波形模拟验证速度模型操作流程

（更新于 2024 年 2 月 26 日）

陈举庆

主要分为两个部分，第一部分，速度模型正演模拟合成理论波形；第二部分，基于互相关的方法分别从走时和振幅两个角度对理论波形与合成波形在不同的周期范围内进行比较，从而对速度模型进行统计评估，特别要说明的是，速度模型验证可以通过合成天然地震波形进行比较，也可以通过合成噪声互相关波形进行比较，这里仅说明如何通过合成噪声互相关（ZZ 分量， Rayleigh 面波）进行比较。利用有限差分正演的程序已经部署在太乙服务器上，通过公共账号进行操作（具体操作细节本操作流程后面部分会详细介绍），其他相关的程序都在本程序包 ModelValidCode 中，下面会按步骤进行相应介绍。

一、速度模型正演

正演模拟的程序为张伟老师有限差分正演模拟程序，已被部署在太乙服务器的公共账号上，**服务器地址为 172.18.6.175，账户名为 ess-wangp，密码是 l1UvG3r9（第一个字母是小写的 L，不是大写!!!)**，具体命令是 ssh -X ess-wangp@172.18.6.175。另外需要说明的是.175 这个节点不能开图形界面，也就是不能画图，所以如果需要在太乙服务器上画图的话，可以在本地重新另开一个终端登录到.176 节点上，具体命令是 ssh -X ess-wangp@172.18.6.176。需要**特别注意**的是公共账号下的 home 目录勿动，因为这里配置好了波形正演的程序。另外，因为太乙服务器上对每个账户 home 下的存储空间有限制，所以**建议个人不要在 home 下存放数据**，但是可以建立以自己名字命名的文件夹用来存放脚本。

1. **准备参数文件**（网格，速度，台站，震源）(程序包：parameter_prepare)

（1）**速度模型插值**（程序：VsInterp0.m 和 VsInterp1.m）

原始速度模型如果网格较粗（网格间距大于 0.5 度），要进行适当插值。程序包中给的这两个插值程序的输入速度模型（待插值模型）是.mat 格式的三维数据体（lat*lon*dep)。这两个程序的主要差别在于对某一个深度上水平方向

的二维插值方法不同。VsInterp0.m 用的 matlab 自带的 interp2 函数，而 VsInterp1.m 用的是从网上找的克里金插值函数 kriging。相比之下，interp2 要比 kriging 运行速度更快，插值结果也更加稳定（VsInterp1.m 中的 kriging 有时插值结果不稳定），所以推荐使用 VsInterp0.m。当然，也可以使用自己的插值函数来进行插值，这里的两个插值程序只是提供参考。

(2) 准备网格文件（程序：topo3.m）

有限差分程序要划分网格，所以根据自己速度模型的空间展布进行网格划分（一般设置的网格尺度跟速度模型尺度大致相同即可），具体程序见 topo3.m。输入参数说明如下（以华南东部为例，华南东部模型的经度范围是 108°-123° E；纬度范围是 20°-36° N）：

```
nx=1500; %EW
ny=1800; %NS
dx=1e3; %
dy=1e3;


Xlen = nx*dx;%1500KM
Ylen = ny*dy;%1600KM
Zdep = 75e3;%here doesnot need to exceed bounds
```

dx，dy：东西向和南北向上的网格大小，单位是 m。

nx，ny：东西向和南北向上的网格数，分别乘以 dx，dy，便是东西向和南北向的展布 Xlen 与 Ylen。

Zdep：深度方向展布

最后**输出的参数文件**包括：gridvmap_F.dat，grid_FX.dat，grid_FY.dat。

(3) **准备速度文件**（程序：velw3.m）

将插值后的速度模型转为.nc 格式的文件，具体程序见 velw3.m。最后**输出的参数文件**为 Vsim_fortran_new.nc。本程序在写入速度文件时，实际上是利用了 Brocher (2005)中的经验关系(见下图)去得到V_p和ρ，如果在V_s反演中用到的同样是 Brocher (2005)中的经验关系，这里就不需要改，而如果在V_s反演中用到的经验关系并不是 Brocher (2005)中的经验关系，那这里**建议改成反演中所用到的经验关系**，使二者统一。

```
Vp = 0.9409+2.0947.*Vs-0.8206.*Vs.^2+0.2683.*Vs.^3-0.0251.*Vs.^4;
rho  = (1.6612.*Vp-0.4721.*Vp.^2+0.0671.*Vp.^3-0.0043.*Vp.^4+0.000106.*Vp.^5);
```

