MANUAL FOR FORTRAN SUBROUTINE

TEM1D

FOR CALCULATION OF TRANSIENT

ELECTROMAGNETIC RESPONSES

Niels B. Christensen · Aarsdalevej 4 · DK-8210 Aarhus V
nbc@geo.au.dk

CONTENTS

INTRODUCTION

This is a users manual for the open source FORTRAN subroutine TEM1D that calulates transient electromagnetic (TEM) responses from one-dimensional (1D) earth models. If you are interested in using the routine, please read the publication:

Christensen N.B., Christiansen A.V., Auken E. and Foged N., 2025. An open source FORTRAN subroutine for calculation of TEM responses and derivatives from 1D models. Computers and Geosciences, XX

Presently,there is intense activity in the EM community in the writing of new programs in several different languages, e.g. Python and MATLAB, involving TEM responses. An asset of these relatively new languages is that they have powerful built-in functions and very efficient commands for vector and matrix manipulations, and they are thus very well suited for writing the inversion part of a TEM inversion program. However, they are often not compilable to an executable program. A consequence of this is that functions and subroutines that are called repeatedly will define the overall execution speed of the whole program. One way of getting the best of all worlds is to write a program where the inversion routines are written in a higher order language and where the routines that are called tens of thousands of times are relegated to an external code written in a language that can be compiled. This will overcome the problems with computation time and at the same time permit an inversion formulation in a higher language.

In the paper mentioned above, the considerations and the numerical approaches behind the subroutine is discussed in detail. However, below a few paragraphs will be repeated from the paper to introduce the capabilities of the subroutine: what it can so and what it will not do.

BASIC CODE DEVELOPMENT CONSIDERATIONS

The overall considerations concern the question of what should be included in the code and what should be left to the user. It has been important to keep the code as general as possible, i.e. it should accommodate all the basic tasks involved in modelling most of the measuring configurations presently used in TEM surveys, including a full implementation of the effects of instrumentation, often called the system response. The intent has been to build an effective and reliable code: a compact and efficient tool for TEM computations.

Below is a list of some of the most important questions addressed in the development process and the answers reached after discussions.

(1) The code accommodates the modelling of IP effects.

IP parameters can be included in the forward response, but derivatives with respect to the IP parameters will not be calculated. The IP effect is expressed through a Cole-Cole model and is defined by three additional model parameters for each layer. Only rarely is brute force inversion carried out on IP parameters. The user can of course calculate numerical derivatives of the IP parameters if needed.

(2) The supported waveform is regarded as a piecewise linear waveform.

With new digital instruments, there is an increasing number of different ways to record and sample the waveform, and it will not be possible to accommodate them all. Besides, presently, practices change quite rapidly. The waveform is treated as a piecewise linear waveform, and if a user puts in a densely sampled waveform with thousands of samples, it will result in a long calculation time. It is recommended that users reduce their waveform definition to fewer samples while still maintaining the accuracy and resolution of the waveform.

(3) Presently the code accommodates the vertical, but nor the horizontal field component.

The vertical part of the secondary TEM field is by far the most used for one-dimensional (1D) inversion of TEM data. A horizontal component response might be included later, or, with a limited effort, implemented by a user who needs it.

(4) The code will not include an option of approximate responses.

Computational resources have reached a point where the calculation of accurate responses is not a general issue. If a user wishes to make use of approximate response, their calculation is so fast that it can be included in a calling program. If interested, see Christensen (2002; 2016).

(5) The code will not include the option of two moments.

Though some TEM instruments make use of two transmitter (Tx) moments, the code will include only one moment. If more than one moment is required, the response routine can be called twice which of course entails a certain computational overhead.

(6) The code does not accommodate integration over gates.

With the advent of digital systems and a dense sampling of the instrument signals, an improved gating has become possible by choosing a smooth, several times differentiable gating weight function where both its value and several derivatives of the weight function will go to zero at the gate end points, thereby suppressing noise much better than a simple box-car weight function, see e.g. https://en.wikipedia.org/wiki/Window_function. At the moment, there are many approaches to doing this and several different weight functions are in use. It was therefore decided that the code does not deliver an integration over the gates and that this is left up to the user. Most gate integration procedures are quite simple to implement and computationally fast, so there is no point in the code getting in the way of the practices of the various users by forcing a special gate integration on the response. The response and the derivatives are delivered densely sampled in a wide time interval so that the user can implement his/her own gate integration.

