

User Manual of JDSurfG

Nanqiao Du

2020 年 7 月 29 日

目录

1 Introduction

JDSurfG 包含了三个模块，分别用来计算球坐标系下的重力矩阵，直接面波反演和面波-重力联合反演。其中重力部分基于自适应 Gauss-Legendre 数值积分方法，面波部分基于面波一步法计算面波走时敏感核。该方法通过岩石物理中的经验关系式构建 S 波速度和密度的转换公式，从而将重力数据和面波走时数据统一到一个反演框架中。该方法能够用来反演地壳-上地幔内的 S 波速度结构。

该程序由 c++ 和 Fortran 混编写成，用 c++ 来构建反演框架，封装数据，处理 I/O，并计算重力矩阵和面波一维敏感核，Fortran 主要负责计算面波走时和面波二维敏感核。

2 Preliminaries

该程序使用 **Eigen** 来进行多维数组的存取，并用 **GSL** 库来计算 Gauss-Legendre 节点的位置和权重。因此务必在您的计算机上装好这两个库。同时，请确保您的 g++ 编译器支持 c++11 标准 (gcc $\geq 4.8.1$)。

3 Installation

下载.zip 包之后，按如下方式解压

```
1 unzip JDSurfG.zip
```

如果您没有将 Eigen 和 GSL 的 include (或) lib 文件夹添加到环境变量中，那么请在 JDSurfG/include/Makefile 中添加他们的路径，例如：

```
1 path_to_eigen= -I/path-to-eigen
```

之后，通过如下方式编译

```
1 cd JDSurfG
2 make
```

之后，你会惊人地发现在 bin 文件夹下，会出现三个可执行文件

```
1 mkmat DSurfTomo JointSG
```

4 面波直接反演模块

该模块需要三个输入文件 (如果需要做检测版, 则需要四个):

- DSurfTomo.in: 包含模型信息、使用的频散的信息, 阻尼等
- surfdataSC.dat : 面波频散数据
- MOD: 初始模型
- MOD.true: 检测板模型

在说明输入格式之前, 有几点注意的地方:

- (1) 该模型在经纬度方向是均匀网格, 在深度方向的网格间距可以变化
- (2) 输入的模型要比反演区域大, 具体来说, 如果需要反演的区域在区域: $(lat_0 \sim lat_1, lon_0 \sim lon_1, 0 \sim dep_1)$ 之间, 则输入模型的区域是: $(lat_0 - dlat \sim lat_1 + dlat, lon_0 - dlon \sim lon_1 + dlon, 0 \sim dep_1 - dz)$
- (3) 台站不要放在模型边界区域。不然有可能射线追踪时部分射线跑到模型边界上, 引起计算误差。

下面分别说明几个文件的输入格式:

4.1 参数文件

例子中的参数文件为 DSurfTomo.in, 它是一个 self-explanatory 的文件,

- 第 1 行三个数依次是纬度方向节点个数, 经度方向节点个数以及深度方向节点个数
- 第 2 行左上角位置的经纬度 (注意这是**输入模型**的左上角!)
- 第 3 行纬度、经度网格间距 (度)
- 第 4 行反演的正则化系数和阻尼
- 第 5 行因为输入模型是网格节点, 需要将其转换为层状介质计算频散, 该参数表示将节点变为层状介质需要在节点间插入几层, 通常来讲取为 3 足够。

- 第 6 行模型的最大和最小速度
- 第 7 行最大迭代次数
- 第 8 行使用的全部 Rayleigh 波相速度周期个数, 如果没使用则置零。
- 第 9 行如果 8 不为 0, 将所有周期列出来, 否则依次进行 Rayleigh 群速度, Love 相速度、群速度
- 第 ? 行是否进行检测板测试? 如果是填 1
- ? noiselevel, 加入噪声的大小 (注意此处是和数据的相对大小):

$$d_i = d_{syn} + (d_{syn} * noiselevel * N(0, 1))$$

4.2 初始模型和检测板模型文件

例子中初始模型和检测板模型文件分别为 MOD 和 MOD.true, 这两个文件格式稍有不同。主要差别在第一行上。

MOD: 第一行存放所有 z 方向节点的位置 (km)。MOD.true 不用放。之后的行存放输入的速度模型: 每一行, 存放一个给定深度的, 每个给定经度的, 所有纬度上的速度 (km/s)。即速度按行优先方法存为 V(nz,nlon,nlat)。第 i 行存放 V(:,i);