(4) **准备台站文件**（程序：station.m）。

将台站经纬度位置转为波形正演程序中的网格点位置，具体程序为 station.m。程序中具体的参数说明如下：

```
load('Staloc.dat')
Stalon = Staloc(:,1);
Stalat = Staloc(:,2);
%%
Clat1 = 20; Clat2 = 36;
Xlon = 108; Ylat = 20;
lonunit = 2*pi*6359.752/360*cos((Clat1+Clat2)/2/180*pi)*1e3; % unit length of longitude
latunit = 2*pi*6378.137/360*1e3; % unit length of latitude
```

Staloc.dat：准备进行模型验证的所有台站位置（两列，第一列是经度，第二列是纬度）；
Clat1 与 Clat2：速度模型的纬度范围，即最小纬度与最大纬度；
Xlon 与 Ylat：台站位置由经纬度转为网格点位置的锚点经纬度位置，一般设置为速度模型的最小经度与最小纬度。这里的**输出文件**是 stapos.dat。

(5) **准备虚拟震源文件（程序：virsrcSta.m）**。

从所有的台站中均匀挑选出一定数量的台站（一般是按照 10：1 的比例即可，例如果总共 600 个台站，可以挑出 60 个）作为虚拟震源，这一步骤暂时没有实现自动化，需要自己手动挑选，然后将虚拟震源的经纬度位置转为网格点位置，具体程序为 virsrcSta.m，如果每次只改变速度模型，那么震源位置就不用修改。需要说明的是，virsrcSta.m 其实和第（4）步中的 station.m 总体相似，差别仅在于位置的写入不同，因为波形模拟程序中台站位置与震源位置的设置是不一样的，这里的**输出文件是 virsrc.dat,建议不要改文件名**，因为后续批量进行波形模拟时需要批量写入震源位置，对应的 shell 脚本 update_source.sh 会读写这个 virsrc.dat 文件。

2. **波形正演**（程序包：wave_forward）

波形正演操作的思路是，批量建立多个震源文件夹，然后批量提交任务，完成多震源的波形正演。在具体操作时建议按照“先单个调试后批量计算”的思路进行。

2.1 **单个震源调试**

(1) 登录太乙，因为 home 目录下的存储空间有限制，而/scratch/目录下的存储没有限制，所以直接在/scratch/目录下建立以自己用户名为名称的文件夹。具体步骤如下 cd /scratch/2022-11-30(这里是当前日期即可)； mkdir ess-wangp；**需要注意的是/scratch 目录下的文件仅保存 10 天，所以请注意保存自己的文件夹以免被删除**。

(2)上传 wave_forward 程序包到/scratch/目录下自己建好的文件夹下 /scratch/2022-11-30/ess-wangp/。

(3) 上传参数文件，也就是把之前生成的 gridvmap_F.dat，grid_FX.dat，grid_FY.dat 和 Vsim_fortran_new.nc，通过 scp 从本地上传到 /scratch/2022-11-30/ess-wangp/wave_forward/template/parfile/中。

