

Due Date: 17:00, Sunday, October 23, 2011
TA: Yunyue (Elita) Li (yunyue.li@sep.stanford.edu)

Lab 3 - Model fitting by least squares

*Enlai Zhou*¹

ABSTRACT

In this computer exercise you are given a module which performs a simple velocity transform. First you will improve the linear operator by implementing linear interpolation instead of nearest-neighbor interpolation. In the second part of the assignment, you will apply an iterative least-square optimization for velocity transform inversion.

INTRODUCTION

Chapter 4 of Basic Earth Imaging ² introduces the concept of velocity spectra. Performing NMO and stack with different stacking velocities transforms a CMP gather into velocity space. Primary and multiple reflection events usually have different apparent velocities. In this sense, they can be separated from the primaries in the velocity space. If we could forward model the CMP gather from its velocity-space representation, it would allow us to model multiples only and to subtract them from the original gather.

Figure 1, borrowed (with some modifications) from BEI, illustrates this idea. An idealized velocity spectrum (the top left plot) consists of four high-velocity spikes and a low-velocity spike. In this figure, the horizontal axis in the velocity transform domain is slowness, and the vertical axis is time. The top right plot shows a model CMP gather, derived from that model. The bottom left figure is the result of a velocity transform, and the bottom right figure is a CMP gather derived from it.

Both the forward modeling and the velocity transform were performed with module `velsimp`. To better understand this program, recall the following pseudo-code

¹**e-mail:** enlai.zhou@prc.cn

²<http://sepwww.stanford.edu/sep/prof/bei/toc.html/>

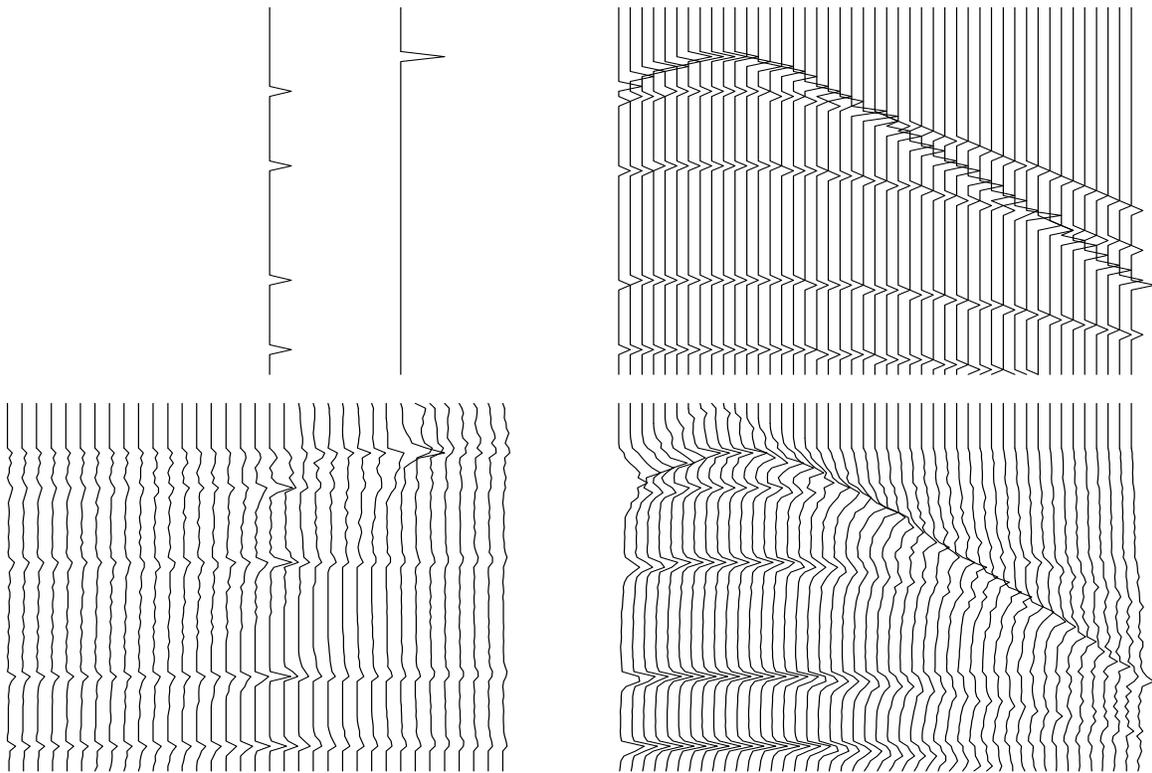


Figure 1: Iteration between spaces. Left are model spaces. Right are data spaces. Right derived from left. Lower model space derived from upper data space.

```

do v {
do τ {
do x {
  t = √(τ² + x²/v²)
  if hyperbola superposition
    data(t, x) = data(t, x) + vspace(τ, v)
  else if velocity analysis
    vspace(τ, v) = vspace(τ, v) + data(t, x)
}}}
```

