



FWI 基本步骤—基于Devito

Li



FWI方法

Pratt(1999)重新介绍了FWI，引起地球物理勘探的广泛关注，因为FWI可以在复杂地质环境下，实现自动化、构建高分辨率的速度模型。

FWI方法就是**迭代最小化计算**，地震勘探获取数据与应用一个地下评估速度模型初始化地震波模拟的合成数据之间的**差异**。因此，本质上FWI框架组成包括：一个正演模拟预测数据的**地震波模拟器**(wave simulator)，和一个由misfit数据更新计算速度模型的**伴随模拟器**（adjoint simulator）。

FWI的综述可参考**Virieux and Operto, 2009**。下面结合devito这个特定域语言库，分别介绍FWI的基本步骤如下。



FWI程序调研

全波形反演（FWI）是前沿的高精度地震反演模式，FWI是基于PDE约束的优化问题，计算量大，特别是在3D情况。下面从源码开放性、2D/3D、编程语言、并行模式、实用性等几个维度，对FWI程序做了调研。

Devito Devito是求解差分方程的特定域语言(DSL)库，使用Python快速生成求解差分方法（类似Firedrak和FEniCS）。

JUDI JUDI是在devito库的基础上，使用julia新一代编程语言，对于地球物理学家，具有更好的生产率。JUDI是第一个联合地球物理学家、数学家和HPC/编译器专家产出的地震反演软件框架，整合了DSL的优势和地球物理专门领域需求的编译器技术。



FWI程序调研

SeisCL

SeisCL实施时间域上的2D和3D粘弹性FWI，基于OpenCL的并行化，因此可以执行多CPU和GPU集群并行（MPI+OpenCL）。从github仓库看，SeisCL示例处于开发中，尚不完善，仓库的活跃度不高。

PyFWI

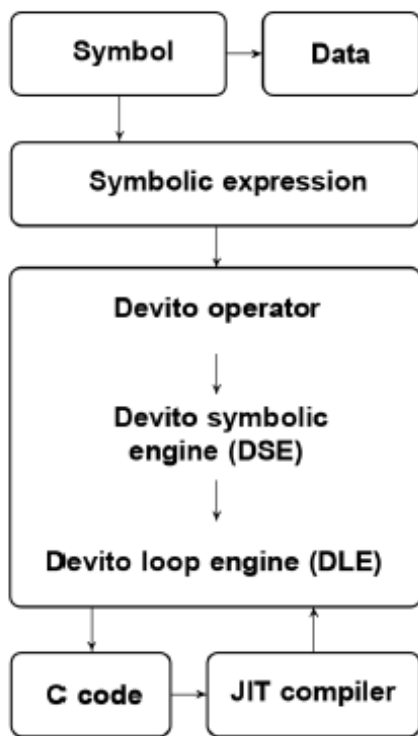
PyFWI使用Python语言，核心代码使用OpenCL的并行方式。目前仅能用于2D问题反演。

SWIT-2D-FWI

SeisFlows

小结：从程序调研看，多数的FWI属于学术研究阶段，重点是解决HPC的问题。基于devito库开发的地震反演程序，在HPC、实用性和文档及tutorial操作具有可行性。

ADSeismic.jl



```
u = TimeFunction(name='u', grid=grid)
m = Function(name='m', grid=grid)
```

High-level function symbols
associated with user data

```
eqn = m * u.dt2 - u.laplace
```

Symbolic equations that ex-
pand finite-difference stencils

```
op = Operator(expressions)
op.apply(time=ntime)
```

Automatic code generation and ex-
ecution from high-level expressions

Symbolic optimization to reduce
computation per stencil point

Loop-level optimization for
efficient parallel execution

Just-in-time compilation of
optimized C code

图1 Devito框架概览和
相关工作流程

Devito的顶层API允许用户使用携带数据的函数目标，生成符号的模板表述，可用于SymPy的符号表述。从该顶层定义，生成算子，编译，然后执行高性能的C代码。

Mathias Louboutin, et al. Devito (v3.1.0). 2019. an embedded domain-specific language for finite differences and geophysical exploration. Geosci. Model Dev., 12:1165-1187.



图13显示了迭代15次后的速度模型。对于复杂问题，如低精度的初始模型、噪声数据或现场记录的数据，需要更先进的算法和约束。但是，这不影响wave propagato

