MANUAL

FOR THE FORTRAN SUBROUTINE

TEM1DPROG

FOR CALCULATION OF TRANSIENT

ELECTROMAGNETIC RESPONSES

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

CONTENTS

INTRODUCTION

Presently, several of the commercially available programs for forward calculation, analysis and inversion of TEM data have restrictions in terms of proprietary rights of usage and modification. At the same time, there is intense activity in the EM community in the writing of new programs in several different languages, e.g. Python, MATLAB, and others. for the calculation of transient electromagnetic (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 built on an interpretation approach, meaning that, in essence, they are executed one line at a time and not compiled to an executable program. A consequence of this is that functions and subroutines that are called repeatedly will set a boundary for the 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 compilable language. This will overcome the problems with computation time and at the same time permit an inversion formulation in a higher language.

In the case of McMC approaches, where forward responses may be needed hundreds of thousands of times or more, it would be advantageous to have an externally compiled code for the forward responses. Also in the training of AI networks, rapid forward calculations are of crucial importance.

All of the cases mentioned above have at least one requirement in common: a fast forward routine for TEM responses, preferably also delivering the derivatives of the response with regard to model parameters to facilitate inversion. Most modern languages are capable of including an external routine as a compiled unit in a different language that will permit a fast computation of the forward responses and derivatives. An external FORTRAN routine would meet that need.

These considerations lie behind the development of a FORTRAN code that can deliver forward responses and derivatives. The code is formulated as a subroutine and it is up to the user to see to that it is properly compiled to a form that can be integrated with the user's own code. It has been attempted to make a subroutine that would cover most of the instrument configurations and modes of application that are in use today, but obviously not everything can be covered. The subroutine is delivered as source code written in FORTRAN 77, and everyone is welcome to update or change the code according to their particular needs. And to fix bugs, of course, but if you find some, please communicate your findings to the undersigned.

BASIC CODE DEVELOPMENT CONSIDERATIONS

In the planning of the development of the code, several questions were discussed with people from the local EM community.. The overall considerations concern the question of what should be included in the code and what should be left to the user. The program is intended for widespread and diverse usage, and this emphasises the importance of keeping the code as general as possible. At the same time, the code should be able to accommodate all the basic tasks involved in modelling most of the measuring configurations presently used in TEM surveys, and a full implementation of the effects of instrumentation, often called the system response. So the intent is to build an effective and reliable code: a compact tool for TEM computational issues.

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 will not include the option of *approximate responses*.

Computational resources have reached a point where the calculation of accurate responses is not an issue. If a user wishes to apply an approximate response, it is so fast that it can be included in a calling program. If interested, see Christensen (1997; 2002; 2016).

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

The code will include only one moment. Some TEM instruments make use of two transmitter (Tx) moments, but in that case the response routine can be called twice with only a small computational overhead.

(3) Presently the code accommodates z- *responses* but not x- responses.

The vertical part of the secondary TEM field is by far the most used for inversion of TEM data. An x-response might be included later.

(4) The code accommodates *IP effects*.

The 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 defined by three additional model parameters for each layer. The IP parameters are easily integrated in the recursive calculation of the kernel function with only a somewhat increased computation time. Only rarely is brute force inversion carried out on IP parameters. The user can of course make numerical derivatives of the IP parameters if needed.

(5) The supported waveform is a *piecewise linear (PWL) waveform.*

With new digital instruments, there are 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. If people put in a densely sampled waveform with thousands of samples, it will be treated as a piecewise linear waveform, but that can cause a substantial increase in calculation time. It is suggested that users reduce their waveform definition to fewer samples while still maintaining the accuracy and resolution of the waveform.

(6) The code does not accommodate *integration over the 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 gate 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. At the moment, there are many approaches to doing this and many weight functions are in use. It is 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, so there is no point in the code ruining the intentions of the user by forcing a special gate integration on the response. The response - subjected to all of the other system response convolutions - will be delivered densely sampled in a wide time interval so that the user can implement his/her own gate integration.

