

# 基于 Specfem 3D Cartesian 噪声伴随成像 Python 框架的使用说明

赵 杨 [zhaoyanginsane@foxmail.com](mailto:zhaoyanginsane@foxmail.com)

2021 年 12 月

**注意：**所有的参数选择和流程都按照官方 manual 实现，因此请认真阅读 [manual\\_SPECFEM3D\\_Cartesian.pdf](#)。编辑 Python 代码的 IDE 建议使用 vscode 或 pycharm，vscode 比较适合学术使用，支持很多语言包括 shell；pycharm 在工业界用的比较多，python 集成性也比较好，可以下载商业版（用 edu 的邮箱可以免费使用一年），本次所有 Python 都在 pycharm 编写实现。Python 版本选用 3.6 及以上即可。

**使用本代码框架时请引用：**

- 1) Zhao, Y., Guo, Z., Wang, K., & Yang, Y. J. (2021). A Large Magma Reservoir Beneath the Tengchong Volcano Revealed by Ambient Noise Adjoint Tomography. *Journal of Geophysical Research: Solid Earth*, 126(7), e2021JB022116.
- 2) Wang, K., Yang, Y., Basini, P., Tong, P., Tape, C., & Liu, Q. (2018). Refined crustal and uppermost mantle structure of southern California by ambient noise adjoint tomography. *Geophysical Journal International*, 215(2), 844-863.
- 3) Fan, X., Guo, Z., Zhao, Y., & Chen, Q. F. (2022). Crust and uppermost mantle magma plumbing system beneath Changbaishan intraplate volcano, China/North Korea, revealed by ambient noise adjoint tomography. *Geophysical Research Letters*, e2022GL098308.

## 1. 背景介绍

从 20 世纪 90 年代起，相当多的科学家开始着手开发和推广基于伴随成像的相关软件包。最初的伴随成像软件包 SPECFEM3D 是由法国巴黎 Institut de Physique du Globe 的 Dimitri Komatitsch 和 Jean-Pierre Vilotte 在 1995 年至 1997 年期间联合开发的，然后由哈佛大学和美国加州理工学院的 Dimitri Komatitsch 和 Jeroen Tromp 在 1998 年继承并发展该方法。自此，越来越多的学者开始完善和维护 SPECFEM3D 软件包，相关代码和工具可在 <https://geodynamics.org/cig/software/specfem3d/> 下载获得。SPECFEM3D 软件包目前针对研究区尺度大小已有两个版本：SPECFEM3D\_Cartesian 和 SPECFEM3D\_Global。SPECFEM3D\_Cartesian 主要针对的是小尺度或区域尺度的研究区，它忽略了地球曲率对于 UTM 坐标投影带来的误差。本代码框架采用的是 SPECFEM3D\_Cartesian。

SPECFEM3D\_Cartesian 基于谱元法(Spectrum Element Method, SEM)进行波场模拟,能充分地考虑了复杂的三维地球模型,如地形的起伏、壳幔内的不连续面、各向同性、各向异性、振幅的衰减等,反演的参数涵盖了剪切波速度、压缩波速度、密度、各向异性、重力。同时,它使用 MPI 接口(Message Passing Interface,即信息传递接口)实现多个 CPU 或 GPU 之间跨节点的信息传递,以达到伴随成像算法能在并行计算的框架下运行,极大地缩短了计算时间。SPECFEM3D\_Cartesian 程序包除了伴随成像的主程序包,还包含如 FLEXWIN、球坐标和大地坐标互相转换的脚本、地下三维网格(mesh)的生成包、正演数据预处理包、方法演示的样例等。SPECFEM3D 将复杂的伴随算法成功地集成到一个程序包里,其代码书写非常干净、框架简洁、命名规范合理,计算能力和效率也十分突出,尤其是对于地震学上的应用有着杰出的贡献,因此 SPECFEM3D 在 SuperComputing 2003 会议上获得了超算界的诺贝尔戈登贝尔奖。