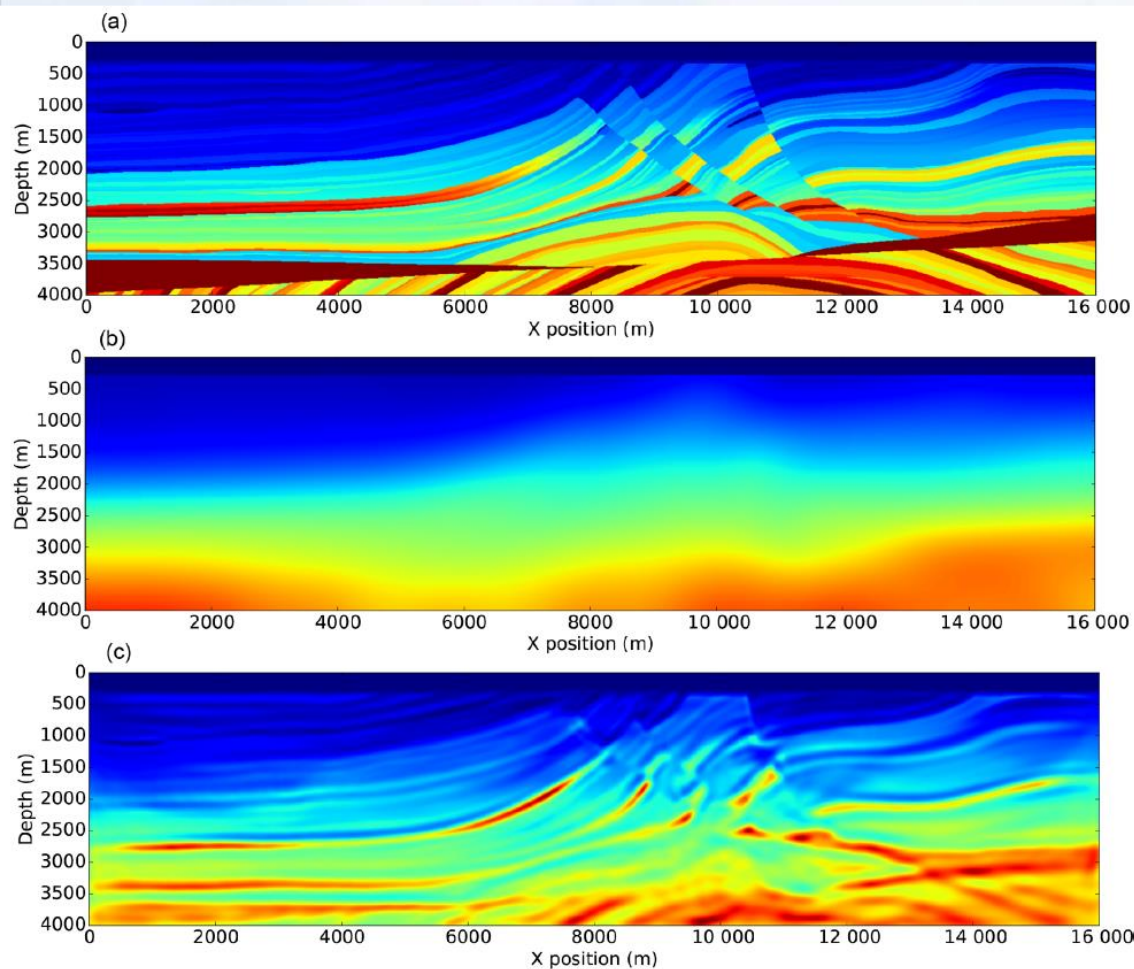


图13 Marmousi-II模型的FWI反演



(1) Full-waveform inversion, Part 1: Forward modeling

Devito使用传统的全波形层析成像的adjoint-state公式，实现Virieux and Operto, 2009介绍的FWI的大部分方法，当然还有其他方法改善质量较差的初始速度模型下FWI的表现。下面仅介绍联合forward/adjoint pair of propagators和一个基于相关性的梯度的标准计算式；讨论如何设置反演的地震波模拟，包括如何在Devito中符号表述地震波方程，以及如何处理计算域的几何。



(1) Full-waveform inversion, Part 1: Forward modeling

反演用的地震波模拟

声波波动方程， $m(x,y)=c^{-2}(x,y)$ ，其中 $c(x,y)$ 是未知的空间上变化的波速，计算式为：

$$m \frac{d^2 u(t, x, y)}{dt^2} - \Delta u(t, x, y) + \eta \frac{du(t, x, y)}{dt} = q(t, x, y; x_s, y_s)$$

其中， L 为拉普拉斯算子， $q(t, x, y; x_s, y_s)$ 是位于 (x_s, y_s) 上的地震源， $\eta(x, y)$ 是空间相关的吸收边界条件中的衰减参数。如图1，物理模型在各维度上离散为 $nbpm1$ 个网格节点，模拟无限域。 $\eta du/dt$ 在抑制层吸收地震波，防止地震波在边界上反射。在Devito中， m 和 η 的离散表征包含在`model`对象中，`model`对象包含所有相关信息的`grid`对象，诸如坐标系统原点、网格间距、模型尺寸和`time, x, y`维度。



