Geol 335.3

Lab #4: Seismic Modelling

We will study the response of the direct, reflected and refracted waves to the following Earth model:

Layer	Thickness (m)	Velocity (m/s)
1	100	450
2	200	2000
3	200	3000
4	∞	6000

A seismic survey is conducted with the following specifications: Maximum source-receiver separation is 1000 m and the spacing between receivers is 20 m.

(1) (10%) Determine all: a) critical ray parameters; b) critical distances, c) intercept times for the refractions, d) t_0 times and RMS velocities for the reflections. In Matlab, assign identifiers to these quantities, such as:

```
p_crit_1 = …;     % critical ray parameter
x_crit_1 = …;     % critical distance
t_int_1 = …'      % intercept time
t_int_2 = …;
t0_1 = …;         % normal-incidence reflection time;
v_rms_1 = …;
```

and so on <u>In the following, *use these identifiers* rather than constants</u>.

<u>Note:</u> the units for distance should be m, time – ms, velocity – accordingly, m/ms (or km/s).

(2) (25%) Using Matlab, compute the arrival times of the direct, reflected and refracted waves from all the acoustic discontinuities for all receivers. Plot these arrival times on a common T-X graph. It would be helpful to write two Matlab *functions* to calculate reflection and refraction travel times, respectively:

`trefl(t0,v_rms)` and `trefr(t_intercept,v_refractor)`

In the subsequent steps, you could simply use those functions, instead of retyping the formulas.

Next, look into the function `wavelet_ft(t)` that returns a wavelet (array) of amplitude `1.0`, centred at time `t=0` and discretized at times specified in array `'t'` (in milliseconds). The wavelet is obtained by summation of 81 cosine waves of amplitude `1.0/81.0`, with frequencies from 20 Hz to 60 Hz with a 0.5 Hz interval. Finally, the waveform is multiplied by a sine "taper" function to ensure that its values at the ends of the time interval equal zero.

This is your *source signature* (aligned at 0 ms time). Such symmetric with respect to time = 0 waveforms are called "zero-phase".

(3) (10%) Test this function by executing something like this:

```
global tref wref

tref = -500:1:500;
wref = wavelet_ft(tref);
plot(tref,wref,'b-')
save wavelet.dat tref wref
```

These commands create an array of times `tref` centred at `t=0`, model wavelet `wref` on it, plot `wref`, and save it in a file. This precomputed wavelet can be loaded like this:

```
global tref wref
load wavelet.dat
```

and will be used later.

Note that we use `'.dat'` file name extensions instead of the conventional `'.mat'` because Windows displays `'*.mat'` as system files.

(4) (10%) Look into the second Matlab function `wavelet(a0,t0,t)`. This function uses the precomputed `tref` and `wref` and returns the same wavelet but with arbitrary amplitude `'a0'` and centred at arbitrary time `'t0'` [ms]. The output is obtained by simple linear interpolation (function `interp1(…)`) and is much faster than the summations above.

Test this function like this (you can change parameter values or make multiple tests):

```
t = 0:2:1000;
plot(t, wavelet(5.0,100.0, t ) );
```

Print the resulting plot out and indicate whether the time and amplitude of the waveform are as expected.

(5) (40%) Using the calculated arrival travel-times and `wavelet(…)` function, use Matlab to plot a *shot gather* – a set of seismic traces simulated at each of the 50 geophone locations. Use 0-1200 ms time range, at 2 ms intervals. Note that for each interface, you will have two arrivals (direct wave and reflection) at pre-critical offsets and three arrivals (direct, reflected, and head wave) beyond that range.

For plotting, you will have to use Matlab's `hold` command (to prevent from starting a display window on every `plot()` call) and arrange a loop over the receivers. For *i*-th receiver (beyond the critical distance), you will first for the trace and then plot it:

```
% determine the source-receiver distance of this trace
x = i*20.0;

% put direct wave into new trace
trace = wavelet(amp, x/v1, t );

% add reflection from the first interface
trace = trace + wavelet(amp, trefl(t0_1,v_rms1), t );

% add refraction from the first interface

if x > x_crit_1 then
  trace = trace + wavelet(amp, trefr(t_int_1,v2), t );
end

% ...and similarly for the other two interfaces

plot(trace+x,t); % plot the resulting trace
```

Here, `t_direct`, `t_head`, and `t_reflected` should be the times of your arrivals. The horizontal axis of the plot will become the distance, and the shifts of `x` added to the '`trace`' [m] will place your wiggles ("traces") at the positions of geophones. Parameter `amp` becomes trace excursion (swing amplitude) and should be chosen (I suggest ~15-20) so that the plot looks nice. Time will be plotted along the vertical axis, as it is commonly done in refraction/reflection record sections.

(6) (5%) Notice the interference between the different waves near critical and crossover distances, and far offsets. Mark these interfering events in your section. Comment on the tuning effects (enhancement or cancellation of arrival amplitudes as they approach each other within a dominant period.

(7) (**Bonus 10%**) Speculate on the nature of the artefact in the lower-right corner of your plot (hint: look at the `wavelet` function). Try and remove it.

Theory:

The cosine function of amplitude *A*, frequency *f*, and centered at time t_0, is:

$$u(t) = A\cos(2\pi f(t - t_0))$$

For a stack of layers, travel time of the reflection from the bottom of *n*-th of them is given by the same formula as within a single layer:

$$t(x) = \sqrt{t_0^2 + \left(\frac{x}{V_{RMS}}\right)^2} \ ,$$

where t_0 is the vertical-incidence travel time, and V_{RMS} is the effective (“root-mean-square”) velocity of the stack of layers:

$$V_{RMS}^2 = \frac{\sum_{i=1}^{n} V_i^2 \Delta t_i}{\sum_{i=1}^{n} \Delta t_i} \ ,$$

where $\Delta t_i = \Delta z_i / V_i$ is the vertical travel time for each layer.

Hand in:

Code printouts, plots, and comments in a compressed directory.