(4) 对于模板文件夹 template 的说明。

template 文件夹其实就包含了波形正演的输入和输出。它包含以下重要的子文件夹或者文件：

input：存放输入参数的文件夹(这里的输入参数并不是前面生成的介质参数文件)

output：存放波形模拟结果的文件夹

parfile：存放介质参数的文件夹，也就是存放之前产生的

gridvmap_F.dat，grid_FX.dat，grid_FY.dat 和 Vsim_fortran_new.nc。

pyplot:存放画图和计算 misfit 的 python 脚本。

SeisFD3D.conf：有限差分模拟的主要配置文件

SeisGrid.conf：网格的配置文件

SeisMedia.conf：介质的配置文件

SeisSource.conf：震源的配置文件

runAll.sh：提交任务的作业脚本。

(5)配置 SeisFD3D.conf、SeisGrid.conf、SeisSource.conf。

① SeisFD3D.conf 的配置：

```
##############################################################################
#                           for mpi_mod                                      #
##############################################################################
dims = 5 8 5
#dims = 5 4 1
##############################################################################
#                          for main program                                  #
##############################################################################
ni    = 300  #1500
nj    = 225  #1800
nk    = 15  #75
#ni    = 300
#nj    = 400
#nk    = 200

nt    = 4000
stept = 0.08
steph = 1000.0
```

首先是上图中的 dims 和 ni,nj,nk，这个就根据网格尺度来设置了，dims 三个维度相乘得到单次波形正演用到的核数，这里便是 dims[1]*dims[2]*dims[3]=5*8*5 = 200 核（**建议保持 200 核**，如果这里核数改的话，相应的作业脚本 runALL.sh 中核数-n 也要改）。这里 dims 的设置要和 ni,nj,nk 匹配起来，使得 dims[1]*ni = 沿经度方向上的网格数，这里是 5*300=1500，其他两个维度以此类推，主要要保证 dims 三个维度与 ni,nj,nk 都是整数。

然后将台站位置复制到 SeisFD3D.conf 中相应位置上，见下图：

```
# seismo-point output
number_of_recv = 652
recv_001 = 884013.734712         1177760.212593   9000.0E3
recv_002 = 918864.653123         1277135.122024   9000.0E3
recv_003 = 911200.587263         1436633.688433   9000.0E3
recv_004 = 805256.147438         1268886.347756   9000.0E3
recv_005 = 1007511.041487        1367615.604141   9000.0E3
recv_006 = 928008.608938         1407545.905488   9000.0E3
recv_007 = 764750.285292         1441142.127810   9000.0E3
recv_008 = 810548.469131         1263331.505166   9000.0E3
recv_009 = 861472.364537         1556435.724424   9000.0E3
recv_010 = 895970.461501         1316241.659140   9000.0E3
recv_011 = 862834.647343         1406354.786937   9000.0E3
recv_012 = 985714.516587         1310185.878841   9000.0E3
recv_013 = 1017086.223514        1080478.109589   9000.0E3
```

② SeisGrid.conf 的配置

```
vmap_spacing_number    = 74 ###
vmap_spacing_isequal   = T
vmap_layer_filename    = ./parfile/gridvmap_F.dat
```

SeisGrid.conf 仅需要配置 vmap_spacing_number，这里是 74。其计算方式是 nk*dims[3]-1，这里的 nk，dims[3]都是在 SeisFD3D.conf 中。

③ SeisSource.conf 的配置

```
#                        single force source                           #
####################################################################
number_of_force_source = 1
force_stf_window = 1
force_stf_type = ricker
force_stf_timefactor = 15  # gauss t0; ricker t0; bell starting
force_stf_freqfactor = 0.1 # gauss a;  ricker fc; bell width
#force_stf_type = ricker
#force_stf_timefactor = 15  # gauss t0; ricker t0; bell starting
#force_stf_freqfactor = 0.1 # gauss a;  ricker fc; bell width
# x,y,z           | start  | f0    | fx   fy   fz
# AH-ANQ 884013.734712   1177760.212593   4.5e3  0.0    1.0e+22   0.0 0.0 1.0
<anchor_force>
 884013.734712  1177760.212593 4.5e3 0.0   1.0e+22   1.0 1.0 1.0
```

SeisSource.conf 中首先需要修改震源位置，也就是上图中的<anchor_force>。在调试阶段可以手动复制一个虚拟震源的位置，而在批量计算阶段，直接利用 update_source.sh 进行自动更新。

