

Due Date: 17:00, Mon, February 23, 2015
TA: Yi Shen (yishen@sep.stanford.edu)

Lab 5 - Interval velocity estimation

*Your name*¹

ABSTRACT

This lab will guide you through most of the steps - and pitfalls - of interval velocity estimation, perhaps the most important problem in exploration geophysics. You will apply a least squares slowness scan, and then pick RMS slowness using an auto-picker. You will then solve the Dix equation as a least squares problem. Finally, you will test the effects of using regularization weighting on the Dix inversion.

INTRODUCTION

Exploration seismologists propagate waves through the Earth's surface and record the traveltimes. In oversimplified terms, the traveltimes are the path integrals of the interval slowness (1/velocity) over the raypath between source and receiver. "Inversion" of the traveltimes for the underlying velocity field is perhaps the most important problem in our branch of geophysics.

Under suitable assumptions, we can directly estimate root-mean-square (RMS) velocity from multioffset seismic data. You have seen the theory in *Basic Earth Imaging* and experimented with it in Lab 3 of this course. However, in order to justify our salaries, we geophysicists must go beyond RMS velocity, and estimate a field of interval velocities, which is needed for seismic depth conversion.

For a stratified ($v(z)$) earth, we can easily write the relationship between RMS velocity (V) and interval velocity (v):

$$V^2(\tau) = \frac{1}{\tau} \sum_i v_i^2 \Delta\tau_i \quad (1)$$

I say "easily" because the above relationship is implied by the words *root-mean-square*. Equation (1) has an analytic inverse, known as the Dix equation:

$$v_k^2 = \frac{\tau_k V_k^2 - \tau_{k-1} V_{k-1}^2}{\tau_k - \tau_{k-1}} \quad (2)$$

¹**e-mail:** youremail@stanford.edu

For a detailed derivation, please see *Basic Earth Imaging*. In practice, the Dix equation often fails to produce a pleasing result, particularly in regions of complex stratigraphy.

In this lab, you will use your expertise in optimization to cast the Dix equation as a least squares problem, in order to more robustly estimate an interval velocity function from a marine CMP gather.

YOUR ASSIGNMENT

In this lab, we will work with a CMP gather shown in Figure 1 which can be found in the directory `Dat`. There are three parts to this lab. First, you will compute a slowness scan using the NMO operator (similar to Lab 3). Second, you will pick the RMS slowness from these scans using a picking program and convert them to RMS velocity. Finally, you will do a Dix inversion of the RMS velocity to convert it to interval velocity. You are provided with modules for the stepper, solvers and dot product test as well as a picking program. The provided modules and program are complete and do not need to be modified. You will code several modules and programs to answer the questions and perform the required operations. Moreover, no parameters are provided for you for any step. Use your best judgment to estimate these parameters. Modify the `Makefile` to add rules that generate the required tests, results and figures for each question. Make sure to include the correct and complete set of prerequisites for each `Makefile` rule (including the executable programs). Also, do not forget to add the PDF figures to the default target and include the figures in this paper.

Questions

Slowness scan optimization

1. Use your NMO operator from Lab 3 and estimate the slowness scan by an adjoint only. Plot the scan and comment on its quality and artifacts (if they exist).

YOUR ANSWER.

2. With all the experience you have from pervious labs, you know that you can do better than an adjoint. Do a least-squares optimization to compute the slowness scan using data-fitting only. Plot your results. Are these results better? Why?

YOUR ANSWER.