PROGRAM OPTIONS AND CAPABILITIES

The forward responses

The code will calculate forward response of a TEM system in the quasi-static approximation (QSA), meaning that magnetic permeability and set to its vacuum value and the electrical permittivity is set to zero. The instrument must be on or above a one-dimensional (1D) earth model where the model layers are completely characterised by an electrical conductivity that is homogeneous and isotropic and by the layer thickness. In the case of including IP parameters, there will be an additional three parameters per layer. The IP model used is a Cole-Cole model defined by:

$$\rho = \rho_0 \cdot \{1 - m \cdot [1 - 1/(1 + (s\tau)^c)]\}$$

where ρ is the layer resistivity; ρ_0 is the no-IP layer resistivity; m is the chargeability; s is the Laplace variable; τ is the time constant; and c is the power

The program has the option of calculating three different forward responses: principal step responses, principal impulse responses, both without system response modelling, plus the response after having been subjected to the influence of the instrument and electronics system: the system response.

The two transformations necessary to obtain a forward response - from frequency/Laplace domain to time domain and from wavenumber domain to space domain - are implemented as an inverse Laplace transform using the Gaver-Stehfest method (Knight and Raiche, 1982) and a Hankel transform using the Fast Hankel Transform method (Christensen, 1990; 1991; 1996), respectively

Responses can be calculated for a circular Tx loop and a polygonal Tx loop consisting of piecewise linear segments. For airborne TEM applications - instrument heights above $\approx 5\,\mathrm{m}$ - numerical experiments have shown that even though the Tx loop is polygonal, an equivalent circular Tx loop is an excellent approximation. In this context, *equivalent* means a Tx loop with the same area. Also, if an instrument with a polygonal Tx loop has a zero-coupled Rx, the code permits the calculation of the equivalent lateral Rx position that will make it zero-coupled for the approximate circular loop geometry. Calculation time for a polygonal Tx loop is slightly longer than for a circular loop.

Both central loop and offset loop configurations are accommodated.

The receiver (Rx) is always modelled as a dipole.

The system response

To be able to obtain a reliable inversion of TEM data, it is crucial that the effects of the instrument and the measuring strategy is adequately taken into account. Essentially, for a given delay time, all the elements of the system response can be expressed as convolutions with a principal step response. The elements of modelling the system response are:

- Repetition
- The low-pass filter characteristics of the Rx, including its electronics.
- The low-pass filter characteristics of the amplification and recording system.
- The convolution with the waveform.

Repetition

Most instruments apply a repeated, alternating waveform, and the measured responses in a given measurement cycle also comprise the responses from previous segments of the waveform. Especially for the late time part of the recording sequence, these contribution can be substantial, up to $\approx 15\,\%$. The code will model the repetition effect by adding the contributions from the previous three segments of the waveform. Numerical experiments have shown that this suffices to obtain a modelling error considerable smaller that the expected noise.

Low-pass filter characteristics

The low pass filter characteristics of the Rx system and the amplification and recording system are each modelled by a first- or second-order Butterworth filter applied as a functional factor in the calculation of the kernel function in the Laplace domain. This is the most direct and accurate way of including the filter effects.

Convolution with the waveform

The convolution with the waveform is done in the time domain by convolving the step response with the second derivative of the waveform. When the waveform is given as a piecewise linear function, its second derivative will consist of a series of delta functions at the waveform sampling points. Thereby the convolution with the waveform reduces to an interpolation of the value of the time domain response at the time differences defined by the delay time and the time of the delta functions - multiplied with the second derivative of the waveform at that sample.

As a rule of thumb, the modelling of the system response increases computation time with $\approx 50\%$ relative to the principal response.

The derivatives

Besides the forward response, the program has the option of delivering the derivatives of the response with regard to the model parameters and with regard to the Tx height. The derivatives are as numerically accurate as the forward response. They are obtained by differentiating through the recursion formulas for the forward response and applying the inverse Laplace transform and the Hankel transform, just like for the forward responses. In a way similar to the forward responses, the derivatives can optionally be subjected to the system response.

OUTPUT FROM THE CODE

The output from the code - responses and derivatives - are given as a time series from 10 ns to 10-100 ms (depending on the repetition frequency) sampled with a density of 10 per decade. For an airborne system, the ultra early times are not really physically valid; only samples with a delay time greater than the twice the instrument height divided by the speed of light are physically and conceptually valid. If e.g. Tx and Rx are both a 50 m height, only delay times greater than ≈ 333 ns can be used.