另外合成波形的周期范围是由所用子波的参数确定的，如果想要合成不同周期范围的波形，就要调整子波主频 fc（这里对应的是 force_stf_freqfactor）和子波移动时间 t0（这里对应的是 force_stf_timefactor），之所以要设置子波移动时间，是因为我们希望子波加载是从 0 开始慢慢达到最大值的，而 Ricker 子波一般有是围绕 0 轴对称的，所以我们要把负半轴的子波移动到正半轴。子波移动时间和主频之间的关系大概为 t0=1.5/fc。而合成波形的周期范围大致围绕主频，比如这里主频 fc 是 0.1Hz，对应的就是 10s，可以根据自己关心的周期范围以及实际互相关数据的周期范围去设置主频。现在目前的默认设置是可以合成 3-40s 的波形，所以对于一般大尺度的速度模型，这个周期范围应该是足够的。

（6）在 template 中加载网格、介质、震源和台站

打开 runAll.sh，如下图：

```
ulimit -s unlimited
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mpirun -np $LSB_DJOB_NUMPROC /work/ess-wangp/FD3DtopoEw/src/bin/seis3d_grid_mpi SeisFD3D.conf 1
mpirun -np $LSB_DJOB_NUMPROC /work/ess-wangp/FD3DtopoEw/src/bin/seis3d_metric_mpi SeisFD3D.conf 2
mpirun -np $LSB_DJOB_NUMPROC /work/ess-wangp/FD3DtopoEw/src/bin/seis3d_media_mpi SeisFD3D.conf 3
#/work/ess-wangp/FD3DtopoEw/src/bin/seis3d_source SeisFD3D.conf 4
#/work/ess-wangp/FD3DtopoEw/src/bin/seis3d_station SeisFD3D.conf 5

#mpirun -np $LSB_DJOB_NUMPROC /work/ess-wangp/FD3DtopoEw/src/bin/seis3d_wave_mpi SeisFD3D.conf 6
```

主要有 6 步， 首先是确认 1，2，3 步打开注释，4，5，6 处于注释状态，然后提交作业脚本，即 bsub <runAll.sh。利用 bjobs 查看作业运行状态，作业一旦运行，生成的文件可以在 input 文件夹下查看。运行前三步的目的在于加载网格和介质，前三步运行结束后，若程序没有报错（可以查看一下文件 sc.err 中有无内容，如果没有内容，程序就是没有报错的），通过 pyplot 画图

程序包下的 figmedia3d.py 查看加载的介质有无明显异常（有无明显条带），使用命令示范：./pyplot/figmedia3d.py --mapdim=1 --direction=x --whichitem=Vs --xystep=10 --zstep=10 --nprofile=100。

在前三步运行结束后，记得查看文件 sc.out 中末尾位置所输出的波形模拟允许的最大时间步长（保证计算稳定性），如下图 ：

```
Maximum allowed time step is  8.9139968E-02  in thread          12
```

然后，需要检查 SeisFD3D.conf 中对于计算时间步长（stept，见下图）的设置是否合理（必须小于允许的最大时间步长）。

```
nt    = 4000
stept = 0.08
```

这里的 stept 就是设置的计算时间步长，而 nt 是需要计算的总的时间步数，nt*stept 就定义了计算出的波形的时间长度，在这个例子当中，那就是计算 4000*0.08=320 s 的波形（需要计算多长时间的波形可以根据自己的实际需要进行设置）。

然后，在命令行加载台站和震源（对应了第 4 步和第 5 步，但实际无先后顺序）：

加载震源：/work/ess-wangp/FD3DtopoEw/src/bin/seis3d_source（**注意！这里有一个空格哦**）SeisFD3D.conf，如下图：

```
/work/ess-wangp/FD3DtopoEw/src/bin/seis3d_source SeisFD3D.conf
```

加载台站：/work/ess-wangp/FD3DtopoEw/src/bin/seis3d_station（**注意！这里有一个空格哦**）SeisFD3D.conf，如下图：

```
/work/ess-wangp/FD3DtopoEw/src/bin/seis3d_station SeisFD3D.conf
```