(1) Full-waveform inversion, Part 1: Forward modeling

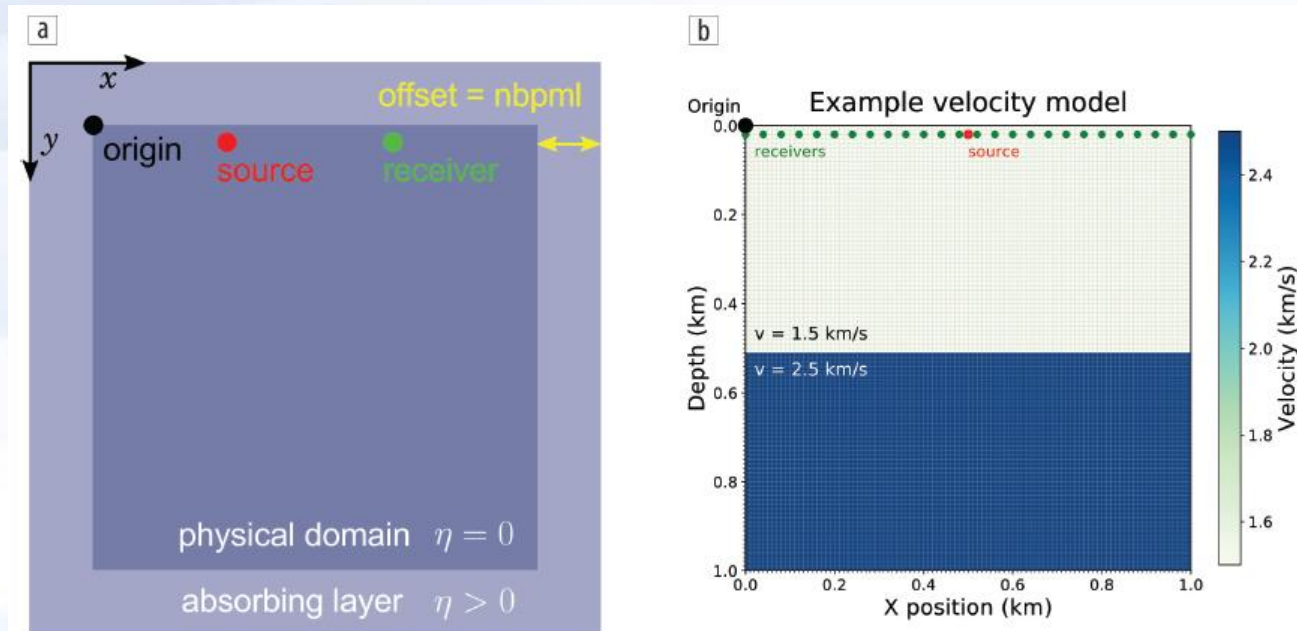


Figure 1. (a) Diagram showing the model domain, with the perfectly matched layer (PML) as an absorbing layer to attenuate the wavefield at the model boundary. (b) The example model used in this tutorial, with the source and receivers indicated. The grid lines show the cell boundaries.



(1) Full-waveform inversion, Part 1: Forward modeling

波动方程在devito中表述如下：

```
model = Model(vp=v,           # A velocity model.  
              origin=(0, 0),  # Top left corner.  
              shape=(101, 101), # Number of grid points.  
              spacing=(10, 10), # Grid spacing in m.  
              nbpml=40)       # Boundary Layer.
```




波动传播算子的符号定义

为模拟地震数据，求解声波方程，第一步就是离散求解偏微分方程（PDE），包括离散标准速度模型和波动场，以及使用有限差分空间和时间导数的近似。有限差分离散的底层手工编码是很容易出错的。Devito的主要涉及目标是允许用户基于高级的符号定义，定义复杂的无矩阵的有限差分近似，同时使用自动生成代码，创建高度优化的底层C代码。使用符号代数软件包SymPy，方便自动创建导数计算式，Devito生成高效的地震波传播模拟代码。



波动传播算子的符号定义

为展示Devito的符号计算能力，下面考虑时间相关函数 $u(t,x,y)$ 表示离散的正演波动场：

```
u = TimeFunction(name="u", grid=model.grid,  
                  time_order=2, space_order=2,  
                  save=True, time_dim=nt)
```

其中，`model`提供的`grid`对象，定义了分配内存大小，`time_order`和`space_order`定义了导数表达式的离散阶数。

现在使用这个波场的符号描述生成简单的离散的有限差分导数近似的表达式，如`u.dt`和`u.dt2`分别表示 du/dt , d^2u/dt^2 ：

```
>>> u.dt  
-u(time - dt, x, y)/(2*dt) + u(time + dt, x, y)/(2*dt)  
>>> u.dt2  
-2*u(time, x, y)/dt**2 + u(time - dt, x, y)/dt**2 + u(time + dt, x, y)/dt**2
```




计算域几何的设置

为描述时间步推进，上一节推导的式子不包含地震源函数，因此在新的时间步更新波动场仅根据之前定义的2个波场。但是，在式（1）中，地震实验的波动场通常是被一个激活的（脉冲的）的源 $q(t, x, y, x_s, y_s)$ 激发的， q 是时间和空间的函数（与波动场 u 一样）。为计算格式中包含这样的一个源项，将源在式（3）中实施为附加项：

$$u[\text{time} + dt] = 2u[\text{time}] - u[\text{time} - dt] + \frac{dt^2}{m}(\Delta u[\text{time}] + q[\text{time}])$$

RickerSource

```
f0 = 0.010 # kHz, peak frequency.  
src = RickerSource(name="src", grid=model.grid, f0=f0,  
time=time, coordinates=src_coords)
```

Src.inject函数将Ricker小波的当前时间采样（在式4中使用 dt^2/m 加权），在指定的坐标位置，注入到更新的波动场 $u.forward$ ：

```
src_term = src.inject(field=u.forward,  
expr=src * dt**2 / model.m,  
offset=model.nbpml)
```