WHAT IS NOT MODELLED

To summarise: integration over the gates is not implemented in the code. Also, in the case where modelling of two moments is needed, the subroutine will have to be called twice. Remember also that the code assumes the QSA to be valid, so situations where values of the electrical permittivity is different from zero and/or where the value of the magnetic permeability is different from that of free space are not accommodated.

Besides these restrictions, it should be mentioned that the code does not perform any data processing at all. All of the data processing, noise reduction measures, and formulation of a noise model must be done in the calling code - or another program. The code does not (yet?) support the calculation of the horizontal component of the measured secondary magnetic field and it does not model the primary field. However, the code delivers output in a wide time range, and it is possible to model the measured field also during the turn-off time interval of the primary field.

CODE STRUCTURE AND CODE ELEMENTS

The files

The code is assembled in eight files:

TEMTEST.for
TEM1DPROG.for
TEM1DRESP.for
TEM1DRESPIP.for
TEM1DRESPPOLY.for
TEM1DRESPPOLYIP.for
TEM1DFHT.for
TEM1DFUNC.for

The individual routines contain a header that explains its function, but in addition, here is an overview of the routines of each file and their function.

<u>TEMTEST.for</u>
This is the driver program that can be used for tests of the TEM1DPROG subroutine. As such it is not a part of the program, but is included in the program package for user convenience.

<u>TEM1DPROG.for</u>
This is the main subroutine; the entry point for using the code. The parameters of the call define all the necessary information concerning the model, the instrument, the measuring configuration, and the type of fields to be calculated.

Most of the parameters of the call are transferred to COMMON blocks. A few initial calculations that are needed only once are included in the subroutine, e.g. some parameters used for polygonal Tx loop calculations.

<u>TERM1DRESP.for</u>
This routine calculates the fields from a circular Tx loop. The file also contains the functions FTEM and DFTEM that calculate the inverse Laplace transform of the kernel function of the response and of the derivatives, respectively.

<u>TEM1DRESPIP.for</u>
Same as TERM1DRESP.FOR, but for the case where IP parameters are included in the model. The file also contains the functions FTEMIP and DFTEMIP that calculate the inverse Laplace transform of the kernel function of the response and of the derivatives, respectively.

<u>TEM1DRESPPOLY.for</u>
Same as TERM1DRESP.FOR, but for the case of a polygonal Tx loop. The file also contains the routine used in the integration over the linear segments of the Tx loop.

TEM1DRESPPOLYIP.for
Same as TEM1DRESPPOLY.FOR, but for the case where IP parameters are included in the model.

TEM1DFHT.for
The routines of this file calculate the filter coefficients needed and perform the Fast Hankel Transform as a convolution between kernel function values and the filter coefficients (Christensen, 1990; 1991; 1996). The first time the routine is called, Hankel filters for J0 and J1 transforms are calculated and stored for later use.

TEM1DFUNC.for
This is a collection of smaller subroutines and functions that are called from other routines.

OTHER PROGRAM ELEMENTS: COMMON BLOCK PARAMETERS

As mentioned above, the FORTRAN code makes use of COMMON blocks. Each of these are defined in an INCLUDE file which is loaded into the routines where the parameters are needed.

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

All arrays of integers are declared as INTEGER*4 and all of the real arrays as REAL*8. Also, all routines contain an IMPLICIT statement defining default INTEGERS and REALS:

```
IMPLICIT INTEGER*4 (I-N)
IMPLICIT REAL*8 (A-H,O-Z)
```

Below you will find an explanation of the parameters in each of the COMMON blocks. This also serves as an explanation of the parameters used in the call of TEM1DPROG.