## 2. 软件安装

SPECFEM3D\_Cartesian 程序包放置在 [specfem3d/](#)下。脚本 [install\\_all.sh](#) 可供安装参考。

- 1) 首先,安装时要确认服务器是否安装 1、ifort、icc、mpif90 或者 2、gfortran、gcc、mpif90,在环境配置时需要用到这些包。
- 2) configure 没报错,证明环境搭建成功,若有错误请仔细检查 1) 的步骤。
- 3) 接下来就是编译,make all 成功则编译完成,编译好的可执行程序在 [specfem3d/bin/](#) 下面。
- 4) 假如无论怎么调试 make all 都报错,就单独 make 几个主要的程序就行,包括 xmeshfem3D、xgenerate\_databases、xspecfem3D 等,它们分别是网格的构建、初始模型、波场模拟。

## 3. 噪声互相关

噪声互相关的程序 ANDP(Ambient Noise Data Processing)由成都理工大学范兴利博士([fanxingli@cdut.edu.cn](mailto:fanxingli@cdut.edu.cn))完成,源程序和实例说明都放置在 [ANDP-2.3](#)。ANDP 是基于 mpi 的并行计算框架,可以大大缩短传统噪声互相关的计算时间。本人曾采用 ANDP 对 365 个地震台进行噪声数据处理,记录时间长度超过 2 年半,使用 144 核同时对所有台站进行互相关计算,计算时间的开销不足 10 小时。

在 [ANAT/scripts/data\\_processing/](#)下提供了从 ANDP 计算完成的互相关数据到噪声伴随成像所需数据的格式转换脚本,可供参考。互相关结果放置在 [ANAT/data/](#)下。

## 4. 噪声伴随成像程序包

噪声伴随成像 Python 程序包放置在 ANAT/下。 [Parameters .py](#) 是成像过程中可能会修改的参数， [01,02,03....py](#) 是流程脚本。 [data/](#) 是格式修改完成后的互相关结果； [fwd\\_test/](#) 是一个事件的正演测试，建议在做正式伴随计算前首先完成 [fwd\\_test/](#) 确定参数的选择； [scripts/initial\\_model/](#) 是关于构建初始模型的脚本，可供参考； [parameter\\_files/](#) 是伴随成像参数配置文件； [scripts/](#) 包含流程中使用到的脚本； [src\\_rec/](#) 放置激发台站(sources)和接收台站(receivers)的信息； [tools/](#) 是流程中用到的程序包。接下来会按照参数设置顺序详细介绍每一模块。

**注意：建议不要修改文件的命名方式，以免程序报错！**

### 4.1 网格设置

伴随成像的第一步是生成基于 SEM 算法的地下三维网格。对于网格生成工具的选择，目前有两种主要途径：(1)利用外部软件 CUBIT 生成网格 (<http://cubit.sandia.gov>)。CUBIT 是一个集成较好的网格生成软件，可用于为任意形状模型创建有限元网格，经验表明使用 CUBIT 将极大地简化和加快了各种复杂的地球物理模型网格的生成和准备。当 CUBIT 生成网格文件后将格式改为 SPECFEM3D\_Cartesian 的输入文件形式即可。(2)使用 SPECFEM3D\_Cartesian 中的网格工具 [specfem3d/bin/xmeshfem3D](#) 作为 CUBIT 的替代方案。由于 CUBIT 需要外部接口产生，不利于用户直接使用 SPECFEM3D\_Cartesian，而 [xmeshfem3D](#) 在相对简单网格模型上的使用效果与 CUBIT 一致。考虑到算法框架的简洁性，噪声伴随成像的网格生成方式采用 [xmeshfem3D](#)。

网格参数的设置在 [parameter\\_files/DATA/meshfem3D\\_files/](#) 下。主要包括 [interfaces.dat](#) 和 [Mesh\\_Par\\_file](#)。

