

CGFDM3D-EQR-SC

跨断层地震动模拟软件

V1.0

使用手册

王文强¹ 徐天鸿² 张振国²

1. 国家超级计算深圳中心（深圳云计算中心）
2. 南方科技大学地球与空间科学系

目 录

第 1 章 软件介绍	3
第 2 章 软件安装	4
2.1 软件运行环境.....	4
2.2 软件编译.....	4
第 3 章 参数配置	6
第 4 章 震源输入	8
第 5 章 地形设置	9
第 6 章 速度结构设置	10
第 7 章 软件输出	12
参考文献.....	14

第1章 软件介绍

CGFDM3D-EQR-SC 跨断层地震动模拟软件(以下简称 CGFDM3D-EQR-SC)是一套用于生成断层两侧成对位移的数值模拟软件,它是基于王文强等(2022)所提出的 CGFDM3D-EQR 地震灾害快速响应平台所编写的^[1]。原程序使用的数值方案是曲线网格有限差分法(CGFD)。在原程序的基础上,加入了激波捕获格式(Shock-Capturing)中的可选通量加权本质无振荡格式(AWENO),用于解决地震波场模拟中可能出现的激波问题^[2, 3]。

软件使用 MPI 和 CUDA C/C++混合编程,使其能够在拥有多 GPU 计算卡的服务器上高效运行。我们混合了原 AWENO 格式与一种新型高效的 AWENO 格式^[4],称为 ME-AWENO 格式,使得计算效率得到进一步提高,内存占用显著下降。软件的整体构架如图 1 所示,共包括三个模块:数据预处理模块,快速地震模拟模块以及烈度输出模块。利用该软件,可以提供从震源输入到地震动输出的一站式服务。

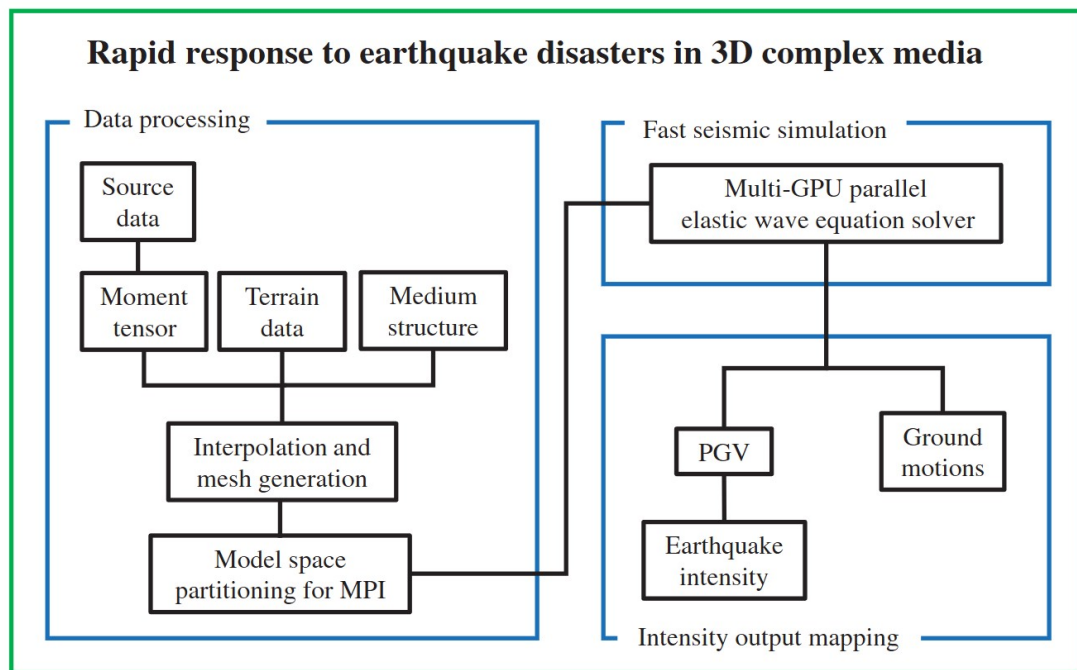


图 1 CGFDM3D-EQR-SC 的软件构架

第2章 软件安装

2.1 软件运行环境

CGFDM3D-EQR-SC 软件是在 CentOS7 环境下编写的, 使用 gcc/g++/nvcc 混合编译。需求软件环境:

- GCC 4.8.5 或以上
- G++ 4.8.5 或以上
- CUDA 10.0 或以上
- OpenMPI 4.1.1 或以上
- Proj 8.1.0 或以上