正演

现在可以通过增加源和接收项到格子对象，定义正演传播(forward propagator):

```
op_fwd = Operator([stencil] + src_term + rec_term)
```

波动场正演模拟:

```
op_fwd(time=nt, dt=model.critical_dt)
```

完成计算后，得到的波动场存储在u.data，炮记录(shot record)存储在rec.data。可以将2D数组作图为图像，如图2。

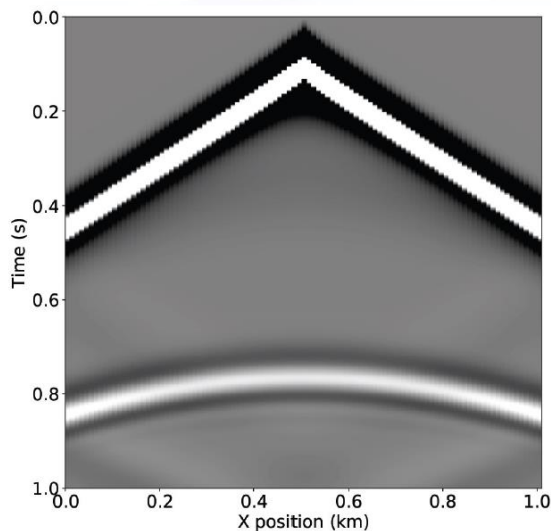


Figure 2. The shot record generated by Devito for the example velocity model.



正演

正演的波动场的snapshot做动画，如图3。

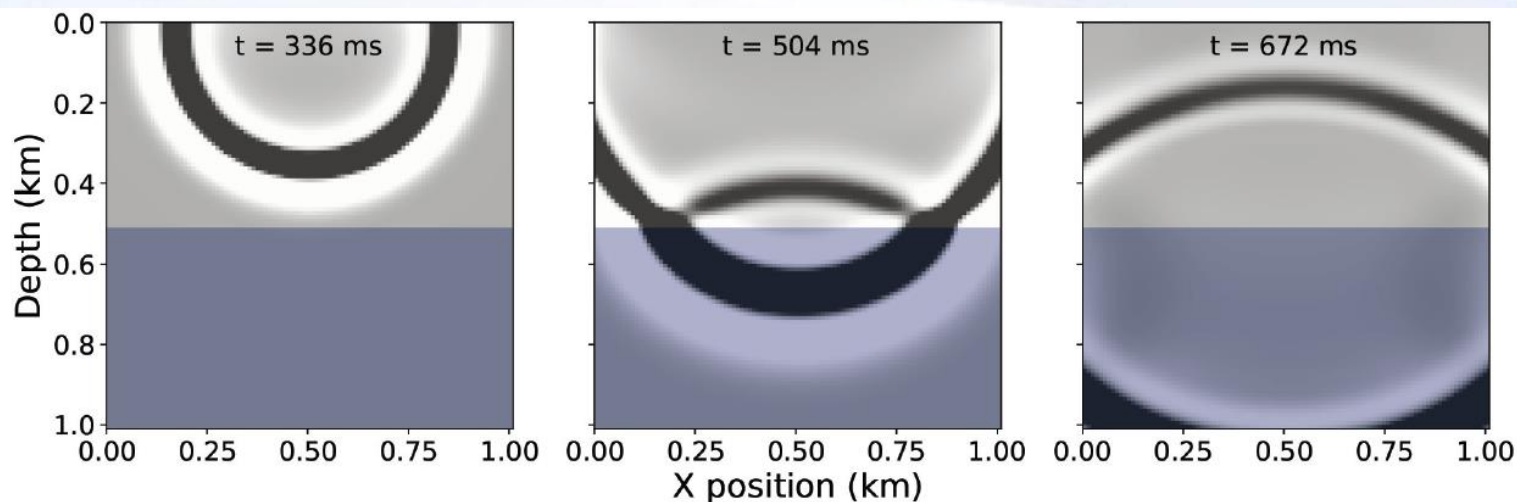


Figure 3. Three snapshots from the wavefield simulation that resulted in the shot record in Figure 2. You can generate an animated version in the Notebook at github.com/seg.



(2) Full-waveform inversion - Part 2: adjoint modeling

下面演示如何使用Devito设置和求解伴随波动方程，基于此，计算梯度和FWI目标函数的函数值。

通常，通过伴随状态法(adjoint-state method)-交叉相关正演和伴随波动场，并对所有时间步的贡献求和-计算FWI的梯度。计算一个震源位置处的梯度由3步组成：

- 1、求解正演波动方程，创建一个炮记录。必须存储随时间变化的波动场，供第3步使用。这里可使用如subsampling技术降低内存要求。
- 2、计算预测数据和观测数之间的数据残差（或misfit）
- 3、使用数据残差作为源，求解对应的离散伴随模型。在伴随（逆）时间循环内，交叉相关伴随波场的2阶时间导数与正演的波动场。求和这些交叉相关形成梯度。



