Manual for CGFDM

Zijia Wang 12231211@mail.sustech.edu.cn

Wenqiang Zhang zwq@stanford.edu

2024.09

Contents

1. Folder Structure

doc: folder contains the manual for this program
bin: folder contains the binary executable files
srcd: folder contains all CUDA source files
obj: folder contains intermediate object files during compiling
Makefile: file for automatically building programs using GNU make
matlab: folder contains Matlab scripts for pre-processing and post-processing
jobs: folder contains examples for several dynamic rupture modelling examples

2. Build the program

2.1 Compiling environment

The operating environment used by the author to develop the program is:

cuda9.2, cuda10.0 or cuda10.1
openmpi (gcc-4.8.5,with cuda)
netcdf 4.4.4

Some parameters in Makefile may need to be modified if you want to build in other environments.

CUDAHOME: The directory of cuda, you may need to modify it according to your environment.
MPIHOME: The directory of openmpi, you may need to modify it according to your environment.
NETCDFHOME: The directory of netcdf, you may need to modify it according to your environment.

2.2 Optional parameters in Makefile

There are some optional parameters in Makefile. Generally, we don't recommend changing those about numerical scheme. Here are some options that may need to be changed when calculating different models:

DoublePrecision: We highly recommend using double precision for rate- and state-dependent friction law. For slip-weakening friction law, you can close this option to increase efficiency.

FreeSurface: If you want to simulate the full-space case, close this option.

Barrier: If you want to simulate a dynamic rupture model with an unbreakable barrier, open this option. But only 1 barrier is allowed in this version of the code.

TPV22/TPV23: If you want to simulate the benchmark model tpv22 or 23, this option must be opened because it controls the calculation of the nucleation process.

FaultSmooth: We suggest open this option if you want to simulate multi-faults. This will incrase the simulation precision of normal stress in overlap region of multi-faults.

Reminder: When you change these options, remember to “make clean” first, and then recompile the code.

3. Running example

3.1 Coordinate and unit

Coordinate conversion:

Y-axis is the fault-strike direction

Z-axis is the up-dip direction (vertical upward)

X-axis if the fault-normal direction

Unit:

Variable	Coord	Vp, Vs	density	stress	Slip rate	time	Displacement
Unit	m	m/s	kg/m^3	Pa	m/s	sec	m

3.2 Parameter configuration file "params.json"

The params.json file is a parameter file read by the program, and the main parameters are explained as follows:

TMAX:

Total simulation time (in seconds)

DT:

Time step (in seconds)

DH:

Space step (in meters)

NX/NY/NZ:

Number of grids in x/y/z direction

PX/PY/PZ:

Number of MPI divisions in x/y/z direction. **Note that NX/NY/NZ must be divisible by PX/PY/PZ, and PX must be 1.** PX*PY*PZ is the number of GPU cards to be used.

INPORT_GRID_TYPE:

Enter the type of the grid, where 0 is configured in the code, 1 and 2 are input by the user. If it is 1, read the 2D fault grid data file specified by the "Fault_geometry". The program will automatically build a 3D grid based on your 2D grid input. If it is 2, read the 3D fault grid data file specified by the "Fault_geometry". That is, if you want to consider the topography or multi-faults, you should set this keyword as 2.

INPORT_STRESS_TYPE:

Enter the type of initial stress, where 0 is configured in the code, 1 is input by the user, and if it is 1, read the data file specified by the "Fault_init_stress".

Friction_type:

Friction law type, 0 for slip-weakening law; 1 for rate- and state-dependent (RSF) law with ageing law; 2 for RSF law with strong rate weakening (slip law consider flash heating).

EXPORT_WAVE_SLICE_X:

Wavefield y-z plane slice

EXPORT_WAVE_SLICE_Y:

Wavefield x-z plane slice

EXPORT_WAVE_SLICE_Z:

Wavefield x-y plane slice, set to NZ-1 to output the wavefield of the free surface

igpu:

Specifies to run on a GPU card numbered igpu. This is valid only when a single card is running

Fault_geometry:

The fault geometry file to be input (nc format).

Fault_init_stress:

The initial stress file to be input (nc format).

num_fault:

The number of faults.

src_i:

The coordinate point number of each fault in X direction (the point number start from 1).

Fault_grid:

Fault rupturable region (the point number start from 1).

Barrier_grid:

The location of barrier (the point number start from 1). Only valid when "Barrier" is opened in Makefile.

Asp_grid:

Initial stress asperity area. If INPORT_STRESS_TYPE=1, it is invalid.

mu_s, mu_d, Dc, C0:

Parameters of slip-weakening law, If INPORT_STRESS_TYPE=1, it is invalid.

RS_V0, RS_Vini, RS_f0, RS_fw:

Parameters of rate-and-state friction law, If INPORT_STRESS_TYPE=1, it is invalid.

PML_xxx:

PML absorption boundary related parameters, we don't need to modify them.

vp1, vs1, rho1:

Media parameters. If Media1D or Media3D is set, it is invalid.

bi_vp1,bi_vs1,bi_rho1,bi_vp2,bi_vs2,bi_rho2:

Media parameters of bimaterial fault (you can keep the same with vp1, vs1 and rho1 to make it invalid)

OUT:

Output folder

smooth_load_T:

Time for smooth loading of stress perturbation; if it is less than or equal to 0, it will not be smoothly loaded (stress perturbation is instantaneously loaded at moment 0 and remains unchanged).

Additional remarks:

If the MATLAB you are using is a higher version than 2016, get_params.m can't be used normally. You can visit https://github.com/fangq/jsonlab to download jsonlab-master package and add it to the MATLAB path. Then change "par = jsondecode(textread(parfile,'%c'));" in the get_params.m script to "par = loadjson(parfile);".