(7) Configuration options

The instrument configurations supported by the subroutine have been chosen so that most commonly used configurations are accommodated. The Tx options are a circular Tx loop and a polygonal, piecewise linear loop. The calculation time is shorter for the circular Tx loop, and for most airborne applications, a circular Tx loop is a very good approximation to a polygonal one, provided it has the same area. For instruments with a polygonal Tx loop and a zero coupled receiver (Rx), an equivalent lateral distance can be found for the circular loop approximation so that the Rx is still zero coupled to the Tx. The Rx is always modelled as a dipole.

To accommodate new developments in instrumentation, the subroutine has the option of including two Txs and two Rxs with user-chosen polarities. In the present version of the subroutine, responses and derivatives are calculated for just one lateral distance between Tx and Rx, so in the case of including more than one Tx-Rx pair, all of the four possible Tx-Rx configurations must have the same lateral Tx-Rx distance. If that is not the case, the subroutine will have to be called more than once.

CALLING THE TEM1D SUBROUTINE - DEFINING THE INPUT PARAMETERS

The routine is called as:

```
 SUBROUTINE TEM1D (
# IMLMi, NLAYi, RHONi, DEPNi,
# IMODIPi, CHAIPi, TAUIPi, POWIPi,
# TXAREAi, RTXRXi, IZEROPOSi,
# ISHTX1i, ISHTX2i, ISHRX1i, ISHRX2i, HTX1i, HTX2i, HRX1i, HRX2i,
# NPOLYi, XPOLYi, YPOLYi, X0RXi, Y0RXi,
# IRESPTYPEi, IDERIVi, IREPi, IWCONVi, NFILTi, REPFREQi, FILTFREQi,
# NWAVEi, TWAVEi, AWAVEi,
# NTOUT,TIMESOUT, RESPOUT, DRESPOUT)
```

All variables starting with the letters I, J, K, L, M, N are INTEGER*4.

All other variables are REAL*8.

There are some COMPLEX*16 variables, but they are all confined to the subroutine TEM1DFHT.for and used in the calculation of the Hankel filter coefficients.

All units are SI units.

Parameters defining the 1D model

IMLMi	[0 | 1] : [Few-layer model (FLM) | Multi-layer model (MLM)]
NLAYi	The number of layers of the 1D model.
RHONi	[1 : NLAYi] The resistivities of the 1D model
DEPNi	[1 : NLAYi] The depths to the top of the NLAYi layers. DEPNi(1) = 0.
IMODIPi	[0 | 1] : [Do not | Do include IP parameters in the model].
CHAIPi	[1 : NLAYi] The IP chargeability of the NLAYi layers.
TAUIPi	[1 : NLAYi] The IP time constants of the NLAYi layers.
POWIPi	[1 : NLAYi] The power of the IP expressions in the NLAYi layers.

The value of IMLMi is relevant only if derivatives are requested. For forward modelling only, it has no significance.

For IMODIPi = 0, the three IP parameter arrays are dummy.

Parameters defining the instrument configuration.

TXAREAi	The transmitter (Tx) area
RTXRXi	The lateral distance between Tx and Rx
IZEROPOSi	Find the zero-coupled position of the Rx in case of using a circular approximation to a polygonal Tx for which the Rx is zero coupled.
ISHTX1i	Polarity of first Tx.
ISHTX2i	Polarity of second Tx.
ISHRX1i	Polarity of first Rx.
ISHRX2i	Polarity of second Rx.
HTX1i	Elevation of first Tx.
HTX2i	Elevation of second Tx.
HRX1i	Elevation of first Rx.
HRX2i	Elevation of first Rx.

The polarity parameters can have the values: -1, and 1, indicating negative and positive polarity, respectively. It can also have a value of 0, meaning that the Tx or Rx does not exists.

Parameters defining a polygonal Tx loop.

NPOLYi	The number of polygonal sides. If NPOLYi = 0, the Tx is circular.
XPOLYi	[1 : NPOLYi] The x–coordinates of the apices of the polygon
YPOLYi	[1 : NPOLYi] The y–coordinates of the apices of the polygon
X0RXi	The x–coordinate of the Rx.
Y0RXi	The y–coordinate of the Rx.

Parameters defining the response types and the system response.

IRESPTYPEi	[0 | 1 | 2] : [Step | Impulse | Convolved response]	
IDERIVi	[0 | 1] : [Do not | Do calculate derivatives].	
IREPi	[0 | 1] : [Do not | Do model repetition effects].	
IWCONVi	[0 | 1] : [Do not | Do convolve with waveform].	
NFILTi	Number of first-order filters to be applied.	
REPFREQi	Repetition frequency of the waveform.	
FILTFREQi	[1 : NFILTi] Repetition frequencies of the filters.	
NWAVEi	Number of samples of the waveform.	
TWAVEi	[1 : NWAVEi]	Times of the waveform samples
AWAVEi)	[1 : NWAVEi]	Amplitudes of the waveform samples.