We can see from Figure 1 that the adjoint process does take us back to the velocity space, but the result is far from perfect. The underlying velocity-space model is not reconstructed, and we are unable to predict the CMP gather from it. In this assignment, you will improve this situation first by enhancing the velocity transform operator, and then by applying an iterative least-square optimization.

YOUR ASSIGNMENT

Visit the GEE class webpage (<http://sep/doku.php?id=sep:courses:gp211>), find and download the source code for this lab, `Lab3.tar`. If you have not already, log into `glad` using your class account. Save this file in your home directory, then type `tar -xf Lab3.tar` to create your `Lab3` directory.

In your `Lab3` directory you will find three main programs (`Veltran.f90`, `Velmod.f90`, and `DPtest.f90`), and two subroutines (`velsimp.lop` and `halfdif.lop`). You will also find a Makefile, a SConstruct file (`SConstruct`) and a L^AT_EX file (`paper.tex`)

Begin the exercise by editing the `paper.tex` file to put your name on the paper and to answer the questions below. If you aren't familiar with `emacs` or `vi`, there is an intuitive editor called `gedit` which you can use by typing `gedit paper.tex`.

To see the results of your editing, type `SCons`. Hopefully the document will rebuild again with your changes. View it with your favourite pdf viewer. If you really mess things up and the document does not rebuild, you can always restore the original document by removing your `Lab3` directory and then type `tar -xf Lab3.tar`.

Answer the following questions.

- Begin by typing `make velvel.view`. A figure with four panels should appear. The upper-left-hand panel is the “known” slowness, spikes corresponding to five hyperbolic events, four of which have the same slowness. For the remainder of the assignment, we will refer to the velocity transform operator as **H**. In other words,

$$\mathbf{Hm} = \mathbf{d}$$

creates (t, x) data, \mathbf{d} , by applying the operator \mathbf{H} to a model, \mathbf{m} , in the space of zero-offset traveltimes and slowness (τ, s) . \mathbf{H} “pushes”, or “spreads” each point in model space (τ, s) to the corresponding hyperbola (apex at τ) in data space (t, x) . Conversely, the adjoint operation,

$$\mathbf{H}^T \mathbf{d} = \mathbf{m},$$

maps data, hopefully consisting of a series of events with hyperbolic moveout, to model space, which hopefully consists of a series of well resolved points, each corresponding to a single event in data space.

In short, we hope the operator \mathbf{H} is unitary, or orthogonal:

$$\mathbf{H}^T \mathbf{H} = \mathbf{I},$$

but as you will see, it is not. This fact is the basis for the lab.

- Type `make velvel.view` again. First look at the upper-right-hand plot, or the data modeled from the initial model vector:

$$\mathbf{d}_{\text{mod}} = \mathbf{H}\mathbf{m}_0.$$

Does the data look as you would expect? If not, explain the source of the trouble.

YOUR ANSWER.

- Now look at the lower-left-hand plot, or the velocity scan derived by applying the adjoint of \mathbf{H} to the modeled data:

$$\mathbf{m} = \mathbf{H}^T \mathbf{d}_{\text{mod}} = \mathbf{H}^T \mathbf{H}\mathbf{m}_0.$$

Compare this panel to the starting velocity model. Obviously, there are considerable differences. Notice that the original spikes in velocity turn into laterally-smearred spikes. What causes this smearing?

YOUR ANSWER.

- Focus still on the lower-left-hand plot. Can you think of any other artifacts that might appear? Do you see them in the lower-left-hand panel? **Hint:** Type `Grey < mod2.H | Tube` to view this panel in grayscale.

YOUR ANSWER.

- Last, focus on the lower-right-hand plot, which is the (t, x) data modeled from the ugly velocity panel in the lower left. These data look quite different from the “theoretical” data, or the data in the upper-right-hand corner, which was modeled directly from spikes in velocity space. What do you think are the two most important differences between this data and the theoretical data? Explain the cause of both differences.