4.3 频散数据文件

例子中频散数据文件为 surfdataSC.dat, 用于存放面波频散数据 (台站间距/面波走时)。如下我们展示了该文件的前九行:

```
# 31.962600 108.646000 1 2 0
33.732800 105.764100 3.0840
34.342500 106.020600 3.1154
33.357400 104.991700 3.0484
33.356800 106.139500 3.0820
34.128300 107.817000 3.1250
33.229300 106.800200 3.0575
# 29.905000 107.232500 1 2 0
34.020000 102.060100 2.7532
```

可以看出, 该文件包括面波激发台站 (带 # 号) 和接收台站 (不带 # 号)。对于激发台站, 该行的格式为 # lat lon period-index wavetype velotype:

- wavetype: 表示面波类型, Rayleigh 波为 2, Love 为 1
- velotype: 表示速度类型, 相速度为 0, 群速度为 1
- period-index: 表示该面波群/相速度的周期的序号, 比如在 Rayleigh 相速度中用到的周期为 4s, 5s, 6s, 7s, 而该台站对之间的数据的周期是 5s, 则倒数三个数分别为 2, 2, 0。

每一个激发台站后面紧跟, 在该面波类型, 该周期下, 所有有频散数据的接收台站, 每一行依次为 lat lon v。

4.4 运行方法

将上述文件准备到一个文件夹中, 进入该文件夹, 命令行执行:

```
1 ./DSurfTomo -h
```

获取帮助, 然后按提示输入即可。

4.5 输出文件

该程序共有两个输出文件夹: kernel/和 results/, 在 kernel 中保存了面波伪 3-D 敏感核文件, results 中保存了每一次迭代后的模型文件和面波走时文件。下面逐次介绍两个文件夹。

- 文件夹 kernel: 保存了面波伪 3-D 敏感核。其中 txt 文件个数 = 并行使用的线程数 (见?? 节), 编号从 0 开始。每一个 txt 文件的格式是: 数据编号, 模型参数编号 (只包括要反演的部分) 以及敏感核的大小。
- 文件夹 results: 保存了每一次迭代的模型文件 (mod_iter) 以及面波走时文件 (res.dat)。其中模型文件的格式为: 经度、纬度、深度、S 波速度。面波走时文件的格式为: 台站间距, 观测到时、理论到时。

5 联合反演模块

该模块需要 4-6 个输入文件:

- JointSG.in: 包含模型信息、使用的频散的信息, 阻尼, 联合反演权重等。
- surfdataSC.dat : 面波频散数据。
- obsgrav.dat : 重力异常数据。
- gravmat.dat : 重力异常对密度的敏感核矩阵。
- MOD: 初始模型。
- MOD.true: 检测板模型, 用于检测板测试。
- MOD.ref : 速度参考模型, 用于计算重力异常理论值。

下面逐次介绍这几个文件。

5.1 参数文件

JointSG.in 和 DSurfTomo.in 的文件格式基本相同。不同之处在于最后两行。为了理解其参数的意义, 这里列出了联合反演的 misfit function:

$$L = \frac{p}{N_1 \sigma_1^2} (d_1 - d_1^o)^2 + \frac{1-p}{N_2 \sigma_2^2} (d_2 - d_2^o)^2 \quad (1)$$

- 倒数第二行: 参数 p , 介于 0-1 之间, $p=0$ 表示重力单独反演。由于位场反演不唯一性较高, p 值通常取得比 0.5 大。
- 倒数第一行: 面波和重力的均方差。

5.2 重力数据文件

重力数据文件比较简单, 将所有重力数据, 按照每行分别为经度, 纬度和该点的重力异常排成一个文件即可。下面给出了该文件前 9 行:

```
100.000000 35.000000 -224.206921
100.000000 34.950000 -224.110699
100.000000 34.900000 -241.774731
100.000000 34.850000 -240.245968
100.000000 34.800000 -234.187542
100.000000 34.750000 -246.670605
```

```
100.000000 34.700000 -238.575939
100.000000 34.650000 -241.357250
100.000000 34.600000 -260.059429
```

