Code Description

Author: You Xiran

Organization: Department of Geophysics, Chang'an University

Email: chdyxr@163.com

Date: May 2024

Introduction

All codes are for submissions to Computers and Geosciences journals, with the corresponding paper title being “Fast forward modeling of grounded electrical source transient electrical based on inverse Laplace transform adaptive hybrid algorithm”.

The code is written in Matlab and runs in version 2019b or higher. Its functions include 1D grounded electrical source TEM forward modeling, comparing the response of various time-frequency conversion methods, and comparing the calculation time of algorithms.

Table 1 Code List

Types	Code name
Main programs	SATEM_For.m OptimizationAlgorithm.m compare_time.m
Subprograms	loadsinhank.m calCm_GS.m calBm_Euler.m calDeltam_Talbot.m gethks.m calrte.m calF.m forword3.m IntGauss_Legendre.m Return_jx_Hz_t.m Return_sum_jx_field_t.m

Please refer to the following text for a detailed explanation and usage of the code.

Program Description

1.Main program

(1) SATEM_For.m

Function: 1D grounded electrical source TEM forward modeling and comparing the response of different frequency-time conversion algorithms. Comparison of analytical solutions for the half-Space model and layered models.

Usage: Firstly, set the parameters required for forward modeling calculation, and the meanings of each parameter can be found in the comments. Then adjust the corresponding switch flag to control the time-frequency conversion method used in the calculation and the physical quantity of the output, and also control whether to consider the induced polarization effect. In the subsequent For-loop, the number of coefficients in the time-frequency conversion algorithm is controlled by M to calculate and plot the response as the coefficients vary. It should be noted that if it is not necessary to draw the half-space response curve multiple times, please comment out this section of code within the loop. It is recommended to run the program section by section.

(2) OptimizationAlgorithm.m

Function: This program uses adaptive hybrid algorithms for 1D grounded electrical sources TEM forward modeling and error analysis.

Usage: In terms of parameter settings, we won't go into detail here. It should be noted that the hybrid algorithm needs to calculate the coefficients of both the G-S algorithm and Talbot algorithm simultaneously, and the number of coefficients can be adjusted at the corresponding function. It is recommended to run the program section by section.

(3) compare_time.m

Function: Use the specified frequency-time conversion algorithm for forward modeling calculation, take the average time of multiple calculations as the algorithm calculation time, and finally output the time distribution and maximum value.

Usage: First, set various parameters. Then, within the For-loop, set the number of calculations (tt's range) and the number of coefficients (M's range). Finally, execute this section of code.

2.Subprogram

(1) loadsinhank.m

Function: Load various filtering coefficients and set the parameters. It is worth noting that this function returns multiple filtering coefficients simultaneously, facilitating direct utilization in subsequent programs. The input M of the function is the number of coefficients for the inverse Laplacian algorithm, and when using such algorithms, it is necessary to provide a positive integer.

(2) calCm_GS.m

Function: Calculate the corresponding G-S algorithm filtering coefficients based on the number of input coefficients M. M must be an even positive integer.

(3) calBm_Euler.m

Function: Calculate the corresponding Euler algorithm filtering coefficients based on the number of input coefficients M. M must be a positive integer.

(4) calDeltam_Talbot.m

Function: Calculate the corresponding Talbot algorithm filtering coefficients based on the number of input coefficients M. M must be a positive integer.

(5) gethks.m

Function: Calculate the corresponding G-S algorithm filtering coefficients based on the number of input coefficients M.

(6) calrte.m

Function: Determine the independent variables for frequency-time conversion and Hankel transformation. It should be noted that when using the Sine transform, the expression for the variable sw needs to be adjusted based on the specific filtering coefficients. Please refer to the annotations for further details.

(7) calF.m

Function: Calculate the frequency domain response and transition to the time domain.

(8) forword3.m

Function: Calculate the time domain response of a single electric dipole.

(9) IntGauss_Legendre.m

Function: The Gauss-Legendre quadrature formula integrates the length ds of an electric dipole along the direction of the wire.

(10) Return_jx_Hz_t.m

Function: Calculate the analytical solution response in the half space time domain.

(11) Return_sum_jx_field_t.m

Function: Obtain the superposition response of the electric dipole field through the integration method.

Additional files

The remaining files in the library are text files that store multiple filtering coefficients. The corresponding information is as follows：

Table 2 Text file list

File name	Content
hkf.txt	140 points J1 Hankel filtering coefficients
sin_xs_123.txt	123 point sine filtering coefficients
sinf.txt	160 point sine filtering coefficients
sin_250.txt	250 point sine filtering coefficients
cos_250.txt	123 point cosine filtering coefficients
Guptasarma.txt	Sampling points and coefficients of Guptasarma algorithm

The above files are automatically loaded by functions, just place them in the same folder as the relevant programs.