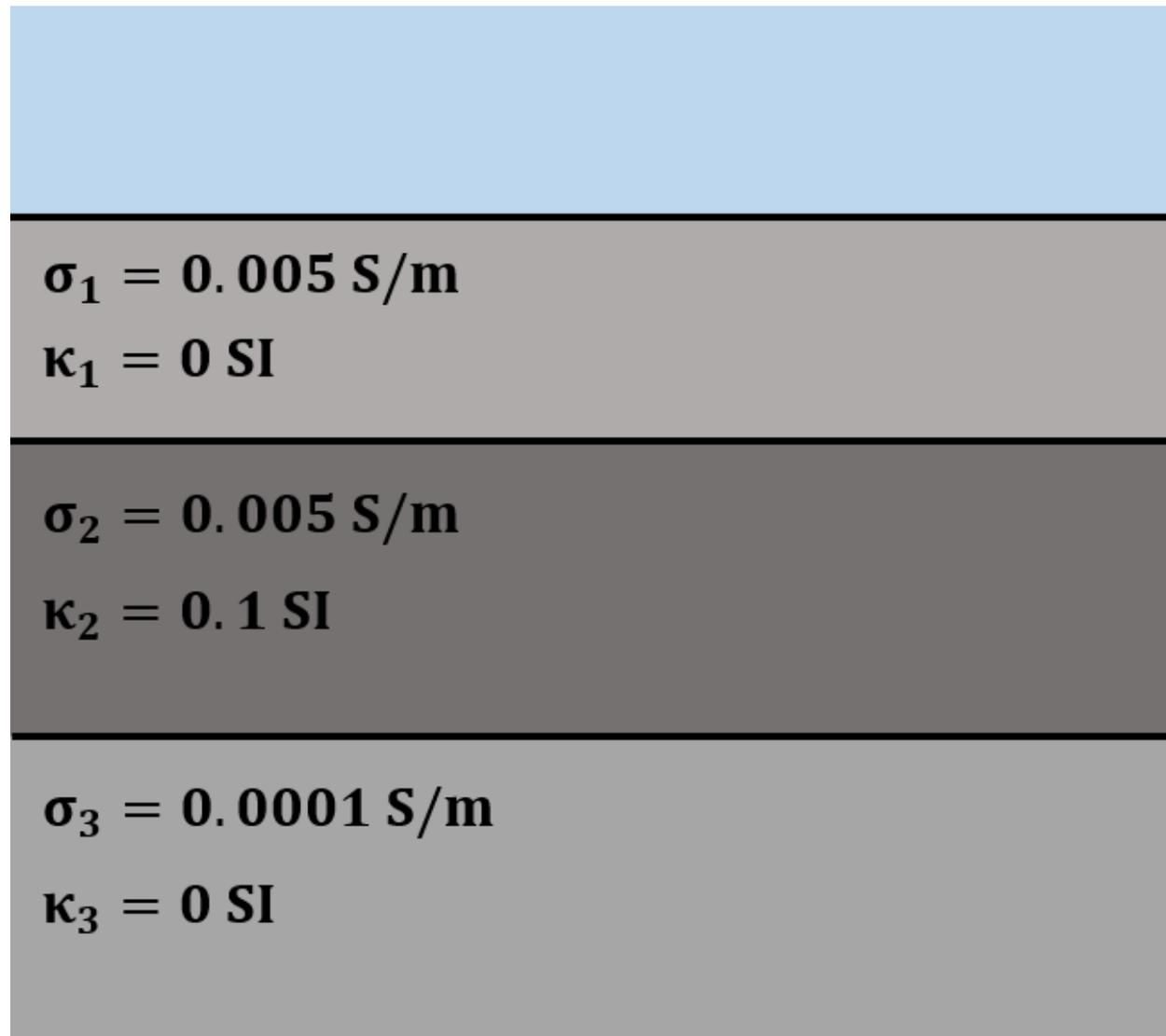


Fig. 5.1: True model

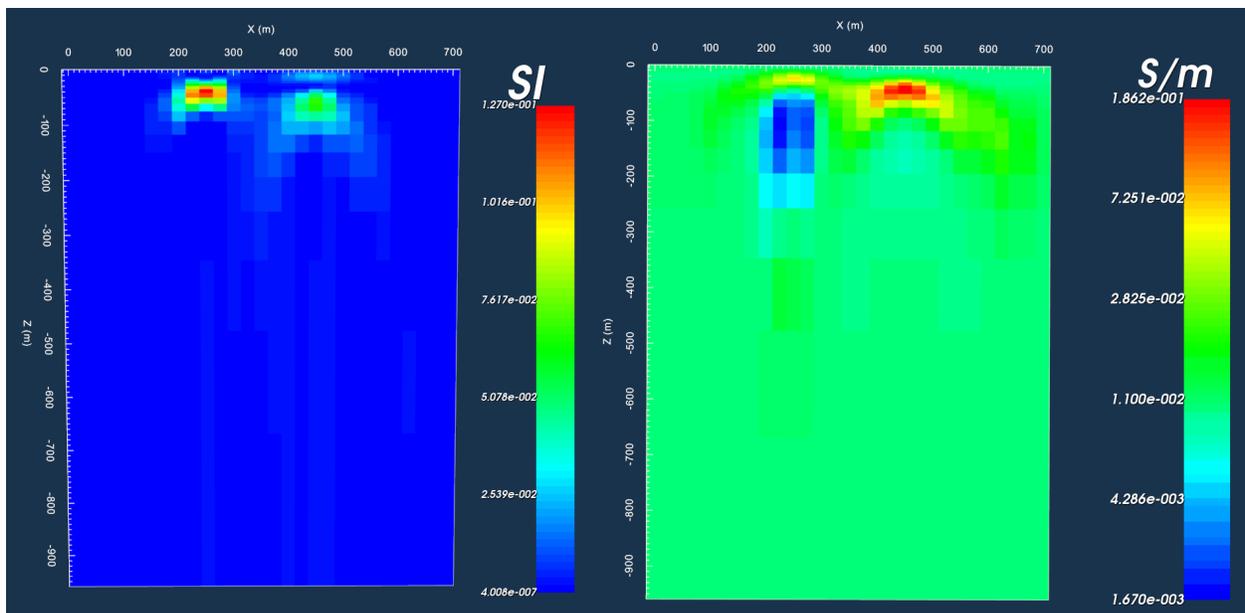


Fig. 5.2: Stitched susceptibility and conductivity models recovered from inversion.



## CHAPTER 6

---

References

---