Notes for the Ray Codes

A C Birch

May 22, 2002

Contents

1 Introduction

The purpose of this document is to provide an outline for some of my ray tracing codes as well a brief review of the underlying equations. In section 2 I give, with minimal description, the basic equations describing rays. These equations are all standard in the literature. In section 3 I describe the numerical approach to integrating over the singularities that occur at the ray turning points. Section 4 describes the main codes in this package. Finally, section 5 describes some simple tests of these codes that I have done. I would to like to emphasize, however, that these codes are far from 100% reliable and could use a lot more testing, in particular with regard to the numerical approximations that have been employed.

2 Equations

One approach to computing rays is described by D'Silva & Duvall (1995), the equations in this section are for the most part taken from that paper. With the neglect of the buoyancy frequency the group velocity along a ray is

$$v_r = \frac{\partial \omega}{\partial k_r} = \frac{k_r c^2}{\omega}\,, \tag{1}$$

$$v_\theta = \frac{\partial \omega}{\partial k_\theta} = \frac{k_\theta c^2}{\omega}\,. \tag{2}$$

In the above equations k_r is the radial component of the wavevector, k_θ the horizontal component of the wave vector, c the sound speed, and $\omega(k)$ gives the dispersion relation. The dispersion relation is given by

$$k_r^2 + k_\theta^2 = \frac{\omega^2 - \omega_{\rm ac}^2}{c^2} \tag{3}$$

where $\omega_{\rm ac}$ is the acoustic cutoff frequency. The horizontal wavenumber is related to the angular degree by

$$k_\theta^2 = \frac{l(l+1)}{r^2}\,. \tag{4}$$

D'Silva & Duvall (1995) show that the first-skip distance and group time are

$$t = 2\int_{r_{\rm L}}^{r_{\rm U}} \frac{{\rm d}r}{v_r}\,, \tag{5}$$

$$\Delta = 2\int_{r_{\rm L}}^{r_{\rm U}} \frac{v_\theta}{v_r}\frac{{\rm d}r}{r}\,, \tag{6}$$

where $r_{\rm L}$ is the lower turning point and $r_{\rm U}$ the upper turning point. Using equations (1) and (2) to replace v_r and v_θ in the above equations we obtain

$$t = 2\omega\int_{r_{\rm L}}^{r_{\rm U}} \frac{{\rm d}r}{k_r c^2}\,, \tag{7}$$

$$\Delta({\rm deg}) = \frac{360}{\pi}\int_{r_{\rm L}}^{r_{\rm U}} \frac{k_\theta}{k_r}\frac{{\rm d}r}{r}\,. \tag{8}$$

If we want to compute the ray path, not merely first-skip group time and distance, we can use, again neglecting the buoyancy frequency,

$$\frac{1}{r}\frac{{\rm d}r}{{\rm d}\theta} = \frac{v_r}{v_h} = \frac{k_r}{k_h}\,. \tag{9}$$

The change in heliocentric angle along the ray is thus given by

$$\theta(r) = \int_{r_{\rm L}}^{r} \frac{k_h}{k_r}\frac{{\rm d}r}{r}\,. \tag{10}$$

(Giles, 1999).

3 Numerics

We use the same basic approach as Christensen-Dalsgaard et al. (1989) to deal with integration over the singularities. Consider the calculation of equation (7). Away from the turning points the integrand is finite and the integration scheme is not important. The integrand is singular, however, at the turning points, where k_r goes to zero. We need to figure out how to integrate over a small region around a turning point. Say the lower turning point is at $r_{\rm L}$ and that we have done the integral up to some point r_1 very near $r_{\rm L}$. Very near the turning point $k_r^2 c^4$ is nearly linear and can be written as

$$k_r^2 c^4 \approx ar + b\,. \tag{11}$$

The turning point is at $r_{\rm L} = -b/a$. To do the integral from $r_{\rm L}$ to r_1 use this linear approximation for $k_r^2 c^4$ to write

$$\delta t = \int_{r_{\rm L}}^{r_1} \frac{\mathrm{d}r}{k_r c^2} \approx \int_{r_{\rm L}}^{r_1} \frac{\mathrm{d}r}{\sqrt{ar+b}} = \frac{2}{a}\sqrt{ar_1 + b} \tag{12}$$

The upper turning point can be done in the same fashion. The computation of Δ and θ can be done in much the same manner.