#### (1) [interfaces.dat](#)

界面的配置文件，**注界面是从模型的底部往上依次给出**，如先给出 moho，再给出 sediment 底界面，最后是 topography。如果 moho 和初始速度模型不是一套模型建议不使用 moho。试验表明是否有 moho 的模型正演出的波形相差不大。同样沉积层不精确可以不用给定。buffer layer 是对地形进行平滑的结果，它可以抑制由于剧烈地形造成的网格变形。陡峭的网格使得 [DATA/Par\\_file](#) 中正演参数 dt 太小从而让计算时间变得非常大。在执行完 [fwd\\_test/04\\_MeshDatabases.py](#) 中 [xmeshfem3D](#) 时会产生 [Station\\_name/OUTPUT\\_FILES/output\\_meshfem3D.txt](#)，如果 [output\\_meshfem3D.txt](#) 里 histogram of skewness 显示网格质量是 bad，可考虑加入 buffer layer。buffer layer 放置在地形下，是地形的一个 smooth 版本的界面（计算方式可参考  $(\text{topo} - \text{Mean\_topo}) * \text{scale}$  系数）。此外，如果存在浅层的低速层（沉积盆地），可在低速层使用 DOUBLINGS 的方式，即在低速层附近加厚一层，相关参数在 [Mesh\\_Par\\_file](#) 中的 Regular mesh。

参数设置从左往右依次是：是否是 UTM 的界面数据（一般界面数据都是经纬度坐标系下，所以设置是 false）、X 方向网格数（经度）、Y 方向网格数（纬度）、界面经度起点、纬度起点、X 方向网格间距、Y 方向网格间距。最下方的 layer number 是 Z 方向每一个 region 划分的层数，从下往上给出。如首先模型底界面和 moho 之间分 2 层，moho 和沉积层底界面分 8 层，沉积层到地形之间分 1 层（前提是这些界面都使用的情况），地壳内建议划分到 5km 一层，地幔 10km 一层。界面数据放在 moho.dat、topo.dat、buffer\_layer.dat 里，数据只有一列（先对经度循环，再对纬度循环，保存一列界面深度数据，向上为正，单位为米）。

## (2) Mesh\_Par\_file

- 1、前五个是模拟区域所在的经纬度和深度范围，建议模拟区域要略大于研究区（即最终成像范围），因为波形传播可能存在边界效应。
- 2、UTM\_PROJECTION\_ZONE 根据研究区所在的 ZONE 不同给出相应的值，如腾冲地区的 ZONE=47，川滇=48。
- 3、NEX\_XI、NEX\_ETA、NPROC\_XI、NPROC\_ETA 的设置非常重要，建议参考官方 manual。NEX\_XI、NEX\_ETA 是模型 X 和 Y 方向的网格数，它的分别值等于  $8 \times (\text{NPROC\_XI or NPROC\_ETA}) \times C$ ，C 是正整数，NPROC\_XI\*NPROC\_ETA 等于波形模拟中使用的总核数。核数越多计算越快，但资源占有也越多（即排队时间会更长）。此外，网格间距建议保持在几公里 km 级别（经纬度/NEX\_XI、NEX\_ETA），**波形最短模拟的最短周期=  $(288 / \text{NEX\_XI}) * 2$** 。
- 4、USE\_REGULAR\_MESH 不使用 DOUBLINGS 时为 true，否则为 false。DOUBLINGS=11 指的是从下往上数第 11 层进行加厚一层。
- 5、NMATERIALS 和 interfaces.dat 的 interface 个数相等，在 NMATERIALS 下给出的是被 interface 划分的每一个 region 的参考数值。
- 6、NREGIONS 和 interfaces.dat 的 interface 个数相等，在 NREGIONS 是每一个 region 的起始和终止的网格点。