5.3 重力矩阵文件

该文件通过重力模块生成 (见第 6 节)

5.4 速度参考模型

由于使用的重力异常是重力绝对大小相对正常椭球体的重力异常。因此在实际计算中,需要归算到局部异常中。因此,在程序中对观测值进行了去均值处理。每次迭代时,计算理论重力异常的方法分三步:

- 计算当前模型与参考模型分别对应的密度值。
- 计算密度异常,之后用重力密度矩阵获取该异常对应的重力值。
- 对所获得的重力异常去均值。

该模型可以用第三方密度模型代替,也可以用程序默认的处理 (对初始模型的每个深度求平均值)。

5.5 运行方法

将上述文件准备到一个文件夹中,进入该文件夹,命令行执行:

```
1 ./JointSG -h
```

获取帮助,然后按提示输入即可。

5.6 输出文件

该程序共有两个输出文件夹: kernel/和 results/。其中 kernel 的格式与?? 节一致。

文件夹 results: 保存了每一次迭代的模型文件 (joint_mod_iter), 面波走时文件 (res_surf.dat), 和重力异常文件 (res_grav*.dat)。面波走时文件和模型文件和 ?? 节一致, 重力异常文件的格式为: 重力异常观测值, 重力异常理论值。

6 重力模块

该模块需要如下几个输入文件:

- JointSG.in 或 DSurfTomo.in : 包含模型网格信息
- MOD(和 MOD.true) : 初始速度模型和真实速度模型
- obsgrav.dat : 重力数据观测值和相应的经纬度

上述格式在之前已经说明过, 在此不再赘述。

6.1 运行方法

将上述文件准备到一个文件夹中, 进入该文件夹, 命令行执行:

```
1 ./mkmat -h
```

获取帮助, 然后按提示输入即可。

6.2 输出文件

该程序共有个输出文件:gravmat.dat。为重力异常的密度敏感核。格式为: 数据编号、模型参数编号和敏感核大小。

7 一些小脚本

8 高级选项

8.1 并行使用的线程数

在 include/openmp.hpp 中有一行:

```
1 #define nthread 4
```

将 4 改为其他线程数即可。

8.2 速度-密度经验关系的选择

在 src/utls/empirical.f90 中

```

1  subroutine empirical_relation(vsz, vpz, &
2      rhoz) bind(c, name="empirical_relation")
3      use, intrinsic :: iso_c_binding
4
5      real(c_float), intent(in) :: vsz
6      real(c_float), intent(out) :: vpz, rhoz
7
8      vpz=0.9409 + 2.0947*vsz - 0.8206*vsz**2+ &
9          0.2683*vsz**3 - 0.0251*vsz**4
10     rhoz=1.6612*vpz- 0.4721*vpz**2 + &
11         0.0671*vpz**3 - 0.0043*vpz**4 + &
12         0.000106*vpz**5
13
14     end subroutine empirical_relation

```

将经验关系换为其他即可。如果要进行联合反演，注意要修改该文件中相应的导数函数。

8.3 Gauss-Legendre 节点个数的选择以及重力矩阵的计算范围

在 src/gravity/gravmat.cpp 中，可以修改最大和最小节点个数，重力矩阵不为 0 的最大距离 (超过该范围的两个点之间的吸引力可以忽略不计)

```

1  #define NMIN 5
2  #define NMAX 256
3  #define maxdis 100.0

```

8.4 面波走时正演计算的网格间距

由于使用的网格不一定能准确计算面波走时，因此在计算过程中，需要用一套新的网格来进行走时正演计算。该计算网格是通过将模型网格进一步划分来实现的，即通过在模型网格中间插入一些新的点。这些新的

点连同模型网格构成一个新的计算网格。插入的点数可以在 `src/Surface-Wave/sphfmm.f90` 的 `synthetic` 和 `CalFrechet` 函数中修改 (参数 `gdx` 和 `gdy`, 默认为 8)。纬度和经度方向分别插入的点数为 `gdx+1` 和 `gdy+1`, 插入这些点后, 计算网格将在纬度方向和经度方向分别有 $(nlat-1) * gdx + 1$ 和 $(nlon+1) * gdy + 1$ 个节点,