For the remainder of this section we use r_1 and r_2 to denote the limits of where we are using a conventional integration scheme, $r_{\rm L} < r_1 < r_2 < r_{\rm U}$. In the following equations a and b are obtained from linear fits to $k_r^2 c^4$ near the appropriate turning point:

$$t_{\rm L} = \frac{4\omega}{a}\sqrt{ar_1 + b}\,, \tag{13}$$

$$t_{\rm U} = \frac{-4\omega}{a}\sqrt{ar_2 + b}\,, \tag{14}$$

$$t = t_{\rm U} + t_{\rm L} + 2\omega \int_{r_1}^{r_2} \frac{dr}{k_r c^2}\,. \tag{15}$$

A similar set of equations can be derived for Δ by using a linear approximation for $k_r^2 r^2 / k_h^2$ near the turning points. The result is

$$\Delta_{\rm L} = \frac{720}{\pi}\sqrt{ar_1 + b}\,, \tag{16}$$

$$\Delta_{\rm U} = \frac{-720}{\pi}\sqrt{ar_2 + b}\,, \tag{17}$$

$$\Delta = \Delta_{\rm L} + \Delta_{\rm U} + \frac{360}{\pi}\int_{r_1}^{r_2} \frac{k_h}{k_r}\frac{\mathrm{d}r}{r}\,. \tag{18}$$

The equations for $\theta(r)$ are very similar.

4 Codes

There are three main codes, `run_tau`, `run_raypath`, and `ray_kernel`. `run_tau` is the code for computing group time as a function of distance. `run_raypath` is for computing ray paths, as well as the phase and group times along the ray path. Finally, `ray_kernel` computes ray kernels for the square of sound speed. The basic operation of these three codes is described in the following subsections.

4.1 `run_tau.m`

This script sets up the inputs for `tau` and calls that function. The user sets the value of frequency and angular degree to compute for by editing the first fews lines in `run_tau.m`. An important point is that you can't specify the distance range that you are interested in directly. Instead you have to specify the angular degree range. A useful little project would be to automate the determination of the angular degree corresponding to a particular distance and frequency.

4.2 `run_raypath.m`

This script sets up the input for `raypath` and then calls that function. The user sets the value of frequency and angular degree to compute for by editing the first fews lines in `run_raypath.m`. Like with `run_tau` you cannot directly specify a distance to compute for. The function `raypath` does the main calculation and returns the radius, heliocentric angle, phase time, and group time along the ray path. Then `run_raypath` calls `ray_kernel` to compute the ray-based travel time-kernel for the given angular degree and frequency.

4.3 `ray_kernel.m`

The function `ray_kernel` does the calculation of the ray kernel, which is very simple and described by Birch (2002). Ray kernels are singular at both the upper and lower turning points, so the value of the kernel at those points is not well defined. Also, the kernel is very large near the surface where the sound gets very small.

5 Tests of the Code

The following figures show some aspects of the ray codes described in this document. The first three figures show the solar model that we used to obtain the results in this section. That particular model was given to the author by R. Nigam. The remaining figures shows some results from running `run_tau` and `run_raypath`. I will leave a description of these results to the figure captions.