2.2 软件编译

编译前请将上述软件的 Home 目录填写于软件包内所包含的 Makefile 文件中 (图 2)。

```
CCHOME := /usr
CUAHOME := /public/software/cuda-10.0
#CUAHOME := /public/software/cuda-11.5
MPIHOME := /public/software/openmpi-4.1.1-cuda.10
PROJHOME := /public/software/proj-8.1.0
```

图 2 Makefile 中软件包 Home 的填写位置

表 1 对 Makefile 中的参数进行了说明:

表 1 Makefile 编译参数表

参数名	说明
SCFDM	是否使用 ME-AWENO 格式, 关闭则使用 CGFDM 格式
LF	黎曼解为 LF 通量, 默认开启, 目前仅有此通量
PML	是否使用 PML 吸收边界 ^[5]
EXP_DECAY	是否使用指数衰减吸收边界, 当 SCFDM 打开时, 仅可以使用此吸收边界
GPU_CUDA	是否使用 GPU 计算, 默认开启, 当 SCFDM 打开时, 仅支持此项开启
XFAST/ZFAST	最快计算轴, 默认为 XFAST, 不建议修改
FREE_SURFACE	是否开启自由表面边界条件, 默认开启。当 SCFDM 打

	开时, 使用特征边界条件 ^[3, 6] ; 当 SCFDM 关闭时, 使用牵引力镜像自由表面边界条件 ^[7, 8]
SOLVE_DISPLACEMENT	是否计算地表位移, 默认开启
Terrain_Smooth	地形平滑, 默认开启, 不建议修改
DealWithFirstLayer	首层处理, 默认开启, 不建议修改
LayeredStructureTerrain	层状地形结构, 默认开启, 不建议修改
StructureTerrain	地形结构, 默认开启, 不建议修改
SET_BASIN	默认关闭, 暂不支持此功能
FLOAT16	默认关闭, 暂不支持此功能

软件提供了两个默认参数 **Makefile** 文件, 分别为 **Makefile_CGFDM** 和 **Makefile_SCFDM**, 两个 **Makefile** 文件分别指定了使用 CGFDM 和 ME-AWENO 格式时的默认参数。随软件包提供了一个脚本 **make.sh**, 可以方便地用于指定方法进行编译。

第3章 参数配置

软件给出了一个 `params.json` 配置文件，运行前需要根据模拟的需要设置合理的参数。表 2 对 `params.json` 中的所有参数进行了说明：

表 2 `params.json` 参数表

参数名	说明
TMAX	模拟的最大时长（单位为 s）
DT	模拟的时间间隔
DH	模拟的空间间隔
IT_SKIP	软件输出的间隔步数，默认为 100
NX/NY/NZ	沿经度方向/纬度方向/竖直方向的模拟点数
Depth	模拟的最大深度（单位 km，此项与速度结构有关，务必填写）
centerX/centerY	投影中心所在的点号
sliceX/sliceY/sliceZ	输出点号所指定的切片，-1 为不输出
sliceFreeSurf	是否输出自由表面切片，1 为输出，0 为不输出
PX/PY/PZ	MPI 分块规则，沿 X/Y/Z 方向分别的块数，此项乘积务必与 <code>gpu_nodes</code> 中所指定的计算卡总块数一致
nPML	PML 层数，默认为 12，当 SCFDM 激活时默认为 30（使用指数衰减吸收边界），不建议修改此项
gpu_nodes	指定参与计算的 GPU
useSingleSource(ricker)	当此参数为 1 时软件为单点源
sourceX/sourceY/sourceZ	单点源时的震源位置
gauss_hill	单点源是否使用高斯山地地形
rickerfc	单点源时 Ricker 子波的主频
strike/dip/rake/Mw	单点源时的双力偶震源设置，分别指定走向、倾角、滑动角、震级来生成震源
useMultiSource	此项为 1 时使用多点源（有限断层）
centerLatitude/ centerLongitude	投影中心所在的经纬度
useTerrain	是否使用输入地形
useMedium	是否使用输入介质
ShenModel	是否使用 ShenWeisen 模型 ^[9]