(2) Full-waveform inversion - Part 2: adjoint modeling

一个简单的实验

小型地震波传播实验：一个中心具有环形高波速的恒定速度模型，如下图在模型的左边放置21个地震源，在右边放置101个接收器。

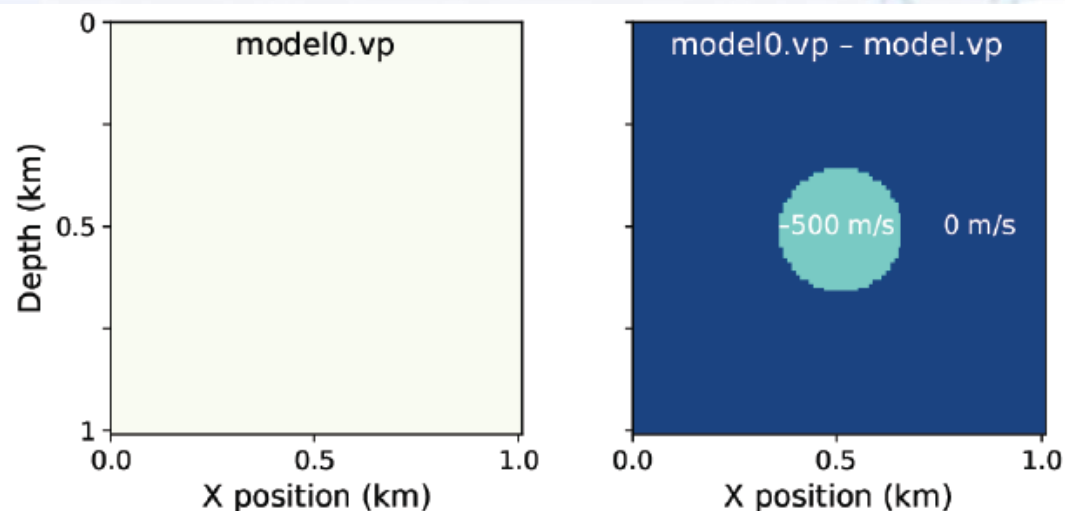
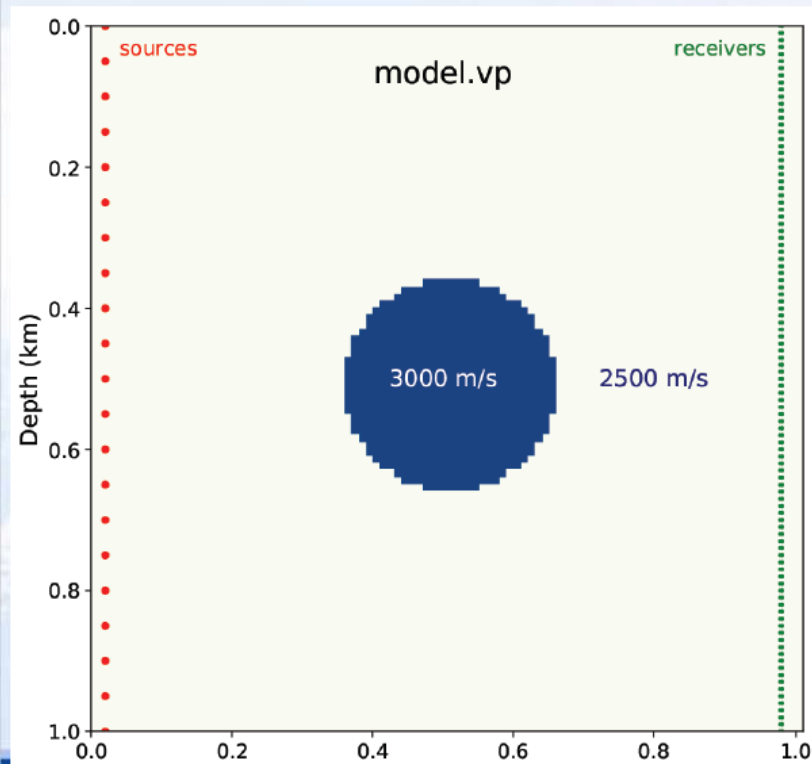


Figure 1. (a) The velocity model, with sources and receivers arranged vertically. (b) The initial estimate. (c) The difference between the model and the initial estimate.



(2) Full-waveform inversion - Part 2: adjoint modeling

伴随波动方程

使用正演部分定义的变量和位于 (x_r, y_r) 处的数据残差 $\delta d(x, y, t; x_r, y_r)$

作为伴随源，给出连续的伴随波动方程：

$$m(x, y) \frac{d^2 v(t, x, y)}{dt^2} - \Delta v(t, x, y) - H(t, x, y) = \delta d(t, x, y; x_r, y_r).$$



(2) Full-waveform inversion - Part 2: adjoint modeling

伴随波动方程

该展示没有实际地震数据，我们通过使用**真实的模型model**的正演模拟，生成“**观测**”数据。从初始模型model0生成合成数据。得到的数据及差异，如图2。

最终，通过增加残差源表达式到之前定义的格子以及设置标识 `time_axis=Backward`，创建了完全的传播算子(propagator)，来定义传播算子在时间上向后运行：

```
op_adj = Operator([stencil_v] + res_term, time_axis=Backward)
```