¤¤¤

```
C-----------------------------------------------------------------------
C --- PARAMETER STATEMENT FOR DIMENSIONING THE ARRAYS OF THE PROGRAM
C-----------------------------------------------------------------------
      PARAMETER (N_ARR=128,N_ARR4=512)
C ----------
C   N_ARR   =  128   General  dimensioning of matrices and vectors
C   N_ARR4  =  512   General  dimensioning of matrices and vectors
C ----------
```

¤¤¤

```
¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

C-----------------------------------------------------------------------
C --- THE INSTRUMENT COMMON BLOCK
C-----------------------------------------------------------------------
      INTEGER*4 ICEN, IZEROPOS
      REAL*8 TXAREA, TXRAD, HTX, HRX, RTXRX, EQRXPOS
      COMMON /INSTRUMENTBL/ ICEN,IZEROPOS,TXAREA,TXRAD,
     #                      HTX,HRX,RTXRX,EQRXPOS
C -----------
C   ICEN:     [0 | 1] : [Offset loop | Central loop] configuration
C   IZEROPOS: [1 | 0] : [Rx is zero coupled to Tx | Not zero coupled]
C   TXAREA:   Area of the Tx.
C   TXRAD:    Radius of a an equivalent circular Tx loop
C             (Calculated if IZEROPOS = 1).
C   HTX:      Tx height.
C   HRX:      Rx height.
C   RTXRX:    Horizontal distance between Tx centre and Rx dipole.
C   EQRXPOS:  Equivalent value of RTXRX for zero-coupled Rx.
C ----------

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

C------------------------------------------------------------------------
C --- THE MODEL COMMON BLOCK
C------------------------------------------------------------------------
      INTEGER*4 IMLM,NLAY
      REAL*8 RHON(N_ARR),DEPN(N_ARR),SIGN(N_ARR),THKN(N_ARR)
      COMMON /MODELBL/ IMLM, NLAY, RHON, DEPN, SIGN, THKN
C ----------
C   IMLM:   IMLM = [ 0 | 1 ] means [ Few-layer model | Multi-layer model]
C   NLAY:   Number of layers in the model.
C   RHON:   Layer resistivities.
C   SIGN:   Layer conductivities.
C   THKN:   Layer thicknesses.
C   DEPN:   Depths to top of layer boundaries, DEPN(1)=0.
C ----------

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

C------------------------------------------------------------------------
C --- THE RESPONSE COMMON BLOCK
C------------------------------------------------------------------------
      INTEGER*4 IRESPTYPE,IDERIV,IRXFILT,IAMPFILT,IREP,IWCONV
      REAL*8 REPFREQ, RXFREQ, AMPFREQ
C ----------
      COMMON /RESPBL/ IRESPTYPE,IDERIV,IRXFILT,IAMPFILT,IREP,IWCONV,
     # REPFREQ, RXFREQ, AMPFREQ
C ----------
C   IRESPTYPE:  [0 | 1 | 2] :
C               [Step response, no repetition or waveform convolution,
C                optional Rx and Amp filters.
C                | Impulse, no repetition or no waveform convolution,
C                optional Rx and Amp filters.
C                | Responses with selected elements of the system response].
C   IDERIV:     [1 | 0] : [Calculate derivatives | No derivatives]
C   ICEN:       [0 | 1] : [Offset loop | Central loop] configuration]
C               The value of ICEN is automatically determined in the program.
C   IRXFILT:    [0 | 1 | 2] : [No Rx filter | First order Rx filter |
C                Second order Rx filter].
C   IAMPFILT:   [0 | 1 | 2] : Same as for IRXFILT, but for Amplifier system.
C   IREP:       [1 | 0] : [Do | Do not] model repetition
C   IWCONV:     [1 | 0] : [Do | Do not] convolve with waveform.
C ----------

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤
```

```
¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

C------------------------------------------------------------------------
C --- THE WAVEFORM COMMON BLOCK
C------------------------------------------------------------------------
      INTEGER*4 NWAVE
      REAL*8 TWAVE(N_ARR),AWAVE(N_ARR),SLOPE(N_ARR),D2WDT2(N_ARR)
C ----------
      COMMON /WAVEBL/ NWAVE,TWAVE,AWAVE,D2WDT2
C ----------
C   NWAVE:             Number of samples in the piecewise linear waveform
C   TWAVE(1:NWAVE):    Sample times of the piecewise linear waveform.
C   AWAVE(1:NWAVE):    Sample amplitude values of the piecewise linear waveform.
C   SLOPE (1:NWAVE-1): The 1st derivative of the waveform.
C   D2WDT2(1:NWAVE):   The 2nd derivative of the waveform.
C ----------

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

C------------------------------------------------------------------------
C --- THE IP COMMON BLOCK
C------------------------------------------------------------------------
      INTEGER*4 IMODIP
      REAL*8 CHAIP(N_ARR),TAUIP(N_ARR),POWIP(N_ARR)
      COMMON /IPBL/ IMODIP, CHAIP, TAUIP, POWIP
C ----------
C   IMODIP:  IMODIP = [ 1 | 0 ] : [ Model IP effect | Do not model IP effect].
C   CHAIP:   The chargeability of the Cole-Cole model for each layer.
C   TAUIP:   The time constant of the Cole-Cole model for each layer.
C   POWIP:   The power of the Cole-Cole model for each layer.
C ----------

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

C--------------------------------------------------------------------------
C --- THE POLYGONAL TX LOOP COMMON BLOCK
C--------------------------------------------------------------------------
      INTEGER*4 NPOLY,NRSAMP,NPSAMP(N_ARR4)
      REAL*8 XPOLY(N_ARR),YPOLY(N_ARR),X0RX,Y0RX,Z0RX,DIPOLEFAC
      REAL*8 DS(N_ARR4),YSAMP(N_ARR4),RSAMP(N_ARR4),RSAMPL(N_ARR4)
      REAL*8 RSAMPMIN,RSAMPMAX,AREAPOLY,PERIMETERPOLY
      COMMON /POLYGONBL/ NPOLY,NRSAMP,NPSAMP,XPOLY,YPOLY,X0RX,Y0RX,ZORX,
     #   DS,YSAMP,RSAMP,RSAMPL,RSAMPMIN,RSAMPMAX,AREAPOLY,PERIMETERPOLY
C ----------
      PARAMETER (DIPOLEFAC = 0.10D0)
C ----------
C  NPOLY:         Number of linear segments of the polygonal Tx coil.
C                 NPOLY=0: Circular Tx coil.
C                 For NPOLY=0: Circular Tx coil, and IP parameters are moot.
C  XPOLY:         X-coordinates of the polygonal Tx coil apices.
C  YPOLY:         Y-coordinates of the polygonal Tx coil apices.
C  X0RX:          The X-coordinate of the Rx dipole.
C  Y0RX:          The Y-coordinates of the Rx dipole.
C  NRSAMP:        Total number of sampling points on the Tx loop perimeter.
C  NPSAMP:        Number of sampling points on each of the Tx loop segments.
C  YSAMP:         Y-coordinate of the Rx position in the coordinate system
C                   of an X-directed electric dipole.
C  RSAMP:         Radial distance to the Rx position in the coordinate system
C                   of an X-directed electric dipole.
C  RSAMPL:        log(RSAMP).
C  DS:            Length of subdivision interval for each side.
C  RSAMPMIN:      Minimum distance for the FHT routine.
C  RSAMPMAX:      Maximum distance for the FHT routine.
C  AREAPOLY:      Area of the Tx loop
C  PERIMETERPOLY: Length of the perimeter of the Tx loop.
C  DIPOLEFAC:     The subdivisions on each loop segment is smaller than
C                 DIPOLRFAC times minimum distance from the Rx to the loop segment.
C ----------

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤
```