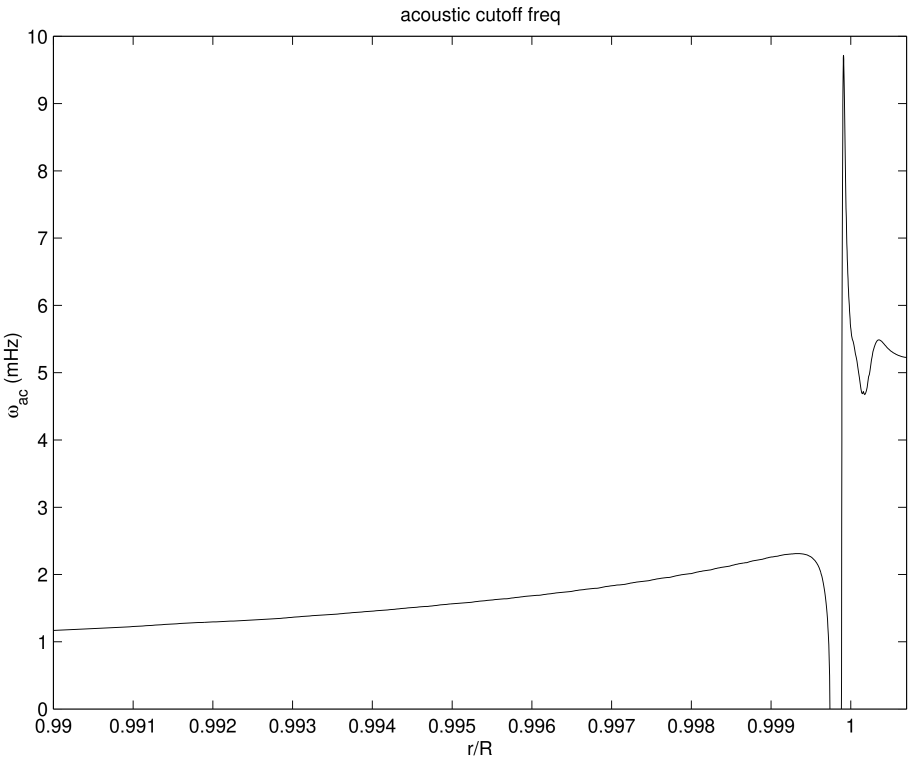

Figure 1: Acoustic cutoff frequency as a function of fractional radius. The cutoff frequency is only really important near the surface. As you go up into the atmosphere the cutoff frequency goes to 5 mHz. The bump around $r/R = 0.999$ can cause some trouble with low frequency rays, as it splits the acoustic cavity into two pieces. It is not at all clear what to do about this. The cutoff frequency shown here was given to the author by R. Nigam.

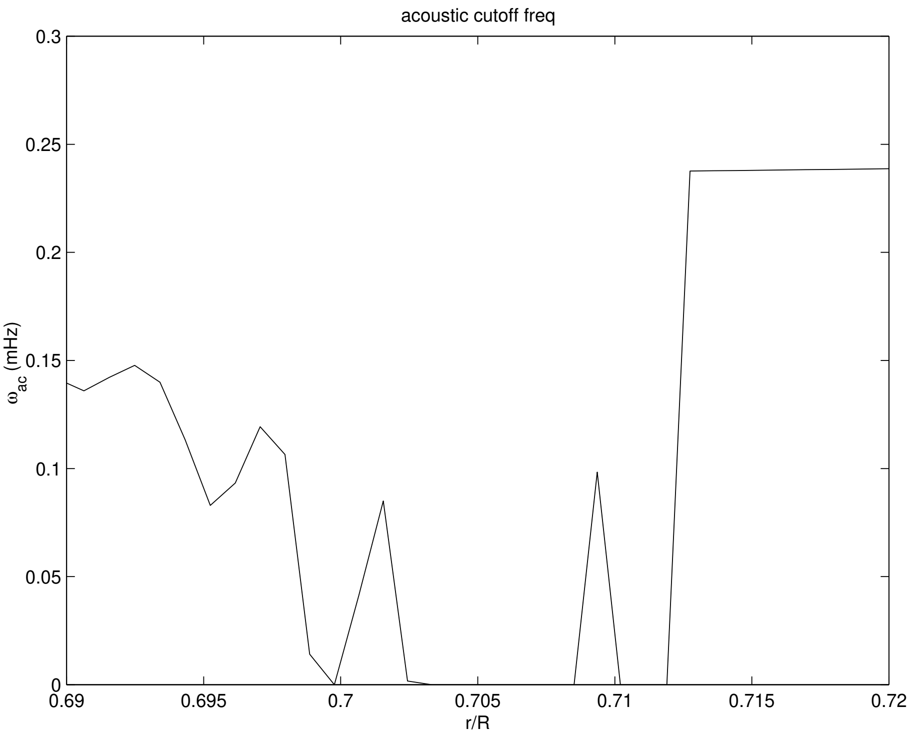

Figure 2: Acoustic cutoff frequency at the base of the convection zone. The oscillations are due to rapidly changing density. For rays with lower turning points at the base of the convection zone these might be important. For all other modes they are pretty irrelevant.