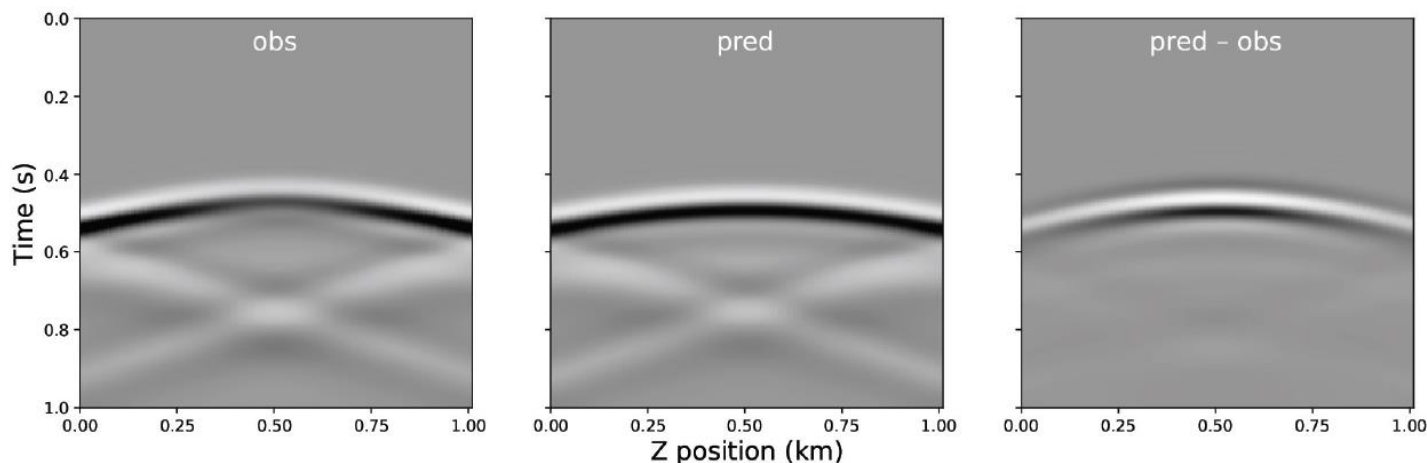


Figure 2. Shot records for a shot at a Z position of 0.5 km. (a) The observed data, using the known model with the high-velocity disc, contains a perturbation not present in (b) the “predicted” data, using the initial estimate of the model, which contains no disc. (c) The residual.



(2) Full-waveform inversion - Part 2: adjoint modeling

计算FWI梯度

$$\underset{\mathbf{m}}{\text{minimize}} f(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} \left\| \mathbf{d}_i^{\text{pred}}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i^{\text{obs}} \right\|_2^2$$

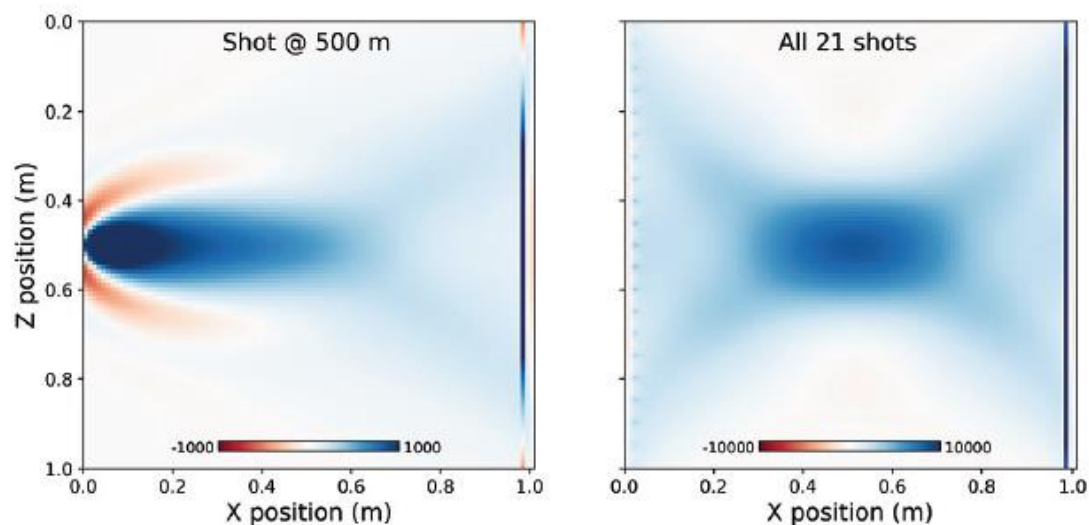


Figure 3. Gradient plots for (a) a single shot at 0.5 km and (b) the sum of all shots.