Crust_1Model	是否使用 Crust1.0 模型
LayeredModel	是否使用层状介质模型
out	软件的输出位置
TerrainTif	地形的 Tif 文件位置
TerrainDir	地形的二进制文件位置
srtm90	是否使用 SRTM90 模型
lonStart/latStart	地形文件的起始经纬度
blockX/blockY	地形文件沿经度/纬度方向的块数
MLonStart/MLatStart	介质文件的起始经纬度
MLonEnd/MLatEnd	介质文件的终止经纬度
MLonStep/MLatStep	介质文件的经纬度步长
MVerticalStep	介质文件的竖直方向间隔 (单位为 m)
MediumDir	介质文件目录
LayeredFileName	层状介质文件名
CrustDir	Crust1.0 介质目录
CrustLonStep/CrustLatStep	Crust1.0 介质经纬度步长
sourceDir	震源文件目录
sourceFile	震源文件名
degree2radian	是否需要角度转弧度, 当震源文件提供的是角度时此参数设置为 1, 提供的是弧度时此参数设置为 0
FAST_AXIS	最快计算轴, 默认为 X, 不建议修改

第4章 震源输入

当 params.json 文件中的 useMultiSource 设置为 1 时，激活有限断层模型输入。有限断层模型文件存放在 sourceDir 及 sourceFile 设定的文件中。有限断层模型文件中需要包含以下数据，如表 3 所示：

表 3 有限断层模型参数

参数名	说明	参数维度
Lon/Lat	每个子断层的经纬度	(nx*ny, 1)
Z	每个子断层所在的深度，单位为 m	(nx*ny, 1)
Area	每个子断层的面积，单位为 m ²	(nx*ny, 1)
Strike/Dip	每个子断层的走向/倾角，可由角度或弧度给出，当设置为角度时 params 中的 degree2radian 参数设置为 1	(nx*ny, 1)
Rake	每个子断层的滑动角，可由角度或弧度给出，当设置为角度时 params 中的 degree2radian 参数设置为 1	(nx*ny, nt)
Rate	每个子断层的滑动速率，单位为 m/s	(nx*ny, nt)

软件随包给出一个震源文件写入 python 脚本，如图 3 所示。

```
def writeData(sourceFileName, Lon, Lat, Z, Area, Strike, Dip, Rake, Rate, NPTS, NT, dt):
    sourceFile = open(sourceFileName, "wb")
    value = struct.pack("i", NPTS)
    sourceFile.write(value)
    value = struct.pack("i", NT)
    sourceFile.write(value)
    value = struct.pack("f", dt)
    sourceFile.write(value)

    for i in range(NPTS):
        value = struct.pack("f", Lon[i])
        sourceFile.write(value)
        value = struct.pack("f", Lat[i])
        sourceFile.write(value)
        value = struct.pack("f", Z[i])
        sourceFile.write(value)
        value = struct.pack("f", Area[i])
        sourceFile.write(value)
        value = struct.pack("f", Strike[i])
        sourceFile.write(value)
        value = struct.pack("f", Dip[i])
        sourceFile.write(value)
        tvalue = struct.pack("f" * NT, *(Rake[i, :]))
        sourceFile.write(tvalue)
        tvalue = struct.pack("f" * NT, *(Rate[i, :]))
        sourceFile.write(tvalue)

    sourceFile.close()
```

图 3 有限断层震源写入脚本

第5章 地形设置

软件提供了 SRTM90 的内置读取, 用户可由 <https://srtm.csi.cgiar.org/srtmdata/> 网站下载, 解压缩后获得 tif 格式文件, 将文件名修改为先纬度后经度的形式, 如图 4 所示:

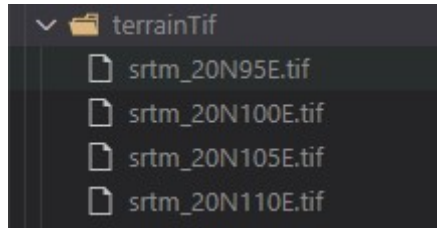


图 4 SRTM90 文件名格式

软件随包提供了一个将 tif 格式转换成软件输入所需的二进制文件的 python 脚本 tif2bin, 如图 5 所示:

```
def tif2bin( lonStart, latStart, terrainTifPath, terrainBinPath, FAST_AXIS ):
    tifFileName = "%s/srtm_%dN%dE.tif" % ( terrainTifPath, latStart, lonStart )
    #terrain = np.zeros( [NY, NX] )
    terrain = np.flipud( imageio.imread( tifFileName ) )

    ( NY, NX ) = np.shape( terrain )
    #print( "NY = %d, NX = %d" % ( NY, NX ) )
    binFileName = "%s/srtm_%dN%dE.bin" % ( terrainBinPath, latStart, lonStart )
    print( "Tif file is being converted to Binary file: %s" % binFileName )
    binFile = open( binFileName, "wb" )
    if FAST_AXIS == 'Z':
        for i in range( NX ):
            for j in range( NY ):
                t = struct.pack( "f", float( terrain[j, i] ) )
                binFile.write( t )
    else:
        for j in range( NY ):
            for i in range( NX ):
                t = struct.pack( "f", float( terrain[j, i] ) )
                binFile.write( t )
```

