User Guide for the DD Solver

Luis Tao[1] **and Fabio Zyserman**[1]

[1]**Universitat Politècnica de Catalunya**
[2]**Centro Científico Tecnológico Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), La Plata, Argentina**

ABSTRACT

This document provides a concise user guide for the DD-POD solver. It describes the solver's purpose, required libraries, input structure, and expected outputs. The goal is to help new users compile, run, and analyze results.

Keywords: Domain Decomposition, Model Reduction, Magnetotellurics, Fortran, Scientific Computing

1. OVERVIEW OF THE DD SOLVER

The DD (Domain Decomposition) solver is a parallel Fortran-based program designed to solve large-scale magnetotelluric (MT) forward problems. It is based on the domain-decomposition formulation of Zyserman and Santos (2000). The solver divides the global computational domain into several subdomains, which are solved independently and coupled iteratively. In its current implementation, the domain decomposition is applied only in the horizontal direction, meaning that subdomains share horizontal interfaces. In addition to the full-order formulation, the DD solver can utilize Reduced Basis-Proper Orthogonal Decomposition (RB-POD) to accelerate computation. When precomputed snapshots are available, this configuration is referred to as the DD-POD solver. The main inputs are the conductivity model, mesh discretization, and frequency list, while the outputs include electromagnetic field components, apparent resistivity, and phase for both polarizations as a function of frequency.

2. REQUIRED LIBRARIES

To compile and execute the DD solver, the following libraries are required:

- **MUMPS** – for parallel sparse direct matrix factorization and linear system solving.
- **MPI** – for inter-process communication between subdomains.
- **Intel MKL (optional)** – for optimized BLAS/LAPACK routines.

3. INPUTS

The DD solver requires several input files and parameters defining the physical model and numerical setup.

3.1 Model and Solver Configuration

Model and solver parameters are specified in the file `sizes.f90`, which controls both the domain partitioning. The key parameters are:

- `nsy, nsz`: Number of subdomains in each direction. In the current version, `nsy` is always set to 1.
- **nprocs_input**: Total number of processors used. Two processors are required per subdomain (one for each polarization); thus, this value must be $2 \times \texttt{nsy} \times \texttt{nsz}$.
- `ngx, ngy, ngz`: Number of elements along the x, y, and z directions, respectively. The size `ngz` must be a multiple of `nsz`.

Model Files

The input mesh file, `input_mesh_MT`, defines the spatial discretization and frequency sampling. The sum of the partitions between consecutive nodes in each direction must match the total number of partitions specified. The file format is as follows:

```
(BEGIN)
Nfrequencies
Frequency_1
Frequency_2
...
Frequency_Nfrequencies
"TE"
Npartitions in x
x_node_1 (m)
x_node_2 (m)
Npartitions in x direction between node_1 and node_2
...
x_node_{ngx} (m)
x_node_{ngx+1} (m)
Npartitions between node_{ngx} and node_{ngx+1}
Npartitions in y
y_node_1 (m)
y_node_2 (m)
Npartitions in y direction between node_1 and node_2
...
y_node_{ngy} (m)
y_node_{ngy+1} (m)
Npartitions between node_{ngy} and node_{ngy+1}
Npartitions in z
z_node_1 (m)
z_node_2 (m)
Npartitions in z direction between node_1 and node_2
...
z_node_{ngz} (m)
z_node_{ngz+1} (m)
Npartitions between node_{ngz} and node_{ngz+1}
(END)
```

The file `conduc_input` defines the conductivity distribution of each element and the layered background model:

```
(BEGIN)
DO l = 1:ngz
    DO k = 1:ngy
        DO j = 1:ngx
            Conductivity of element (j,k,l) (S/m)
        ENDDO
    ENDDO
ENDDO
thickness_layer_1 (air) (m)
thickness_layer_2
...
thickness_layer_{n-1}
conductivity_layer_1 (air)
conductivity_layer_2
...
```

```
conductivity_layer_n
(END)
```

Note: Air layers must be explicitly included in the model. The default configuration assumes six horizontally homogeneous background layers, with the last (bottom) layer representing a half-space, hence no thickness value is provided for that layer. It is recommended only using 1 air layer with the same thickness as the shallower Earth layer.

3.2 Hyperparameters and Physical Constants

These parameters are defined in `Mod_Constants.f90`. They control either the full-order or POD-based solver modes.

For Reduced-Order Simulations:

- `use_pod`: Set logical flag to (`TRUE`)
- `Nsnapshots`: Number of snapshots in the snapshot matrix.
- `tol_svd`: SVD Truncation tolerance.
- `pod_path`: Directory containing the snapshot matrices and the location where the reduced bases are written. Snapshots are stored as a single column vector with $N_{\text{dof}} \times N_{\text{snapshots}}$ rows. Files follow the naming convention `pod_path/sol_`*modo_blockid*`.txt`, where *modo* is either "TE" or "TM", and *blockid* denotes the subdomain identifier.

For Full-Order and Reduced-Order simulations:

- `use_pod`: Set logical flag to (`TRUE`) or (`FALSE`)
- `relx`: Relaxation parameter in the domain-decomposition scheme (value between 0 and 1).
- `penal`: Penalization factor applied to β, influencing convergence rate.
- `maxiter`: Maximum number of iterations allowed.
- `convergence_tol`: Convergence threshold controlling iteration termination.

4. OUTPUTS

The solver produces a set of output files containing physical quantities and performance information.

4.1 Quantities of Interest

The primary outputs include:

- Electric and magnetic field components evaluated at the midpoints of surface elements.
- Apparent resistivity and phase for each polarization, computed at the midpoints of the surface mesh and stored in the `outputs` directory.

4.2 Performance and Timing

Performance logs report timing information for each computational phase and the total number of iterations, are also obtainable in the `outputs` directory.

5. EXAMPLE RUN

Before running the solver, please review the `Makefile` to ensure that the compilation and linking settings are correctly configured for your system.

To execute a simulation with parameters `nsy = 1` and `nsz = 3`, use the following command:

```
make
mpirun -np 6 ./max3d-par
```

Ensure that all required input files (`conduc`, `input_mesh_MT`, etc.) are located in the working directory prior to execution.

ACKNOWLEDGMENTS

The development of this was carried out as part of the Marie Skłodowska-Curie EarthSafe project funded by the European Union's Horizon Europe programme under grant agreement No. 101120556. The author gratefully acknowledges Professor Fabio Zyserman for developing the original code that served as the foundation for this project, and Dr. Constanza Manassero, who developed two of the routines used for assembling the stiffness matrix and generously shared her original code, thereby making this work possible.

REFERENCES

Zyserman, F. I. and Santos, J. E. (2000). Parallel finite element algorithm with domain decomposition for three-dimensional magnetotelluric modelling. *Journal of Applied Geophysics*, 44(4):337–351.