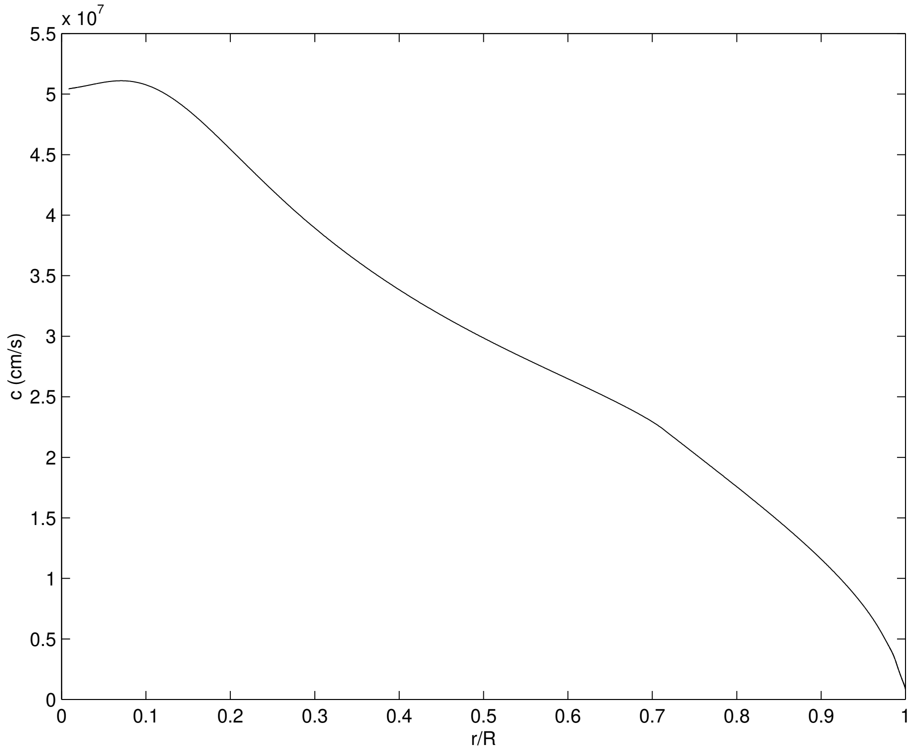

Figure 3: Sound speed as a function of depth.

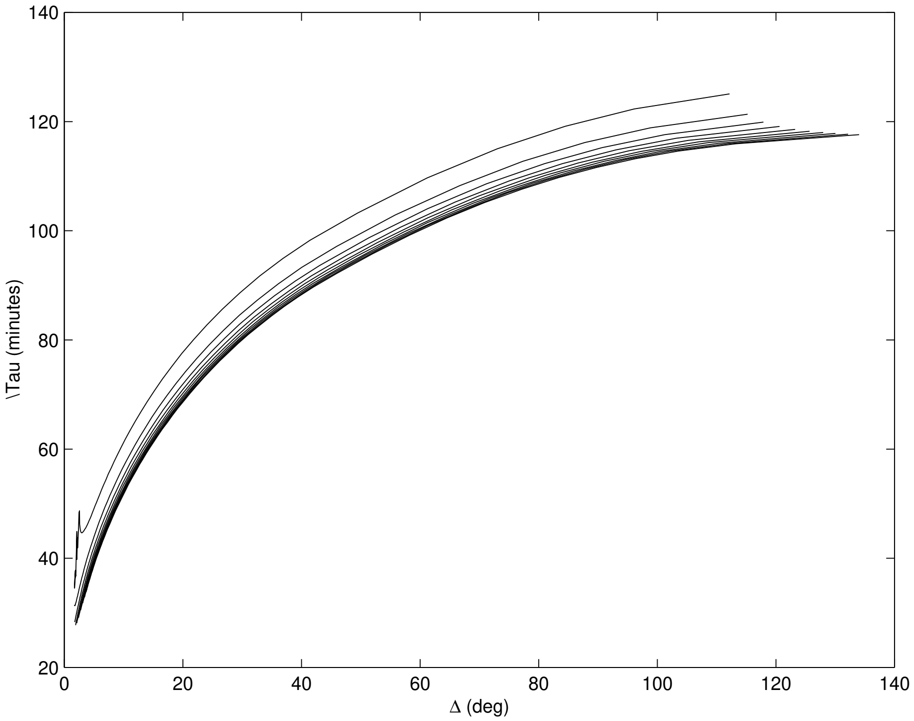

Figure 4: A time-distance plot. The different curves are for different frequencies, the slowest rays are those with low frequencies. The top line of this plot is for 2.4 mHz, the bottom is for 4.5 mHz. The weirdness of the low frequency rays for high l is due to the bump in acoustic cutoff frequency near the surface.