最后，注释前 5 步，打开第 6 步，提交作业脚本：bsub runALL.sh。运行结束后，如无报错，通过 figseismo3d.py 来查看波形。使用命令示范：./pyplot/figseismo3d.py -p 35，这里的-p 代表的是第几个台站。主要关注速度三份量的波形，如果没有频散（没有毛刺），而且不是全为 0，一般就是没问题的。

2.2 批量震源计算

波形模拟是以每一个虚拟源为单位进行单独模拟的，也就是说，每一个虚拟源都会有一个与 template 相类似的文件夹，里面的内容除了 SeisSource.conf 不同，其他的都与 template 相同。因为不同的源共用介质、网格、台站。我们建立 template 其实就是为了减少工作量，以其为模板，复制其他不同震源的文件夹。

（1）**精简 template 文件夹。**

先删除调试阶段 template 中 output 文件夹中的所有内容，但保留该文件夹。然后删除 template 中 input 文件夹中生成的与震源相关的文件，即 rm input/source*.nc，最后**再次确认** template 下 runALL.sh 中注释前 5 步，仅打开第 6 步。

（2）**由 template 复制其他震源文件夹。**

具体操作为：./copy.sh， 即可复制 src1，src2，src3，…，src40 这些文件夹，注意修改 copy.sh 中 for 循环的起止参数。

（3）**批量修改震源位置**

首先保证虚拟震源文件 virsrc.dat 已经上传，并和 template 以及 src*这些震源文件夹在同一级目录下，然后更新不同震源文件夹下 SeisSource.conf 中的震源位置。具体操作为./update_source.sh，注意修改 update_source.sh 中 for 循环的起止参数。

（4）**批量加载震源**

对于不同的震源文件夹，加载震源，具体操作为：./excute_source.sh。注意修改 excute.sh 中 for 循环的起止参数。

（5）**批量提交作业脚本**

具体操作为：./submit_job.sh，注意修改 for 循环中的起止参数。建议每次最多提交 10 个任务（如果每个任务是 200 核）。然后通过 bjobs 命令参看提交任务的运行情况，如果均在等待，即处于 pend 的状态，那么暂缓提交任务。太乙服务器规定**每个用户处于等待的任务总核数不超过 2000 核**，但对于处于运行的任务总核数并没有限制。

二、速度模型统计评估

在做这一步之前，建议将太乙服务器上计算好的所有波形数据下载到本地（因为太乙服务器/scratch 下的数据仅保存 10 天），实际需要下载的东西包括 template 文件夹、以及每个震源文件夹下的 output 文件夹以及 SeisFD3D.conf。可以在本地先建立相应的震源文件夹 src1,src2,…,src60，然后通过 datalink.sh 从太乙服务器上将数据下载下来。

1. 单震源下基于互相关计算走时延迟和相关系数(程序包:pyplot)

将合成波形与观测波形进行振幅归一化后互相关，就会得到前者相对于后者的走时延迟（traveltime delay），然后将二者的走时延迟校正后，计算二者之间的互相关系数（cross-correlation coefficient）。

（1）**将真实互相关数据格式转存为.h5**（程序：ReadCcfs2h5.py）

在目前波形比较的程序中，真实的互相关数据的输入格式是.h5，程序读写都是按照.h5 进行的，如果真实的互相关函数是通过 CC-FJpy 计算的，那么就直接可以利用 ReadCcfs2h5.py 进行转换。

（2）**画图（波形比较，**程序：figsgf6.py）

在某个震源文件夹下，运行./pyplot/figsgf6.py -s 1 -d 4，这里的 s 代表第几个震源，d 代表步长，也就是隔几个画一条波形。程序中需要注意的输入参数如下：

```
putinrou = ['/home/cjq/ModelValid/Forward/CCF/CFs/SCB_3600_0.9_summed.h5']
syn_stalist = '/home/cjq/ModelValid/Forward/SCB_locat_v2.txt'
sele_stalist = '/home/cjq/ModelValid/Forward/SCB_locat_v2.txt'
srclist = '/home/cjq/ModelValid/Forward/virsrc.txt'
inputpath = '/home/cjq/ModelValid/Forward/template/model_Chen/input/'
```

putinrou:真实的互相关数据.h5;

syn_stalist:用来进行波形模拟合成理论波形数据的台站列表;
sele_stalist:从所有计算真实互相关数据的台站中挑选出的进行波形比较的台站列表;
srclist:所有虚拟震源的台站位置列表;
inputpath:template 中的 input 文件夹。
这里需要注意的是，syn_stalist 和 sele_stalist 中的台站数量可以不相同。