3.2 Array dimensions of fault geometry data

The fault geometry data file (grid file) is a netcdf file. Here we need to declare that because the fast axis of the program is the Y axis, if you want to input a 3D grid, the array dimensions of x, y and z in the nc file should be (nx, nz, ny). If you want to input a 2D grid, the array dimensions of x, y and z in the nc file should be (nz, ny).

3.3 Initial stress data file structure

The stress configuration file is a netcdf file. Assuming that the file is init_stress.nc, the result of ncdump –h init_stress.nc is:

```
netcdf init_stress {
dimensions:
    ny = 800 ;
    nz = 400 ;
variables:
    float x(nz, ny) ;
    float y(nz, ny) ;
    float z(nz, ny) ;
    float Tx(nz, ny) ;
    float Ty(nz, ny) ;
    float Tz(nz, ny) ;
    float dTx(nz, ny) ;
    float dTy(nz, ny) ;
    float dTz(nz, ny) ;
    float mu_s(nz, ny) ;
    float mu_d(nz, ny) ;
    float Dc(nz, ny) ;
    float C0(nz, ny) ;
    float a(nz, ny) ;
    float b(nz, ny) ;
    float L(nz, ny) ;
```

```
    float Vw(nz, ny) ;
    float State(nz, ny) ;
}
```

Where Tx, Ty, Tz are the components of the initial fault traction in the (x, y, z) coordinate system, and dTx, dTy, dTz are the perturbation of the initial fault traction $\Delta\tau_i$, that is, the shear tractions in nucleation zone satisfy:

$$\tau_i^0(t) = \tau_i^0 + \Delta\tau_i \cdot G(t), (i = x, y, z)$$

where

$$G(t) = \begin{cases} \exp\left[\frac{(t-T)^2}{t(t-2T)}\right], 0 < t < T \\ 1, \quad t \geq T \end{cases}$$

The loading time T is specified by the keyword smooth_load_T in params.json. This nucleation strategy usually used in RSF law. Just set them all to 0 for slip-weakening law.

Note: mu_s, mu_d, Dc and C0 are friction parameters of slip-weakening law. A, b, L, Vw and State are parameters for RSF law. We just need to ensure that the parameters corresponding to the friction law you choose are stored in the stress file.

3.4 Description of output data file

We take fault_mpi000201.nc as an example.

The six digits in the file name are the number of mpi block, separated by two digits, that is, the first two digits 00 indicate the first mpi block in X direction, the middle two digits 02 indicate the third mpi block in Y direction, and the last two digits 01 indicate the second mpi block in Z direction. These six digits indicate the position of the data on the fault plane. In the visualization script, the data output by each mpi will be put together. Use the ncdump -h command to view the file.

Commonly used variables:

init_t0: Rupture initiation time, which is used to calculate the rupture speed

Vs1: Strike-slip rate

Vs2: Dip-slip rate

Ts1: Shear stress of strike-slip component

Ts2: Shear stress of dip-slip component

Tn: Normal stress

Us1: Strike-slip displacement

Us2: Dip-slip displacement

Us0: Displacement

rake: Rake

State: State variable (if RSF law is used)

3.5 General steps to run the example

All the examples in the jobs folder can be run as follows.

1) **Before running any matlab scripts, run addmypath.m first!**
2) Run **conf_fault_grid.m** to generate fault geometry data specified by the Fault_geometry keyword in params.json file.
3) Run **conf_fault_stress.m** to generate the fault initial stress data specified by the Fault_init_stress keyword in params.json file.
4) Run **run.sh** to submit single-card or multi-card GPU tasks. The path of mpirun in the script should be modified before running. Another you need to modify is **echo "garray3 slots=4 garray4 slots=4" > nodelists**, where garray3/garray4 is cluster node name, 4 is the number of GPU cards used at this node.
5) Visualization: You can use MATLAB scripts to visualize the results.

   draw_snaps.m

   Variables in the fault rupture process.

   draw_snaps_wave_xy.

   Wave field propagation on the surface.

   plot_fault_seismo.

   Various seismograms recorded by virtual stations on faults.

   draw_init_t0.m

   Rupture speed.

   Surface_seismol.m

   Waveforms for surface stations.

4. Additional notes on media settings

1) Homogenous media is set directly in params.json.
2) If layered media is used, add "Media1D": "media1d.dat" to the params.json. File example:

```
# layernum
4
#depth(m) rho vp vs
2500    2100    4500    2600
3000    2750    6050    3500
10500   2800    6230    3600
19000   3050    6750    3900
```

3) If 3-D media is used, add "Media3D": "media3d.nc" to the params.json. Media3d.nc needs to contain Vp, Vs and rho of each grid point. File example:

```
netcdf media3d {
dimensions:
        nx = 160 ;
        ny = 920 ;
        nz = 300 ;
variables:
        double vs(nx, nz, ny) ;
        double vp(nx, nz, ny) ;
        double rho(nx, nz, ny) ;
```

References

Zhang, Z., Zhang, W., & Chen, X. (2014). Three-dimensional curved grid finite-difference modelling for non-planar rupture dynamics. Geophysical Journal International, 199(2), 860–879. https://doi.org/10.1093/gji/ggu308

Zhang, W., Zhang, Z., Li, M., & Chen, X. (2020). GPU implementation of curved-grid finite-difference modelling for non-planar rupture dynamics. Geophysical Journal International, 222(3), 2121–2135. https://doi.org/10.1093/gji/ggaa290