## 4.2 初始模型

初始模型放置在 [parameter\\_files/DATA/tomo\\_files/tomography\\_model.xyz](#)。[tomography\\_model.xyz](#) 的数据格式按照 [manual\\_SPECFEM3D\\_Cartesian.pdf](#) 图 14-1 中的方式给出。在 [initial\\_model/](#)下给出了可供参考的初始模型构建脚本。由于初始模型的坐标系是 UTM，因此参考脚本的构建思路是(1)计算出模拟区域的 UTM 范围 (xmax,xmin,ymax,ymin)；(2)构建 UTM 三维网格得到 coordinates\_utm.dat；(3)将 coordinates\_utm.dat 转化为球坐标系(经纬度)的三维网格得到 coordinates\_geo.dat；(4)将前人模型(一般情况下是经纬度的坐标系统)插值到 coordinates\_geo.dat 上，并根据波速比(1.73)和速度-密度经验公式计算得到

P 波速度和密度；(5)此时的三维介质模型是在经纬度坐标系下，需要将经纬度替换成 UTM，由于步骤(3)中 UTM 坐标和经纬度坐标是一一对应的，因此可直接用 UTM 直接替换掉经纬度，可得到 UTM 下的三维介质模型；(6)最后按照 [tomography\\_model.xyz](#) 的数据格式保存。

上述构建思路的目的是产生等间距规则网格的 UTM 初始模型，如果直接将前人的经纬度模型转化为 UTM 会使得初始模型不是等间距和规则的。**注意：**在 [tomography\\_model.xyz](#) 上方还需给出坐标轴的最大最小值，物性参数的最大最小值，脚本 [MinMax\\_value.bash](#) 可计算出每列的最大与最小。

### 4.3 参数设置

在 [parameter\\_files](#) 文件下有三个文件夹，[bin/](#)下放置编译好的程序(在 [02\\_ConfigSimulation.py](#) 中自动实现)；[DATA/](#)下放置界面参数和模拟参数；[OUTPUT\\_FILES/](#)下放正演的波形信息，[OUTPUT\\_FILES/DATABASES\\_MPI/](#)下是每个核的网格、模型（vp、vs、rho 等）和 kernel 等数据。[DATA/](#)下需要修改的参数较多，建议认真阅读官方 manual，以下是 [DATA/](#)参数说明：

- 1) [FORCESOLUTION](#) 是噪声成像中 master station 模拟点源力的文件，会在脚本 [01\\_ConfigSrcRec.py](#) 中自动生成。
- 2) [Par\\_file](#) 是关于正演和伴随计算的参数，这里只给出一些重要的参数，其他请在官方 manual 中查找。
  - 1、[SIMULATION\\_TYPE](#) 的值在每次迭代中会修改，不用单独修改；
  - 2、 $NPROC=NPROC\_XI*NPROC\_ETA$ ；
  - 3、DT 的数值是运行 [xgenerate\\_databases](#) 中产生的 [OUTPUT\\_FILES/output\\_meshfem3D.txt](#) 里，例如‘Maximum suggested time step for simulation = 0.04019591’，DT 的数值不能大于 0.04019591，可设定为 0.04。DT\*NSTEP 是可以模拟的波形长度，假设 NSTEP=500，那么波形长度是  $0.04*500=200s$ 。DT 可在 4.5 节中 [fwd\\_test](#) 中确定。
  - 4、MODEL 这个参数是在脚本里自动修改，使用初始模型为 tomo，在进行第二次迭代后改为 gll。

[DATA/Par\\_file](#) 的参数几乎不用手动修改，所有可变参数都写入到了 [\\_\\_Parameters\\_\\_.py](#)，当用户设置好 [\\_\\_Parameters\\_\\_.py](#) 值时，运行 Python 脚本会自动修改。[\\_\\_Parameters\\_\\_.py](#) 中给出了每个参数详细的注释说明，需要注意的是 dt 和 channel 需要进行一次正演计算才能确定，因此在正式计算前会进行 [fwd\\_test](#) 一个事件的测试，在 4.5 节中会详细介绍。