另外这里需要注意两点，第一、波形验证是分周期进行的，需要设置不同的周期范围对合成波形与真实数据进行加窗，然后在每个周期窗内进行比较，目前程序设置了四个相互重叠的周期范围，如下图：

```
#===========windowed waveform=====================================
fs_obs = 1.0/(tt[1]-tt[0])
bands = np.array([[3,7],[5,10],[8,15],[12,20]])
aver = np.array([3.12,3.12,3.12,3.30])
```

这里的 bands 可以自己设置，但一般大尺度下这四个周期范围是合适的，另外需要说明的是，这里的周期范围也不是随意设置的，其上下限也是由模拟时所用子波的主频确定的，如果要在偏离目前设置太远（过高或者过低）的周期范围进行比较，需要重新设置波形模拟的子波，重新合成波形数据。

第二、这里的 aver 设置的是每个周期范围下的平均速度，目的是要估计每个周期范围下信号窗的位置，然后进行加窗置零操作。一般情况下不用修改，但是特殊情况下还是需要观察实际互相关数据在不同周期范围下的波形确定这个平均速度.

（3）**计算单个震源下的走时差和相关系数**（程序：misfit_cal.py）

Misfit_cal.py 其实和 figsgf6.py 整体思路是一样的，所以大部分代码基本相同，只不过将 figsgf6.py 中的画图部分改成了计算走时差和相关系数的部分。运行：./pyplot/misfit_cal.py -s 1 -d 1

这里的计算思路是对于单个震源，从该震源到其他台站会形成多条射线路径，先把整个速度模型的尺度离散成一个一个的小网格，然后统计出每条射线经过的网格点，每条射线路径都会计算出一个走时差和相关系数，就把这个走时差和相关系数投射到这条射线路径经过的所有网格点上。

misfit_cal.py 的输入参数和 figsgf6.py 是一样的，在这里要注意 misfit_cal.py 中引入了一个确定射线路径经过哪些网格点的函数 ray_cross_path。如下图：

```
def ray_cross_path(stloc, edloc, dx):
    #---------------------------------------
    # stloc: start location (lon, lat)
    # edloc: end location (lon, lat)
    # dx: interval for longitude and latitude
    #---------------------------------------

    #print("stloc: ", stloc)
    #print("edloc: ", edloc)
    xyn = []
    X0 = 108; Y0 = 20
    idx1 = int(np.ceil((stloc[0]-X0) / dx))
    idx2 = int(np.ceil((edloc[0]-X0) / dx))
    idy1 = int(np.ceil((stloc[1]-Y0) / dx))
    idy2 = int(np.ceil((edloc[1]-Y0) / dx))
```

注意修改这里的 X0,Y0，这是速度模型的最小经度与最小纬度，相当于锚点。程序中会把这个（X0,Y0）当作起始点，重新对网格点的位置进行编码，具体编码的程序见 idx1,idx2,idy1,idy2 的计算。

另外程序里可以修改的参数包括 xlimt，lag_thre，ccc_thre。如下图。

```
xlimt = 300 #s
lag_thre = 6 #s
ccc_thre = 0.6
```

xlimt：对合成波形数据按照观测波形的采样时间间隔进行插值的时间长度（合成波形与观测波形进行互相关比较，就需要对齐时间点，而合成波形的采样时间间隔和观测波形的采样时间间隔一般是不一样的，所以就需要重新对合成波形按照观测波形的采样时间间隔进行插值，然后再与观测波形进行互相关比较。）

lag_thre:走时延迟的阈值。大于这个值的射线路径被舍弃。

ccc_thre：相关系数的阈值。小于这个值的射线路径被舍弃。

（只有同时满足 lag_thre 与 ccc_thre 阈值条件的射线路径才会被计入统计。）

最后的输出文件是以周期范围命名的文件，如 cell_3_7s.txt，记录了 3—7s 范围内不同网格对应的走时延迟和相关系数。格式为“网格横向编号 网格纵向编号 走时延迟 相关系数”，文件中一般会出现同一个网格对应多组数据的情况，这是因为可能同一个网格会出现在多条不同的射线路径下，这也是后面对网格点进行统计评估的基础。