A value of IRESPTYPEi = 0 will result in a principal step response, however, with optional filters, but repetition and waveform is not modelled.

A value of IRESPTYPEi = 1 will result in a principal impulse response, however, with optional filters, but repetition and waveform is not modelled.

A value of IRESPTYPEi = 2 will result in a response convolved with the elements of the system response individually chosen through the system response parameters.

The zero time of TWAVEi is the start of the turnoff part of the waveform.

Derivatives are calculated by differentiating through the recurrence relations defining the response in the wavenumber / Laplace domain. This ensures that they are as accurately calculated as the responses.

>> More about filters.

Output parameters

NTOUT Number of delay times of the output arrays.
TIMESOUT[1 : NTOUT] The delay times of the output arrays.
RESPOUT [1 : NTOUT] The response at the delay times TIMESOUT.
DRESPOUT [1 : NTOUT , 1 : NPARM] The derivatives.

The TIMESOUT array goes from 10 ns to the latest offtime possible given the repetition frequency, or, if no repetition is modelled, to a delay time of 100 ms.

The NPARM variable indicates the number of parameters involved in the derivatives. For IMLMi = 1, only derivatives with regard to the NLAYi layer conductivities are found, plus the derivative with regard to Tx height - in that sequence, so NPARM = NLAYi+1.

For IMLMi = 0, derivatives are found with regard to both the NLAYi layer conductivities and the (NLAYi-1) thicknesses,plus the derivative with regard to Tx height - in that sequence: conductivities, thicknesses and Tx height, so NPARM = 2*NLAYi.

COMMON BLOCKS

The input parameters are copied to variables in COMMON blocks, most of which are defined through INCLUDE statements referring to the INCLUDE files. Beside the parameters in the call of TEM1D, the COMMON blocks will also store parameters calculated in various subroutines that are used in other subroutines.

ARRAYSDIMBL.INC
INSTRUMENTBL.INC
IPBL.INC
MODELBL.INC
POLYGONBL.INC
RESPBL.INC
WAVEBL.INC

NUMERICAL CALCULATION APPROACHES AND ALGORITHMS

TEM responses are calculated through recurrence relations in the wavenumber / frequency domain or - as is the case with this code - in the wavenumber / Laplace domain. The latter has been chosen because it is faster - it involves only REAL calculations, not COMPLEX ones, and because it is more accurate for very early times.

The wavenumber / Laplace expression is then subjected to an inverse Laplace transform using the Gaver-Stehfest algorithm with 16 coefficients. Subsequently, the result is subjected to a Hankel trams from from wavenumber to space domain using the Fast Hankel transform filter approach. The filter coefficients are calculated the first time the convolution routine FHTCONV is called and then stored.

The numerical accuracy of the Hankel transform is determined by the number of samples per decade, NDEC. Experiments show that a value of NDEC = 10 is quite sufficient for reliable computations. The value of NDEC is set in the response routines prior to the call to the FHT convolution routine, but all of these are overridden by the statement at the beginning of the FHTCONV routine where the value is set to 10. If for some reason you'd like to change NDEC, so at the beginning of the FHTCONV routine.

The response from a polygonal loop is calculated as an integration of the vertical magnetic field from a horizontal electric dipole around the perimeter of the loop. The integral of each side is found as the integral of a local spline representation of the sample values on the side.

Repetition modelling is calculated as an interpolation in the primary step response at delay times going back three halfperiods.

Wavefom convolution is calculated as a convolution between the step response and the second derivative of the piecewise linear waveform. This second derivative is a series of delta functions, so that the convolution becomes a simple interpolation.

The subroutine accommodates custom low-pass filters by defining it as a series of first-order low-pass filers (a maximum of 16), and the final filter consists of the product of the first-order filters.

The effect of the series of first-order low-pass filters is modelled by multiplying with the Laplace transform of the filters in the recursion of the kernel function. This is a simple and numerically stable approach.

All interpolations are done assuming a local spline representation of the sampled function.

There are no input / output statements in the TEM1D subroutine and the other subroutines. None of them reads from or writes to files or the default output (console). All parameter exchange is given in the calling statements of the routines. This ensures that there will be no interference between the program shell that calls TEM1D and the TEM1D routine itself in relation to the operating system.

FINAL REMARKS

Finally, do read the paper for more detailed information on the subroutine, its capabilities, limitations, and numerical algorithms. The paper also includes the references relevant for this work.

Good luck with the subroutine! May it serve you well!

May 2025 / Niels B Christensen