The code also contains DATA statements, mostly for basic physical constants such as π and μ_0.

CALCULATION TIMES

Naturally, computation times depend strongly on the platform used and on the efficacy of the interface between the calling program and the FORTRAN .dll resulting from the compilation of the source code. The times listed below are for a standard laptop, one thread calculation, and should be taken as rules of thumb.

In all of the examples below, an offset loop configuration is used. The model is a 6-layer model, so there are a total of 11 model parameter derivatives plus the derivative with regard to Tx height. However, that latter is obtained almost for free in connection with the forward response through a *related transform* approach. The abbreviations in the list are:

Circ	for a circular Tx loop,
Poly	for a polygonal Tx loop,
Deriv	for including derivatives in the calculations,
IP	for including IP parameters in the modelling of responses and derivatives.

Principal response, Circ:	5 ms
Convolved response, Circ:	7 ms
Convolved response, Circ Deriv:	9 ms
Convolved response, Circ IP:	11 ms
Convolved response, Circ Deriv IP:	13 ms
Convolved response, Poly:	8 ms
Convolved response, Poly Deriv:	10 ms
Convolved response, Poly IP:	12 ms
Convolved response, Poly Deriv IP:	15 ms

A few tentative conclusions can be drawn from the table. For a circular Tx loop, including the system response adds $\approx$ 40% to the computation time. Derivatives add $\approx$ 30% to the computation time. Including IP adds $\approx$ 6 ms, i.e. doubles the computation time. Including IP and derivatives adds $\approx$ 8 ms, a factor of $\approx$ 2.5 times.

For a principal response, using a polygonal Tx loop (4 sides) adds $\approx$ 60% relative to the circular Tx loop. Looking at convolved responses we find for a polygonal loop, relative to a principal response: Derivatives (11 of them) add $\approx$ 25% to the computation time. Including IP parameters adds $\approx$ 50% to the computation time. Including IP parameters plus derivatives doubles the computation time.

ACCURACY TEST - COMPARING WITH ANALYTIC SOLUTIONS

There are few TEM responses that can be expressed analytically. In this section we shall use two of them to investigate the accuracy of the program, both of them with Tx and Rx on the surface of a homogeneous halfspace of 30 Ωm. The first is a central loop configuration with circular Tx loop with a radius of 10 m, and the Rx at its centre. The second is an offset vertical magnetic dipole-dipole configuration with 15 m Tx-Rx separation and with the Tx modelled as a small loop with radius 1 m. In both cases, the magnetic moment is set to unity. In Figure 1, the results of the modelling are compared with the analytic expressions given in Ward and Hohman (1987; equations 4.97 for central loop; and 4.69a for offset loop):

$$B_z = \frac{\mu_0 I}{2a}\left[\frac{3}{\sqrt{\pi}\,(\theta a)}\cdot \exp\left[-(\theta a)^2\right] + \left(1-\frac{3}{2\,(\theta a)^2}\right)\cdot \operatorname{erf}(\theta a)\right] \text{ , Central loop}$$

$$B_z = \frac{\mu_0 m}{4\pi\rho^3}\left[\left(\frac{9}{2\,(\theta r)^2} - 1\right)\mathrm{erf}\,(\theta r) - \frac{1}{\sqrt{\pi}}\left(\frac{9}{(\theta r)} + 4(\theta r)\right)\exp\big[-(\theta r)^2\big]\right] \quad , \text{ Offset loop}$$

where $\theta = \sqrt{\mu_0\sigma/(4t)}$, t is the delay time, and σ is the halfspace conductivity; a is the radius of the central loop Tx; r is the Tx-Rx distance of the dipole-dipole offset configuration; and m is the Tx moment. For the central loop configuration, we have, in terms of the moment:

$$\frac{I}{2a} = \frac{I \cdot \pi a^2}{2\pi a^3} = \frac{m}{2\pi a^3}$$

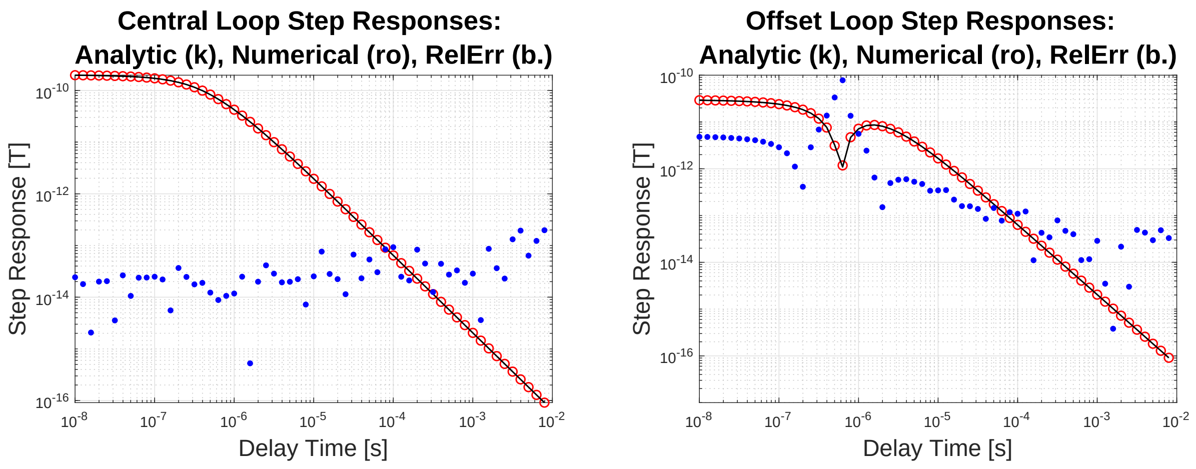

Figure 1: Comparison between numerical (red dots) and analytic (black full line) responses for a central loop configuration (left) and an offset dipole-dipole configuration (right). The blue dots indicate the relative error of the numerical calculations multiplied with a factor of 10^{-9}.

Figure 1 shows that the relative error for the central loop configuration is of the order of 10^{-5} throughout the delay time interval from 10 ns to 10 ms. For the offset configuration, the relative error is higher, especially around the delay time 0.6 μs where the response changes sign and where the relative error is of the order of 10^{-3}. However, for late times the relative error is of the same order of magnitude as for the central loop configuration.

REFERENCES

Christensen N.B. 1990. Optimized Fast Hankel Transform Filters. Geophysical Prospecting 38, 545-568.

Christensen N.B. 1991. Optimized Fast Hankel Transform Filters - reply. Geophysical Prospecting, 39, 449-450. DOI: 10.1111/j.1365-2478.1991.tb00322.x

Christensen N.B. 1996. The Fast Hankel Transform - Comment. Geophysical Prospecting, 44, 469-471. DOI: 10.1111/j.1365-2478.1996.tb00158.x

Christensen N.B. 1997. Electromagnetic subsurface imaging - A case for an adaptive Born approximation. Surveys in Geophysics 18, 477-510.

Christensen N.B. 2002. A generic 1-D imaging method for transient electromagnetic data. Geophysics, 67, 438-447.

Christensen N.B., Reid J.E. and Halkjær M. 2009. Fast, laterally smooth inversion of airborne time-domain electromagnetic data. Near Surface Geophysics, 7, 599-612. DOI: 10.3997/1873-0604.2009047.

Christensen N.B. and Lawrie K. 2012. Resolution analyses for selecting an appropriate airborne electromagnetic (AEM) system. Exploration Geophysics, 43, 213-227. http://dx.doi.org/10.1071/EG12005

[Smiarowski A. and Mulè S. 2014. Comments on: Christensen, N., and Lawrie, K., 2012. Resolution analyses for selecting an appropriate airborne electromagnetic (AEM) system, Exploration Geophysics, 43, 213-227.
http://dx.doi.org/10.1071/EG13091]

Christensen N. and Lawrie K. 2014. Response to comments by Adam Smiarowski and Shane Mulè on: Christensen, N., and Lawrie, K., 2012. Resolution analyses for selecting an appropriate airborne electromagnetic (AEM) system, Exploration Geophysics, 43, 213-227. http://dx.doi.org/10.1071/EG14015

Christensen N.B. 2014. Sensitivity functions of transient electromagnetic methods, Geophysics, 79, E167-E182. doi: 10.1190/geo2013-0364.1

Christensen N.B. 2016. Fast approximate 1D modelling and inversion of transient electromagnetic data. Geophysical Prospecting, 64, 1620-1631.
https://doi.org/10.1111/1365-2478.12373

Knight J.H. and Raiche A.P. 1982. Transient electromagnetic calculations using the Gaver-Stehfest inverse Laplace transform method. Geophysics 47, 47-50.

Ward S.H. and Hohmann G.W. 1987. Electromagnetic theory for geophysical applications. In: Investigations in Geophysics 3: Electromagnetic Methods in Applied Geophysics (ed. M.N. Nabighian), pp. 131-311., Society of Exploration Geophysicists.

¤¤