注：在调试好一个震源文件夹下的 figsgf6.py 与 misfit_cal.py 后，通过 MAD_CCC_regionalize 程序包中的 excute.sh 批量操作进行画图和计算。

2. CT_MAD 与 CCC 的统计评估（程序包：MAD_CCC_regionalize）

(1)提取所有单个震源文件夹下的网格点数据，对于不同的周期范围，以网格点命名，重新储存走时延迟和相关系数。(程序：evaluate0.py)，运行命令：./evaluate0.py，注意修改程序中的文件路径。最后会输出以不同周期范围命名的文件夹，如 dir_3_7s，储存了以不同网格点命名的.txt 文件，如 17_17.txt，里面的数据格式有两列，第一列是走时延迟，第二列是相关系数。

(2)计算每个网格点对应的走时延迟的平均绝对偏差（mean absolute

deviation of traveltime delay, CT_MAD）和平均相关系数（mean correlation coefficient,CCC)。(程序：evaluate1.py)。运行示例：./evaluate1.py -p '3_7s' ，代表计算周期范围 3-7s 下每一个网格点对应的 CT_MAD 和 CCC。最后会输出以周期范围命名的.txt 文件，如 aver_3_7s.txt，里面的数据格式有四列，从左到右依次是经度，纬度，走时延迟，相关系数。

(3) 画图。画出不同周期范围下所有网格点的 CT_MAD 和 CCC。可以在画图之前对上一步得到的每个周期范围下的平均数据进行插值，这里给出了两个参考插值程序，InterpForMAD.m 和 InterpForCCC.m，同样需要注意的是这里使用的 kriging 程序有时会不稳定（插值结果为 0 或者说是 NAN，但多数时候还是稳定的），可以改成 matlab 中自带的 interp2 函数。最后利用 MAD_CCC_distribution.py 对插值后的 CT_MAD 和 CCC 进行画图。最后输出 MAD.png 和 CCC.png。

需要注意的是 MAD_CCC_distribution.py 程序中用的是 pygmt 库，不是 gmt。另外，里面用到的 topo30.grd 数据因为文件太大，所以就没有放在程序包中，可以去组里服务器 seislab 上获取，其在 seislab 上的路径为：/home/data/Topography/SRTM/topo30.grd。

三、附：额外需要注意的问题

1. parfile 中 Vsim_fortran_new.nc 中的变量 depth 要从 0 开始。

2. 波长快照的设置。

```
# final snap output
number_of_snap = 1
# id         start       count      stride  tinv tdim/nc cmp
# fix zeta
snap_001 =  1 1 140    560 360 1     1 1 1  1 1000000 V
#snap_004 =  1 1   1    1080 720 1     1 1 1  1 10000 V
```

Snap_001 对应了 id；start 为起始点的网格点编号，在这个例子中 x，y，z 三个方向的网格数分别为 560，360，140。Stride 为网格步长，count 为 x，y，z 三个方向跑的步数，总起来讲，x 方向为：start_x : stride_x: stride_x*count_x，这里为 1：1：1*560，y 与 z 方向以此类推。

3. Vs，Vp，rho，这些重要的模型参数尽可能不要有明显的跳跃界面，如果有的话，也会出现在一定时间段内波场直接爆掉的情况，数值上就是 NAN。

4. 出 bug 如果实在找不到问题，很大原因就是介质的问题，而常见的问题是介质中会出现一些明显的条带状速度变化，这时候要把它们给平滑掉，平滑是非常管用的，可以插值和平滑组合来用。