YOUR ANSWER.

- The modeled CMP gather in the top right plot of Figure 1 appears to have a step-like moveout. This is clearly an artifact of the nearest-neighbor interpolation, used in module `velsimp`. Put simply, the operator loops over slowness, offset, and zero-offset traveltimes. Nonzero-offset traveltimes are computed via the hyperbola equation:

$$t = \sqrt{\tau^2 + (xs)^2}.$$

Rarely does this value fall on a traveltimes “bin”.

Your next task is to modify the `velsimp` module so that linear interpolation between two adjacent samples in time is used instead of nearest-neighbor interpolation. When you are done, Figure 2 will reflect your changes.

Hints: Notice that the `velsimp_init` interface has a logical argument, `lint`. If `lint = .false.`, nearest-neighbor interpolation is used. If `lint = .true.`, linear interpolation is used. For an example of the algorithm, review operator `lint1` from Chapter 1.

- Once you are finished, check to see if you have coded the adjoint correctly by typing `make dottest`. Modify program `DPtest.f90` so that `velsimp` is called using linear interpolation. You need not report your results, but if you have coded the adjoint incorrectly, your later results will have problems. **Hint:** To get an intuition for calling the `dot_test` function, look at the source code: `prof/gee/Lib/dottest.f90`.

YOUR ANSWER.

- Now type `make velvel-lint.view` to view your results. A two-frame movie will come onto the screen. The first frame shows your nearest-neighbor results, while the second frame shows your improved, linear interpolation results. Compare the CMP data modeled directly from the slowness spikes. Has linear interpolation improved the hyperbola modeling?

YOUR ANSWER.

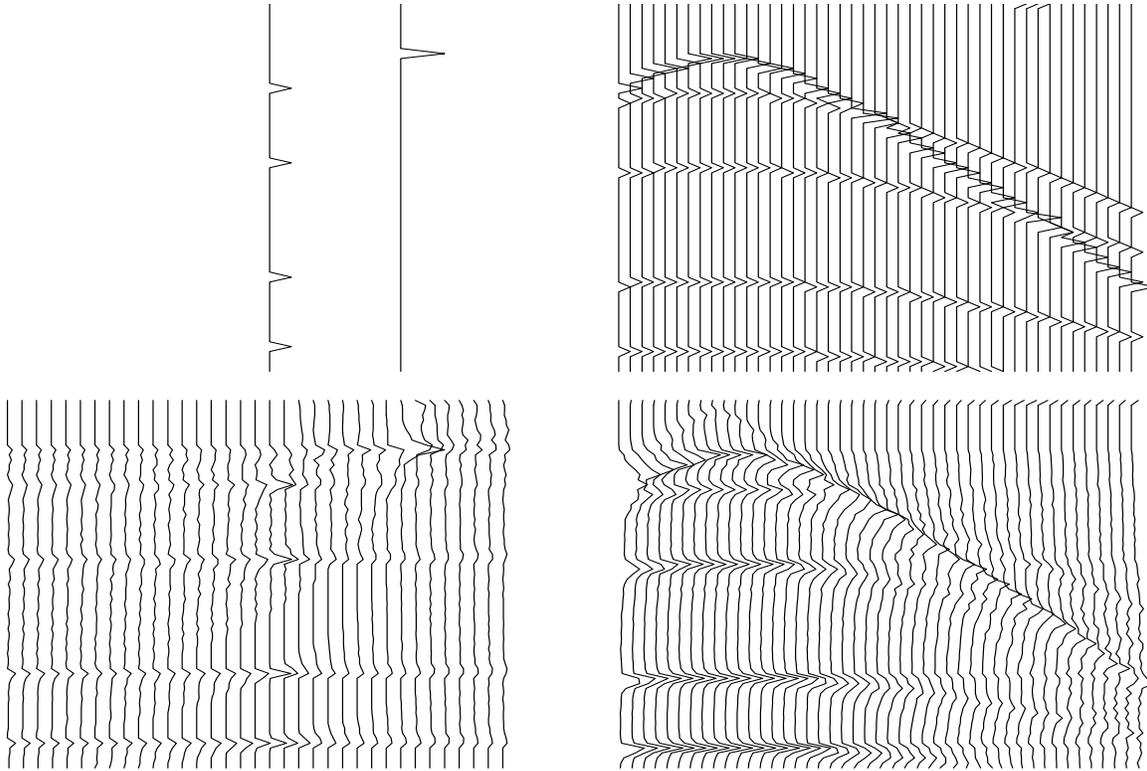


Figure 2: Same as Figure 1, only we use linear interpolation in the velocity transform operator. Iteration between spaces. Left are model spaces. Right are data spaces. Right derived from left. Lower model space derived from upper data space.