图 5 tif 转换成 bin 文件的 python 脚本

第6章 速度结构设置

软件可以设置三种不同的速度结构模型,若 params.json 文件中的 useMedium 设置为 0, 则使用程序内置的均匀介质模型。

对于层状介质模型, 用户可提供模型的文本文件, 如图 6 所示:

1	0.00	6.0000	3.5200	2.7100
2	10.00	6.2500	3.6500	2.7600
3	20.00	6.7500	3.8700	2.8700
4	30.00	8.0700	4.5200	3.2700

图 6 层状介质模型文本文件

软件随包提供了将文本文件转化为软件要求的二进制文件的脚本 setLayeredModel, 如图 7 所示:

```
value = struct.pack("i", Layers)
LayeredFile.write(value)

for i in range(Layers):
    value = struct.pack("f", Dep[i])
    LayeredFile.write(value)
    value = struct.pack("f", Vs[i])
    LayeredFile.write(value)
    value = struct.pack("f", Vp[i])
    LayeredFile.write(value)
    value = struct.pack("f", Rho[i])
    LayeredFile.write(value)

LayeredFile.close()
```

图 7 层状介质模型转化为程序输入脚本

若需要使用三维速度结构, 软件提供了 ShenWeisen 中国大陆三维 S 波速度结构模型^[9]的内置读取, 如图 8 所示:

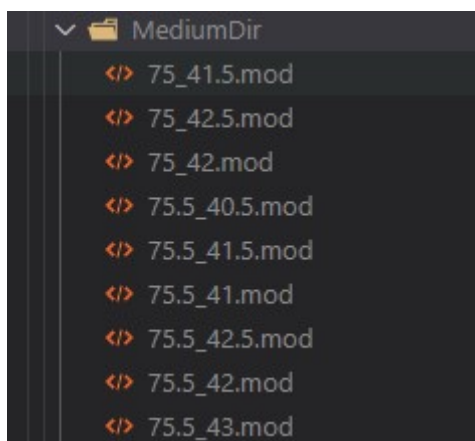


图 8 ShenWeisen 中国大陆三维 S 波速度结构模型

对于介质密度和 P 波速度，使用 Brocher（2005）给出的经验关系来进行计算^[10]。

对于其他三维速度结构，推荐将速度结构格式转换为 ShenWeisen 模型的格式，并于 params.json 文件中设置 MLonStart/MLatStart/MLonEnd/MLatEnd/MLonStep/MLatStep/MVeticalStep 等参数，软件将会按该格式进行读取。

第7章 软件输出

软件提供了一个台站参数配置文件 `station.json`，文件中需要设置每一个台站的位置（该位置是在计算网格中的点号），如图 9 所示：

```
{
  "point": 1,
  "lon_lat": 0,
  "station(point)": {
    "0": [
      300,
      300,
      400
    ],
    "1": [
      400,
      300,
      400
    ],
    "2": [
      500,
      300,
      400
    ],
    "3": [
      500,
      300,
      380
    ]
  }
}
```

图 9 台站配置文件

软件运行过程中，将会按照 `params` 中设定的输出间隔将所有已设置的切片数据存储在每个 MPI 计算块所保存的文件中，随包提供了数个用于绘制切片图像，绘制台站波形的脚本。图 10 展示了程序的波场快照，图 11 展示了台站波形。

