Geol 335.3

Lab #5: Reflection Seismic Synthetics

In this lab, you will use Matlab or Octave programs to visualize and study reflections from relatively complexly shaped reflecting boundaries. This program will also be used to produce a “zero-offset (common-midpoint) section” section in the next lab.

Theory

To model reflection responses from an arbitrary boundary, we can represent the boundary by a sequence of closely spaced reflection (or “scattering”) points. In this way, there is really no difference between linear segments and points at which the reflecting boundary is interrupted or changes direction. Reflections from continuous linear segments add up coherently and form the usual reflection hyperbolas. Reflections from isolated scattering points, kinks, and edges of reflectors provide diffractions.

In the program, we will represent the elements of the model as follows:

To represent:	*Use:*
Coordinates and vectors: (x,z)	2-element arrays `[x,y]`
Reflectors	2-column matrices, in which each row represents a point on the reflector, and the points are assumed to be connected by straight lines
Time series	1-dimensional arrays
Output data sections	2-dimensional array. Columns of this array are the time series of reflection amplitudes, and columns correspond to different receivers.

Accordingly, the calculations are performed in Matlab with the use of vector and array operations.

Code

Most of the code is provided in the the zipped archive. Create a directory for this lab and unpack the archive into it. Look into file `model1.m` and understand the general structure of the code. This structure is similar to the organization of data processing in most software packages. The code is subdivided into a set of functions with simple and well-defined functionalities. These functionalities are described in commentaries at the beginning of each m-file. These commentaries can be viewed by using ‘help’ command in Matlab, for example by typing:

```
help picklayer
```

Look into the headers and bodies of each of these functions and understand the meanings of their parameters, return values, and the ideas of algorithms. In your own code, make sure to document processing by similarly providing commentaries describing the parameters, assumptions, and return values.

Common parameters such as layer parameters and the source and receiver geometries are provided to all functions in the common workspace using the `global` declarations in Matlab:

```
global    receivers % receiver locations

global    sources   % source locations

global    time      % time sampling

global    xref wref % source waveform
```

Here, array `receivers` should contain a list of receiver positions, and array `time` – list of times at which the resulting records are sampled. Variable `velocity1` is the overburden velocity. These variables are set in `model1.m` at the beginning of processing. You can use the suggested values or modify them as you like.

To perform the assignments below, you will need to modify the main code (`model1.m` or additional versions which you create) and pick the desired model boundaries interactively.

Seismic reflection records are generated by using calls to function `reflection`(…), for example (see `model1.m`):

```
section_2_1 =
       reflection(sources(2),receivers,layer1,velocity1);
```

which are later added to the output plots using function `plot_section(…)`.

Give some thought to naming the variables for plot sections, so that they can be conveniently identified and used later. For example, the above notation `section_2_1` means “seismic records for waves from source number 2 reflected from boundary (layer) number 1”.

Assignments

1. (10%) **Execute** code `model1.m` . The code should give you an interactive Figure 1. This figure pops up when the following function is executed:

```
layer1 = picklayer(line_length,depth);
```

Use computer mouse to pick three or four points making a reflecting boundary. Try a boundary at a relatively shallow depth. Press the space bar or click outside of the plot region to stop picking.

After the picking is finished, the code may take some time for computations and then display in Figure 2 the depth section together with reflections from the three sources.

Note that at the end of this script, it saves all results in a .mat file:

```
save model1.mat layer* section*
```

During the next steps of modeling, you can use and update the same .mat file.

2. (25%) Make copies of `model1.m` named `model2.m` and `model3.m` and modify them so that they load the results of the preceding steps and add another reflector. To do this:

    - Uncomment statement '`load model1.mat`' at the beginning of `model2.m.` This should load all results of the previous step.

    - Change calls to `picklayer` like this:

```
layer2 = picklayer(line_length,depth,layer1);

layer3 = picklayer(line_length,depth,layer1,layer2);
```

Note that the last arguments in `picklayer()` here are used to display the previously picked layers, so that you can construct some meaningful model.

- Similarly, change '`layer1`' to '`layer2`' or '`layer3`' in calls to `reflection(…)` and `plot_section(…).`

- Evaluate whether plotting is successful. For example, if a cross-section subplot is empty or not showing all the expected reflecting boundaries, check whether the corresponding `plot(layer…)` commands are present and correct.

- When plotting individual scattering points (particularly in questions 5 and 6 below), the plotting style should be changed to using symbols. For example, parameter '`g+`' in `plot(…)` means plotting scattering points using plus symbols.

3. (25%) **Explore the resulting models**, possibly by changing the picking done in `model1.m`. Try creating deeper reflectors in `layer2` and `layer3`, structures with steeper dips or sharper corners, or constructing a wedge.

    Create plots (pdf, jpeg, or hardcopy) of Figure 2 for the best cases that you wish to report. <u>If using multiple layers</u>, it is convenient to <u>show records from only one source</u> in one plot.

4. (15%) **Briefly describe your observations**:

   a. What are the shapes of reflections like (straight lines, hyperbolas)?

   b. For a given source, at what coordinate *X* does the smallest reflection time occur? Directly below the source or shifted up-dip or down-dip?

   c. How are the times of reflections related to the depths of reflectors?

   d. For a dipping reflector, how do the **shapes of reflections differ for** sources located up- and down dip?

5. (15%) **Repeat step 1 by picking only a single point** ("point scatterer") within the subsurface, near the middle of the model. Create plots and describe the resulting wavefield. Does the onset of the reflection look like a hyperbola?

   a. From the Figure 2 plot, **measure (manually) the (*x*,*t*) slopes** of the far-away branches of the reflection hyperbola. How are these slopes related to the wave velocity within the medium?

6. (10%) **Repeat** step 5 by picking the scatterer at a very shallow depth. **How do the reflections from the shallow scatterer look like**?

Hand in:

Zipped directories or Word, PDF files containing:

1. Your Matlab codes ("m-files");
2. Screen captures or Postscript/PDF figures;
3. Summary and discussion of observations.