- At this point, you should be wondering why the addition of linear interpolation failed to make better-looking lower panels on Figure 2. Certainly, smoother hyperbolas were generated, but say you were an interpreter, picking velocities from real data. The upper-right-hand panel of Figure 2 is made up of perfect hyperbolas, yet the velocity scan is nearly useless for the real goal, which is picking velocities!

Least squares estimation (inversion) comes to the rescue. Rather than simply applying the adjoint and stopping, we will search for the model (velocity panel) which optimally explains the data (CMP gather), given a physical connection (\mathbf{H}) between the data and model.

- Modify program `Veltran` so that in addition to applying the adjoint operator, it is enabled to call subroutine `solver` to run an iterative least-square optimization by the conjugate-gradient method. You can find examples of using the `solver_smp` subroutine in Chapter 2 of the textbook.

- After you have modified the program, run `make velinv-lint.view` to recreate Figure 3. You will see a two frame movie. The first frame is simply Figure 2, while the second is the result using optimization.

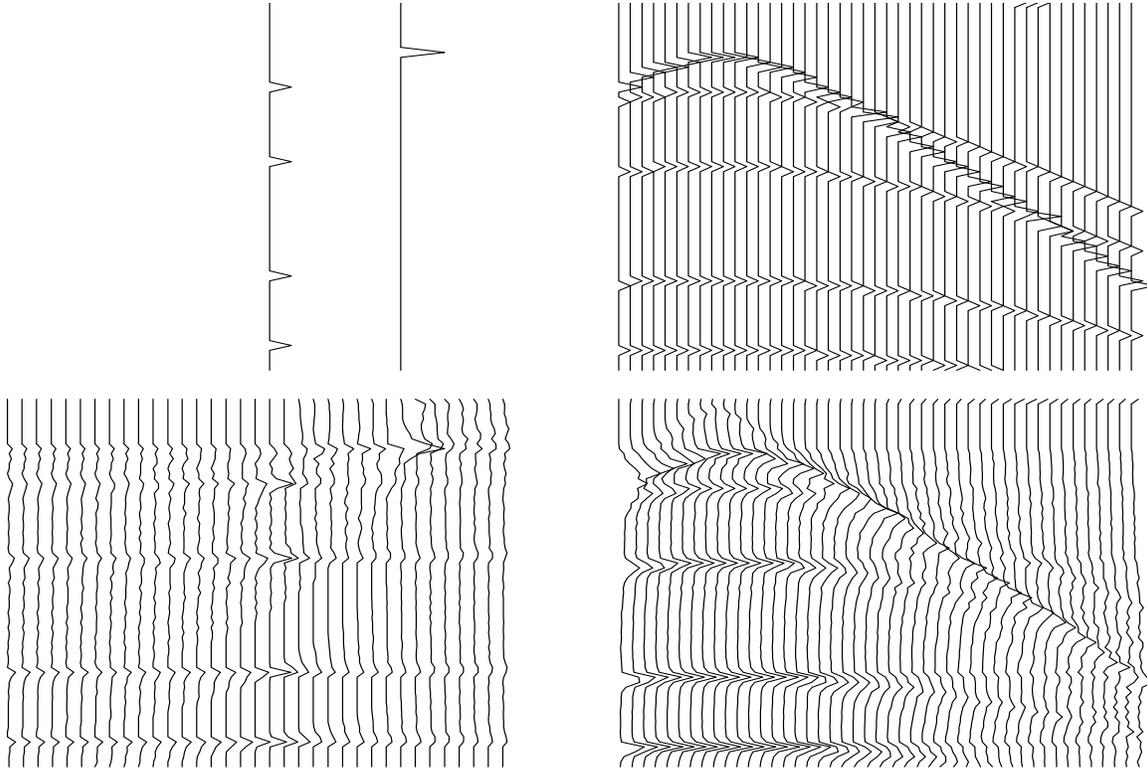


Figure 3: Iterative approach to solving problem, using linear interpolation along the time axis. Left are model spaces. Right are data spaces. Right derived from left. Lower model space derived from upper data space after 20 conjugate-gradient iterations.