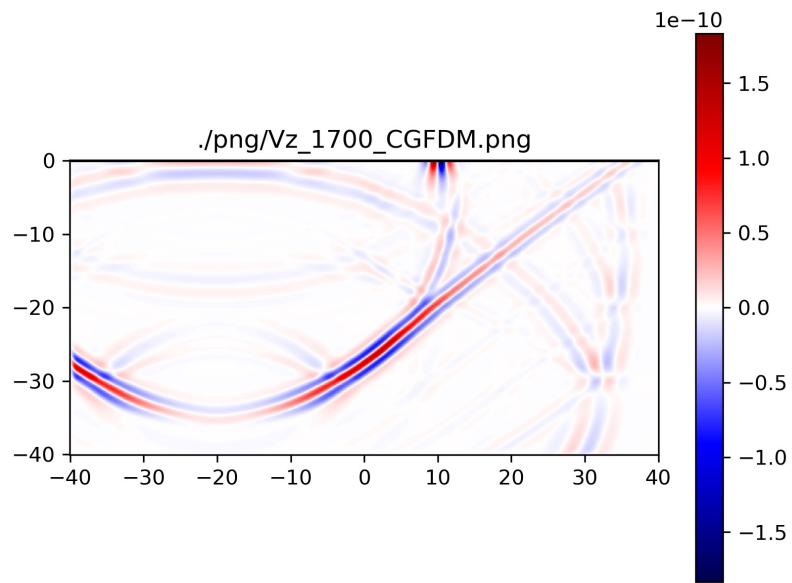


图 10 波场快照

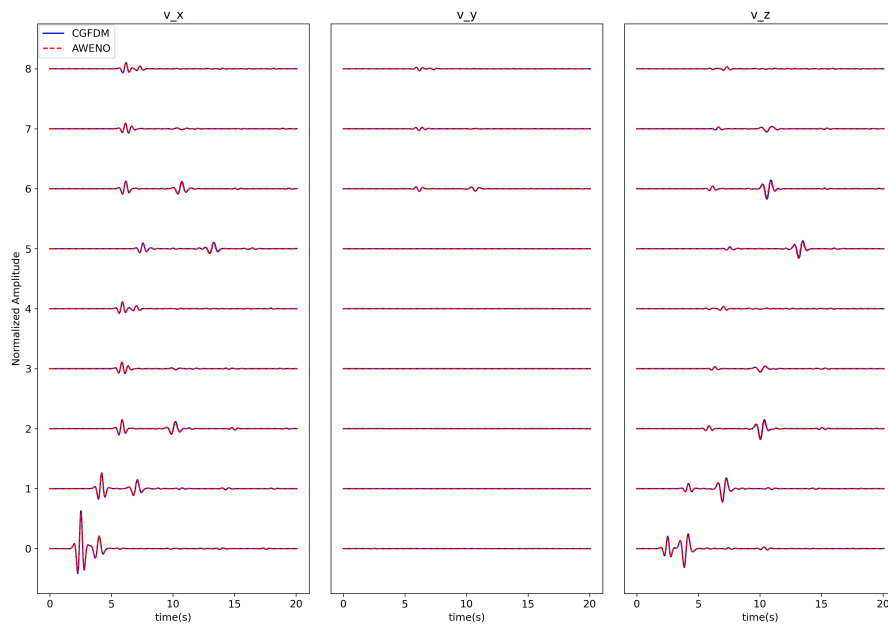


图 11 软件两种数值格式的波形对比

参考文献

- [1] WANG W, ZHANG Z, ZHANG W, et al. CGFDM3D - EQR: A platform for rapid response to earthquake disasters in 3D complex media [J]. *Seismological Research Letters*, 2022, 93(4): 2320-34.
- [2] JIANG Y, SHU C-W, ZHANG M. Free-stream preserving finite difference schemes on curvilinear meshes [J]. *Methods and applications of analysis*, 2014, 21(1): 1-30.
- [3] XU T, ZHANG Z. Numerical simulation of 3D seismic wave based on alternative flux finite-difference WENO scheme [J]. *Geophysical Journal International*, 2024, 238(1): 496-512.
- [4] CHU S, KURGANOV A, XIN R. New more efficient A-WENO schemes [J]. Available at SSRN: <https://ssrncom/abstract=4486288> or <http://dxdoiorg/102139/ssrn4486288>, 2023.
- [5] ZHANG Z, ZHANG W, CHEN X. Complex frequency-shifted multi-axial perfectly matched layer for elastic wave modelling on curvilinear grids [J]. *Geophysical Journal International*, 2014, 198(1): 140-53.
- [6] BAYLISS A, JORDAN K, LEMESURIER B, et al. A fourth-order accurate finite-difference scheme for the computation of elastic waves [J]. *Bulletin of the Seismological Society of America*, 1986, 76(4): 1115-32.
- [7] ZHANG W, CHEN X. Traction image method for irregular free surface boundaries in finite difference seismic wave simulation [J]. *Geophysical Journal International*, 2006, 167(1): 337-53.
- [8] ZHANG W, ZHANG Z, CHEN X. Three-dimensional elastic wave numerical modelling in the presence of surface topography by a collocated-grid finite-difference method on curvilinear grids [J]. *Geophysical Journal International*, 2012, 190(1): 358-78.
- [9] SHEN W, RITZWOLLER M H, KANG D, et al. A seismic reference model for the crust and uppermost mantle beneath China from surface wave dispersion [J]. *Geophysical Journal International*, 2016, 206(2): 954-79.
- [10] BROCHER T M. Empirical relations between elastic wavespeeds and density in the Earth's crust [J]. *Bulletin of the seismological Society of America*, 2005, 95(6): 2081-92.