(3) Full-waveform inversion - Part 3: inversion

在正演波场模拟和使用伴随状态方法计算FWI目标函数梯度的基础上，下面构建反演框架，可用来最小化FWI最小二乘目标函数。

FWI在计算上和数学原理上都是有挑战的问题。计算复杂度源于需要重复求解计算量很大的波动方程，重复次数是优化算法各次迭代中源位置数目。数学原理复杂性源于FWI目标有多个局部最小值（由于循环跳跃）。

下面展示如何设置一个基本的FWI框架，可选择使用2种基于梯度的最优化算法：随机梯度下降和Gauss-Newton法。

使用**JUDI (JUDI.jl)**实施反演框架。



(3) Full-waveform inversion - Part 3: inversion

优化FWI目标函数

$$\underset{\mathbf{m}}{\text{minimize}} f(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} \left\| \mathbf{d}_i^{\text{pred}}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i^{\text{obs}} \right\|_2^2$$



(3) Full-waveform inversion - Part 3: inversion

在JUDI中实施FWI

读取数据集(SEG-Y), 由16个炮记录, 从SEG/EAGE Overthrust模型生成。存储在judiVector:

```
block = segy_read("overthrust_shot_records.segy")  
d_obs = judiVector(block);
```

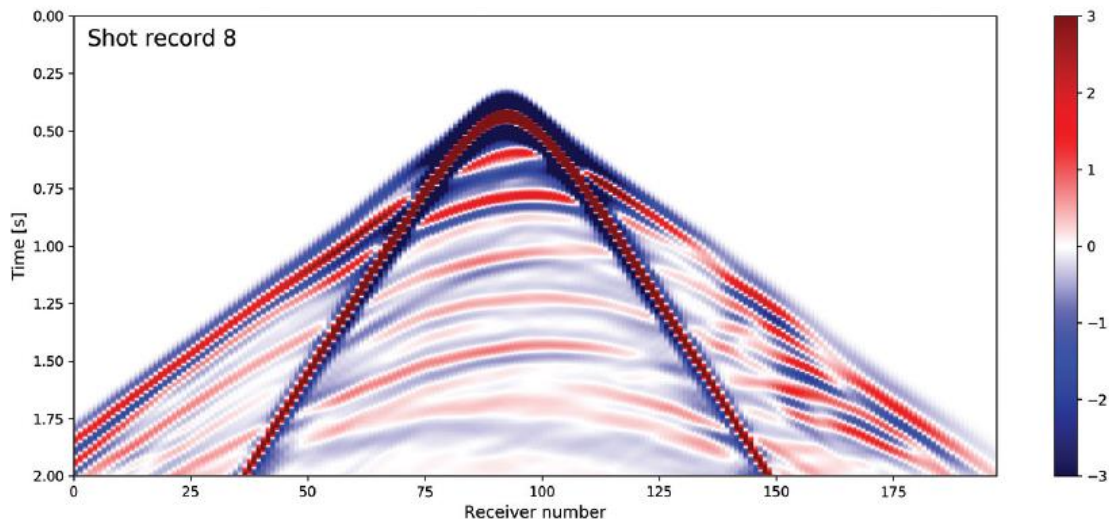
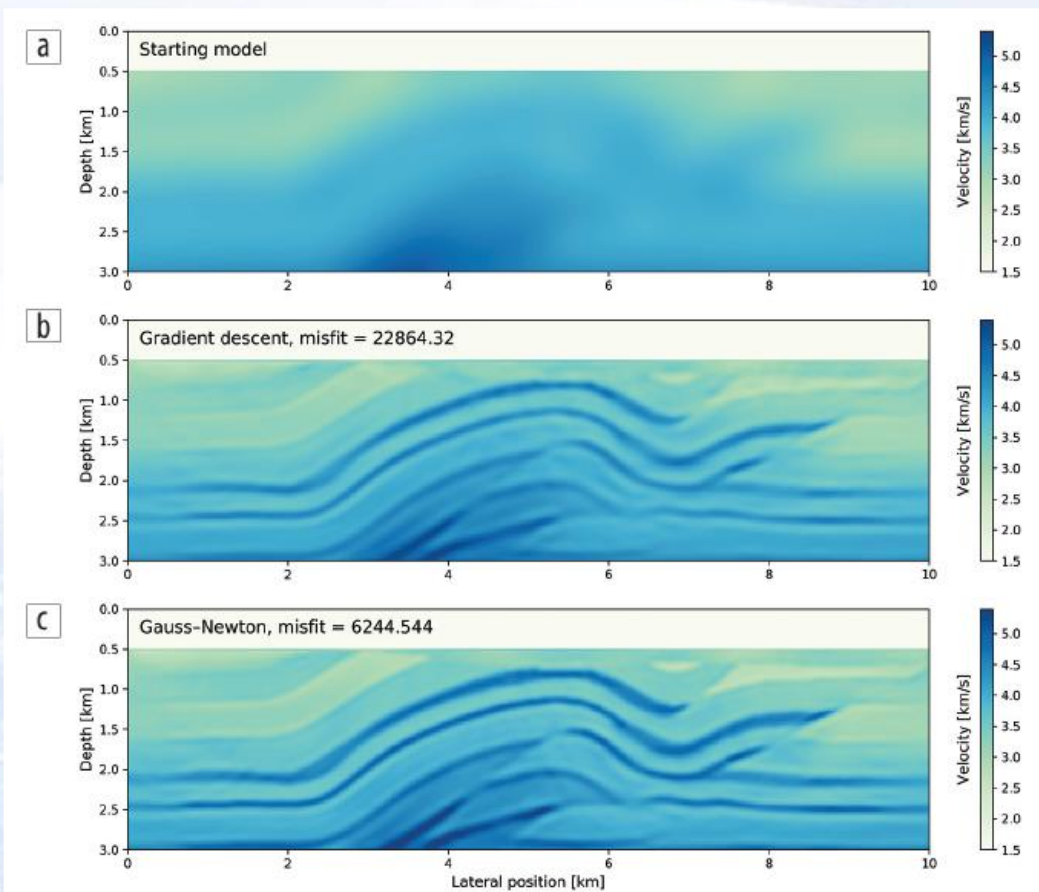


Figure 1. Observed shot record number 8.