- Has optimization improved the quality of the velocity scan (lower-left-hand panel)? Do you see model space artifacts in the inversion result? Explain differences with the adjoint processing result. Explain why artifacts are still or are no longer present.

YOUR ANSWER.

- In practice, the data modeled from the inversion result provides us with an excellent diagnostic tool. In short, if application of the modeling operator to the model we've just estimated doesn't give something that looks like the input data, something has gone wrong. Type `make diff.view` and examine the top row of the plot. From left to right, we have the input CMP data, the modeled

CMP data, and the difference. Describe the difference mathematically.

YOUR ANSWER.

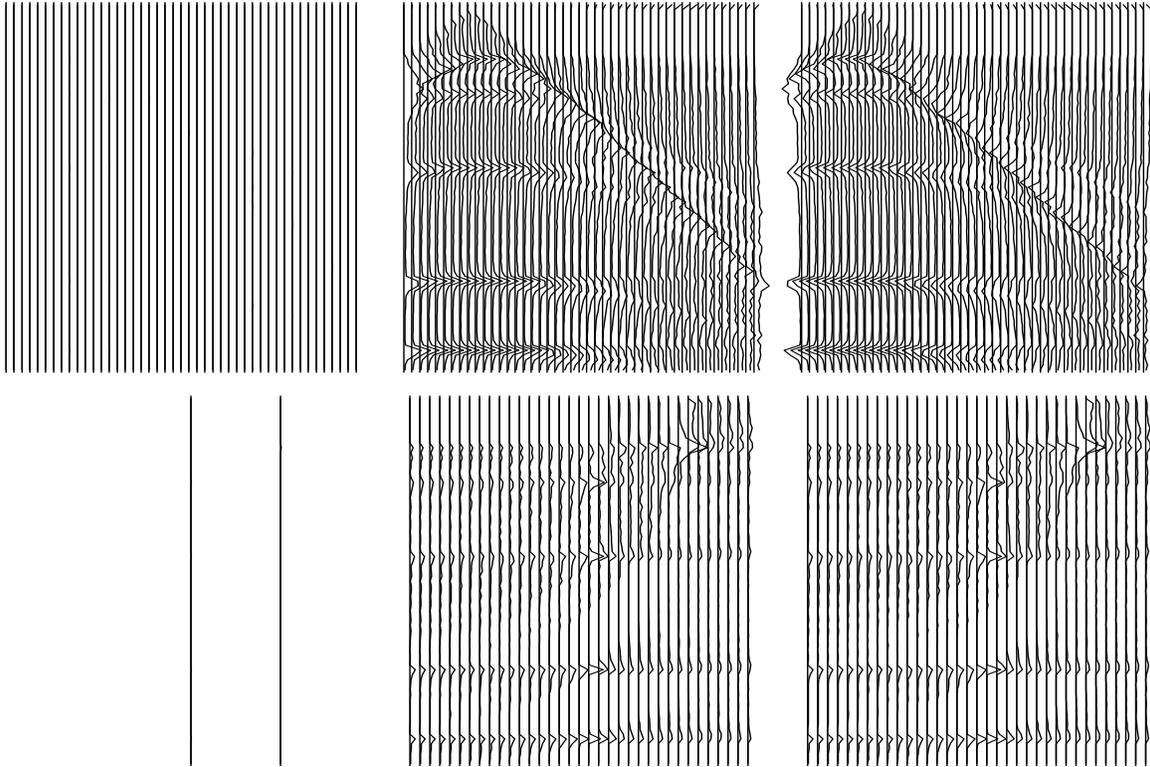


Figure 4: Top, from left to right: Input CMP gather; CMP gather modeled from estimated velocity; Difference. Bottom, from left to right: Input slowness model; Estimated slowness model; Difference.

- Type `make diff.view` and examine the bottom row of the plot. From left to right, we have “known” slowness model, estimated slowness model, and difference. From this information and your answer to the previous question, does the inverse problem appear to have a null space? If so, describe the components of the null space that you directly observe.

YOUR ANSWER.

- In the method of conjugate directions, is the residual power guaranteed to decrease at each iteration?

YOUR ANSWER.

- Is the gradient power guaranteed to decrease?

YOUR ANSWER.

- How would you define the rate of convergence?

YOUR ANSWER.

DONE

When you are all done, print your paper and hand in to your TA. Clean up your directory by typing `make clean`.

In SEP, we highly value the reproducibility of our research. Type `make clean`, `make burn`, `make` and `scons` at the end to make sure all your results in the paper are reproducible.