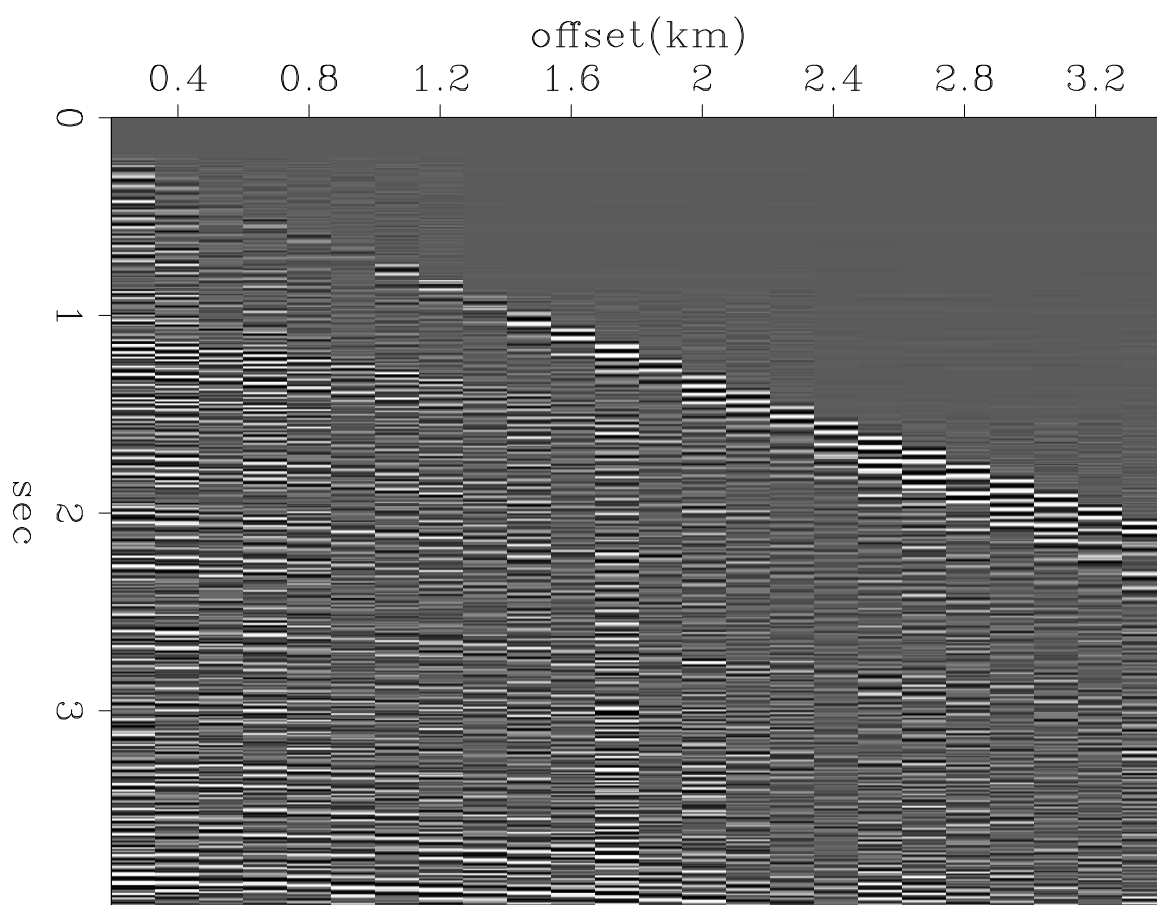


Figure 1: CMP gather.

3. If you looked at the previous results and thought that regularization is needed, you are right. Redo the previous question but now with a regularization as well. Feel free to try and experiment with the different regularizations (e.g. identity, gradient, Laplacian, etc.) and epsilons. Which regularization gave you the best result? Plot your results and comment on the effects of regularization.

YOUR ANSWER.

Slowness RMS picking

Now that you managed to produce a nice and clean slowness scan, we want to pick the RMS slowness from it. But before that, we want to process and simplify the scan to make the picking easier. One way to achieve this is by taking the envelope of the signal. The SEPLib program **Envelope** might come in handy.

Now that your scan is ready to go, use the program **Picker** to do the picking for you. This program picks the maximum of the function given an upper and lower bounds. The bounds are defined by:

$$s_{\text{bound}}(\tau) = s_0 + a\tau + b\sqrt{\tau}, \quad (3)$$

where the parameters s_0 , a and b need to be supplied for each bound. The program **Picker** will have four outputs: the two bounds, the RMS slowness picked and the values of the scan at those picked location.

4. Adjust the bounds parameters until you are happy with the picks. What are your parameters values? Convert the picks from slowness to velocity and plot your result.

YOUR ANSWER.

5. Write a program which applies the NMO correction to the CMP gather using picked RMS slowness. Run this program using the auto-picked RMS slowness and plot your results. Are all the events flat? How well did the picker do?

YOUR ANSWER.

Dix optimization

6. Before we setup a least-squares optimization, let's try using the Dix equations directly. Write a program that computes the interval velocity using the Dix

equation. Run this program on the auto-picked RMS velocity. Plot your results and comment on their accuracy and validity.

YOUR ANSWER.

7. You decide now that using the Dix equation (i.e. using the inverse) might not be a good idea for auto-picked RMS velocity. Instead, you want to setup the interval velocity estimation as a least squares optimization (i.e. using forward and adjoint operators) from equation (1). What is the model space, data space and operator? (Hint: we need the problem to be linear).

YOUR ANSWER.

8. Write a module for the operator you describe in the previous question. Then, write a program that performs the dot product test on this operator. What are your results?

YOUR ANSWER.

9. Your experience tells you that we should not use data-fitting optimization only since you already know that the inverse did not work. Therefore, you decide to jump to a regularized optimization right away. Write a program that implements the regularized solver with a regularization operator of your choice. Run the optimization for some iterations and plot the estimated interval velocity (not the model). How well did your optimization do? Why did you get these results?

YOUR ANSWER.

10. You remember now that your auto-picker also had the values of the scan as an output and decide to use these values to weight the data-fitting equation. Write a module for the weighting operator and remember to pass the weights in the initialization. Then, write a program that implements the regularized solver and uses the picks' values for the weighting operator W_{op} . Run the optimization for some iterations and plot the estimated interval velocity. Did these weights improve the optimization?

YOUR ANSWER.

Extra Credit

1. Does picking the maximum of velocity scan really give us the RMS velocity? If not, what is the difference between the RMS velocity and what is being measured?

YOUR ANSWER.

2. Why is using the Dix equation not stable?

YOUR ANSWER.

3. If you were doing the interval velocity estimation for a whole survey with several CMPs. Would you constrain the Dix optimization any differently? Why?

YOUR ANSWER.

DONE

When you are all finished modifying the source files, **Makefile** and latex file, make sure that will paper will compile correctly from a cleaned directory using the default targets only. In other words, the following sequence of commands should produce your a PDF version of your paper without any problems: **make burn; scons -c; make; scons.**

Once you make sure everything is working properly, clean up your directory by typing **make burn; scons -c.** Then, compress the lab directory using the command **tar cvzf Lab5.zip Lab5** and submit the compressed file to your TA.