通过梯度下降的FWI

使用盒子约束，10次迭代以后得结果，如图2





通过梯度下降的FWI

实际上，通常起始模型比示例的精度要低，通常从低频到高频执行FWI，因为对于较低频，目标函数有更低的局部最小值。在多尺度FWI方法中，首先使用low-pass-filtered的数据反演低分辨率的速度模型，然后添加更高频数据做后面的迭代优化。



中國地質大學
China University of Geosciences

艰苦朴素 求真务实

温家宝

通过高斯-牛顿的FWI

艰苦朴素
求真务实
温家宝

中国地质大学



逆时偏移(Reverse Time Migration)—基于Devito

RTM，能以较高的精度对所有地层倾角成像。

现在的逆时偏移是基于精确的波动方程，而不是近似处理。本质上是一种深度偏移方法。这些方法已被推广到三维叠前偏移，且它的效率和精度都有着相当大的提高。

RTM只产生具有运动学特征的图像（即构造成像），不一定有准确的振幅。

缺点：计算量大、存储数据量大、有稳定性和数值弥散问题等。



基于Devito库，快速实施地震数据处理的计算。

01_modelling.ipynb

类型: IPYNB 文件

02_rtm.ipynb

类型: IPYNB 文件

03_fwi.ipynb

类型: IPYNB 文件

04_dask.ipynb

类型: IPYNB 文件

04_dask_pickling.ipynb

类型: IPYNB 文件

05_staggered_acoustic.ipynb

类型: IPYNB 文件

06_elastic.ipynb

类型: IPYNB 文件

06_elastic_varying_parameters.ipynb

类型: IPYNB 文件

07_DRP_schemes.ipynb

类型: IPYNB 文件

08_snapshotting.ipynb

类型: IPYNB 文件

09_viscoelastic.ipynb

类型: IPYNB 文件

10_nmo_correction.ipynb

类型: IPYNB 文件

11_viscoacoustic.ipynb

类型: IPYNB 文件

12_time_blocking.ipynb

类型: IPYNB 文件

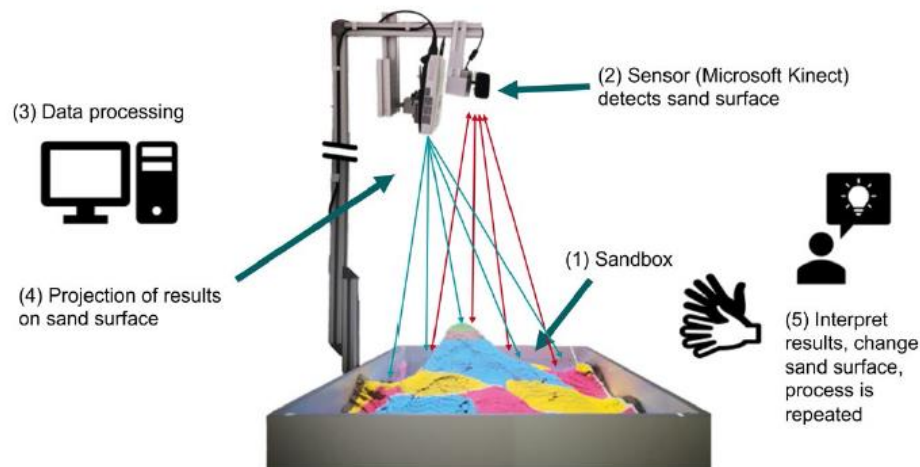
13_LSRTM_acoustic.ipynb

类型: IPYNB 文件

14_creating_synthetics.ipynb



Schematic overview of a typical AR-sandbox setup

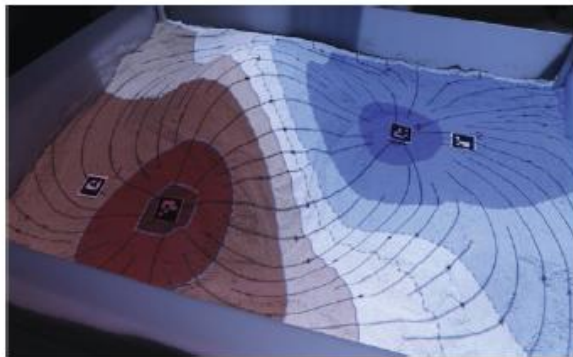


基于增强现实（VR）技术（Kinect2
体感相机+投影仪+沙盘），可实现：
地震海啸
地震波传播
地质模型展示
基本数学概念具体展示
滑坡过程

○ ○ ○

科普、本科教学、科学研究

(A) Geoelectrical field



(B) Seismic waves

