**Prerequisites for code to run:**

To compile the 3D\_TBSBIEM code, your system must meet several prerequisites, primarily related to the OpenMP for CUP parallelization, NVIDIA HPC SDK (which includes nvfortran) and GPU support.

**OpenMP Support:** Check for support for OpenMP. The -fopenmp flag enables OpenMP parallelization. Installation and documentation for openMP can be found in following website: <https://www.openmp.org/resources/refguides/>

**NVIDIA HPC SDK:** To enable compiler support, install the NVIDIA HPC SDK. The code was tested using the nvfortran compiler. The following webpage has installation and usage instructions: <https://docs.nvidia.com/hpc-sdk/index.html>

**CUDA Toolkit:** Install cuda toolkit, for the use of GPU acceleration using OpenACC directives. The -acc flag of openACC enables directive for GPU acceleration using OpenACC. The CUDA Toolkit is also suggested for the cuFFT used in the code, which use cuda programming in the Fortran language to generate discrete fast Fourier Transforms and their inverse with GPU acceleration. The -lcufft flag enables cuFFT.

**Input files:** The input file is given for TPV5, TPV6, and TPV7, as TPV5\_Data\_Input.in, TPV6\_Data\_Input.in, and TPV7\_Data\_Input.in, respectively.

If TPV5 benchmark problem need to be tested, on can change the ‘TPV5\_Data\_Input.in’ file to ‘Data\_Input.in’ and compile the batch file. The same process is valid for TPV6 and TPV7.

Create two folder ‘./Kernels’ and ‘./dats’ before compilation of the code, as the kernels calculated during the simulation are stored in the ‘./Kernels’ folder and the output results are stored in the folder ‘./dats’.

Further, if the kernels for numerical simulation are calculated ahead of simulation and stored in ‘./Kernels’ folder, then you need to give value as ‘0’ at the end of ‘Data\_Input.in’ file. If the kernels are to be calculated during the simulation, you need to give value as ‘0’ at the end of ‘Data\_Input.in’ file.

**Compilation:** Compilation of 3d\_TBSBIEM is handled by one batch script file batch.sh. This batch script enables SLURM based management of number of CPUs, GPUs, and RAM allocation for the job in NVIDIA HPCs. However, running the batch file in smaller systems need not require the SLURM batch script part in the batch file. The following command is used for the compilation.

“nvfortran -fopenmp -fast -g -Minfo=accel -acc -gpu=managed -O3 -o main ./src/\*.cuf -lcuff ”

During the compilation, the temp files ‘.o’ and ‘.mod’ are generated in ‘./src’ then they are deleted after execution file ‘./main’ is created.

Than the batch file includes the ‘./main’ for the execution of the code.

**Output files:** The output files are generated in ‘./dats’ folder. The slip data for contour plot of rupture front are created with ‘Out%4i.dat’ files, where ‘4i’ indicates integer numbering with four digits. The data at several station point are created in ‘Out%2i.dat’ files.

**Post processing output files:** One can follow the matlab codes given in ‘Solved\_Examples’ folder to post process the results. It can be done by transferring the result files in ‘./dats’ to ‘./Results/ResultsData\_12Sec’ in any TPV problem folder. The ‘.png’ pictures are stored in ‘./Figs’ older of TPV problem in ‘Solved\_Examples’ folder.