### 4.4 sources 和 receivers

将所有台站信息放在 [src\\_rec/sources.dat](#) 里，从左到右每列依次是台站号、台



网号、纬度、经度、0、0。运行脚本 [01\\_ConfigSrcRec.py](#) 得到每个台站的 FORCESOLUTION、CMTSOLUTION 和 STATIONS, 其中 FORCESOLUTION 是点源力的参数, 而 CMTSOLUTION 是地震震源的参数, 由于噪声成像没有用到天然地震震源, 因此 CMTSOLUTION 中的参数可以设置为默认。STATIONS 是以一个台站为激发源, 列出其他接收台站的文件。

#### 4.5 fwd\_test

[fwd\\_test](#) 是一个事件的完整正演计算和残差分析, 其目的是检查数据、初始模型、参数设置是否正确, 还可以确定 [\\_\\_Parameters\\_\\_.py](#) 中 `dt` 和 `channel`。所有事件的伴随计算开销巨大, 因此建议在正式伴随计算前首先运行 [fwd\\_test](#) 确定参数模型是否正确。[fwd\\_test](#) 包含 14 个脚本:

1. [01\\_ConfigSrcRec.py](#) 首先配置 [DATA/Par\\_file](#) 下的总核数 `NPROC`, 其次配置了 `sac` 路径, 然后生成了 [src\\_rec/sources.dat](#) 里以第一个台站为激发源的地震事件, 包括 FORCESOLUTION、CMTSOLUTION 和接收台站 STATIONS。
2. [02\\_ConfigSimulation.py](#) 首先将编译好 [specfem3d/bin/](#) 下的可执行程序放入事件 `bin/` 下, 其次配好其他所需文件, 最后修改 [DATA/Par\\_file](#) 里初始模型的选择为 `tomo`。
3. [03\\_ConfigPreMeaAdj.py](#) 是配置好正演波形和数据的预处理脚本(在 [pre\\_proc/](#) 下), 以及残差计算的脚本(在 [measure\\_adj/](#) 下)。其中 [pre\\_proc/](#) 将正演位移转化为 `sac`, 对实际数据求负导数得到经验格林函数, 并对理论和实际波形进行滤波到目标周期段。
4. [04\\_MeshDatabases.py](#) 提交任务至服务器进行网格的计算([bin/xmeshfem3D](#))和初始模型([bin/xgenerate\\_databases](#))的构建。
5. [05\\_ChooseDt.py](#) 由第四步得到最大的 `dt`, [\\_\\_Parameters\\_\\_.py](#) 选择的 `dt` 要小于最大 `dt`。手动修改 [\\_\\_Parameters\\_\\_.py](#) 中的 `dt`, 并根据研究区大小确定 `NSTEP`。
6. [06\\_ModifyParafire.py](#) 将修改后的 `dt` 和 `NSTEP` 写入 [DATA/Par\\_file](#)。
7. [07\\_Get3dVTK.py](#) 得到三维网格和初始模型的 `VTK` 文件, 该文件可由 Paraview 打开, 并能进行三维图像的绘制。
8. [08\\_GetModels.py](#) 利用 `tools/model_slice` 将每个核上的二进制文件 `vs.bin` 合并成 ASCII 的文件, 为成像画图做准备。包括三个部分: (1) `sem_model_slice.f90` 的编译; (2) 成像的经纬度网格; (3) 并行合并二进制文件。
9. [09\\_PlotModels.py](#) 使用 GMT5 进行画图, 确定初始模型是否正确。
10. [10\\_Fwd.py](#) 进行波场正演计算([bin/xspecfem3D](#))。
11. [11\\_ChooseChannel.py](#) 在第 10 步完成后可以在 [OUTPUT\\_FILES/](#) 得到每个接收台站的位移文件, 根据位移文件的命名方式获取 `channel` 名, 再手动修改 [\\_\\_Parameters\\_\\_.py](#) 中的 `channel`。

12. [12\\_PreMea.py](#) 进行波形预处理和残差测量。其中理论波形的滤波结果在下 [OUTPUT\\_FILES/](#) , 实际数据的滤波结果在下 [pre\\_proc/M00/Station\\_name/DATA\\_NORM/](#)下。残差结果放置在 [misfits/](#)下。
13. [13\\_PlotMisfits.py](#) 是绘制不同周期段的残差分布直方图。
14. [14\\_PlotWaveforms.py](#) 是绘制不同周期段的实际数据(黑色曲线)和当前模型(红色曲线)的波形图。

## 4.6 噪声伴随成像

完整噪声伴随成像的正演脚本([01-09.py](#))与 [fwd\\_test](#) 大致相似, 不同之处在于是对所有事件进行正演计算。为了节省计算时间, 将所有台站分成了若干个 `set`, 每个 `set` 的台站数量大致相同, 然后将每个 `set` 当做是一个 `job` 提交至服务器。例如一共有 164 个地震台站, 将其平均分成 11 个 `set`, 每个 `set` 有 15 个台站, 最后一个 `set` 有 14 个台, 当计算资源足够时, 可以同时 11 个 `set` 进行计算, 可以极大缩减计算时间。

[09\\_PostProConfig.py](#) 和 [10\\_PostPro.py](#) 是进行敏感核的计算得到 `gradient`, 包括叠加、预处理因子和光滑, 最后进行 [Parameters.py](#) 中的 `step_range` 进行模型更新, 得到不同步长下的模型参数。

[14\\_LineSearch.py](#) 是对每个步长下的模型参数进行正演、波形预处理和残差计算; [15\\_CheckLineSearch.py](#) 是检查每个步长下的线性搜索正演是否完成; [16\\_LineSearchRecompute.py](#) 是对未完成的台站进行再次计算; [17\\_ChooseStep.py](#) 是评估每个步长下不同周期段的残差去选择合适的步长, 其最优模型作为下次迭代更新的初始模型, 并画出每个步长下的残差分布。

[18\\_MvRmMod.py](#) 是移动并行计算 `log` 和 `error` 文件到指定文件夹, 并删除线性搜索的文件以节省磁盘空间。

## 4.7 Checkboard

Checkboard 的计算过程与速度反演的过程基本一致, 相关脚本放置在 [scripts/checkboard\\_test/](#)下。(1) 棋盘测试的第一步是先合成理论波形数据, 使用与真实噪声伴随成像一致的地震台站分布, 棋盘测试初始模型的生成过程与 4.2 中的一致, 其中扰动模型由 [m01\\_CreateModel.py](#) 生成, [m02\\_add\\_anomaly.m](#) 是将扰动模型加到一维初始模型上, [m03\\_construct\\_mod.m](#) 是生成等间距的 UTM 初始模型文件, 该文件作为 [parameter\\_files/DATA/tomo\\_files/tomography\\_model.xyz](#) 进行 `synthetics` 的模拟。在完成 [01.py-04.py](#) 后可得到模拟的理论波形 (其中 `is_meas` 和伴随 `adjoint` 都设置成 `False`)。然后执行 [m04\\_copy\\_SynsAsData.bash](#) 和 [m05\\_add\\_station\\_event.py](#) 将模拟的波形转化为实测数据放置在 [data/](#)下得到了所

有台站对之间的格林函数，至此检测板异常模型的实际数据生成完毕。

(2)恢复模型的计算流程与完整的噪声伴随成像一致。它将未加异常扰动的一维平均速度模型作为初始模型，与模拟的实测数据进行完整的伴随成像流程01.py-18.py(多次迭代)，其中 `is_checkboard=True`(生成的实际数据是格林函数，不需要再次求导)。最后将得到的速度模型与一维平均速度模型作比较判断恢复程度。