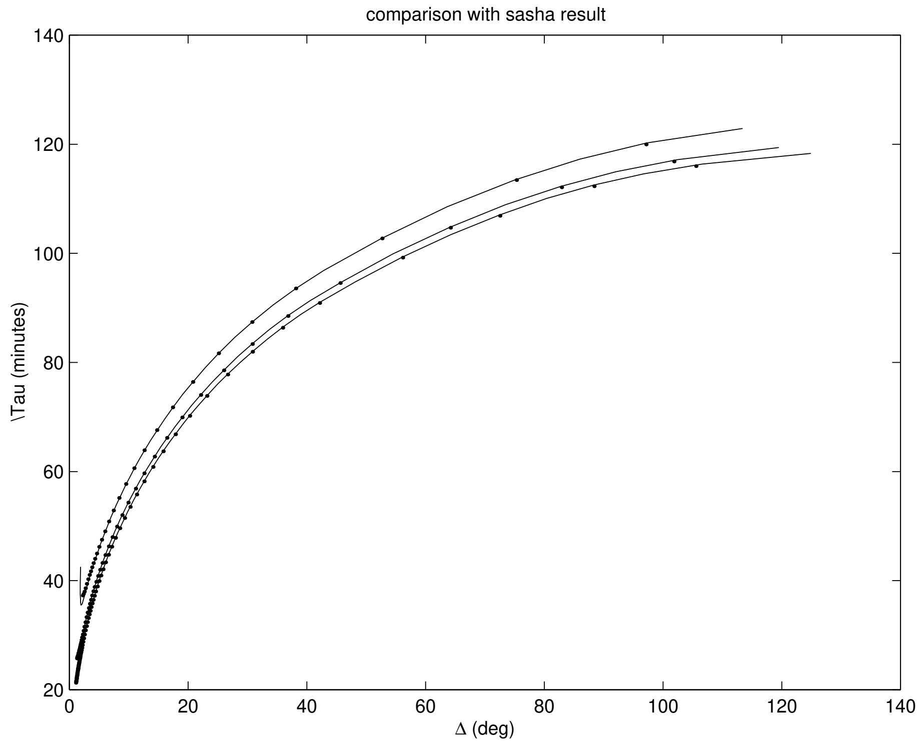

Figure 5: A time-distance plot. The lines are for calculations described here for $2.5, 3.0$ and 3.5 mHz rays. The dots were calculated by A. Kosovichev (private communication) for the same rays. They agree up to about 30 seconds. The best agreement is at small distances and high frequencies.

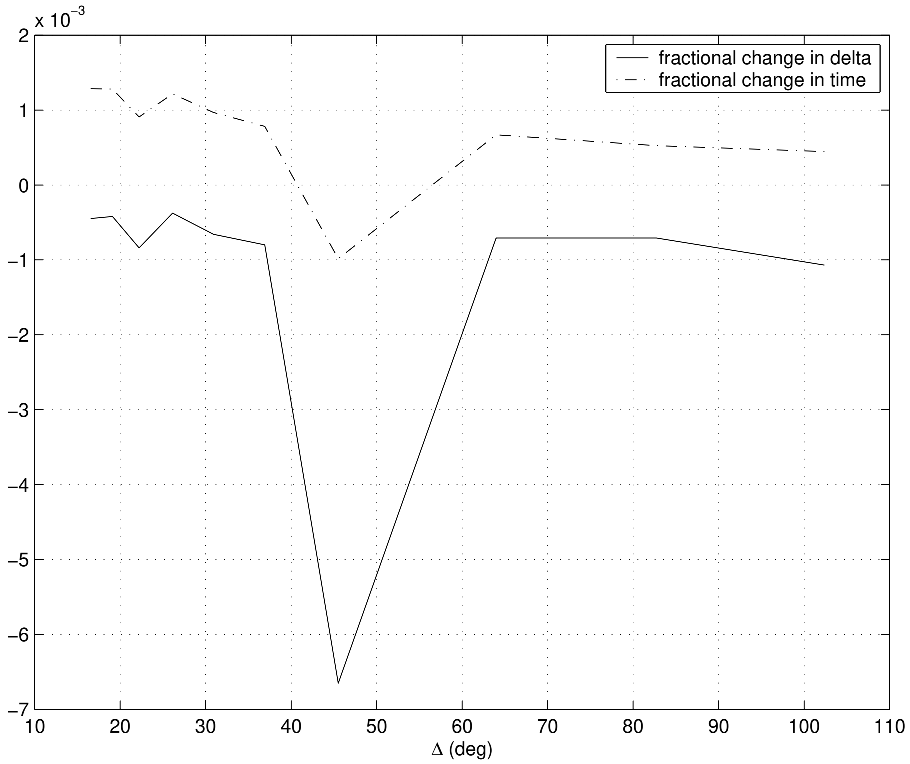

Figure 6: The effect of the parameter N in `tau.m` on the calculation of first-bounce times and distances. The fractional change from $N = 1$ to $N = 5$ in group time (dot-dash curve) and distance (solid curve) is shown as a function of first-bounce distance. Changing N can give almost a one percent effect.

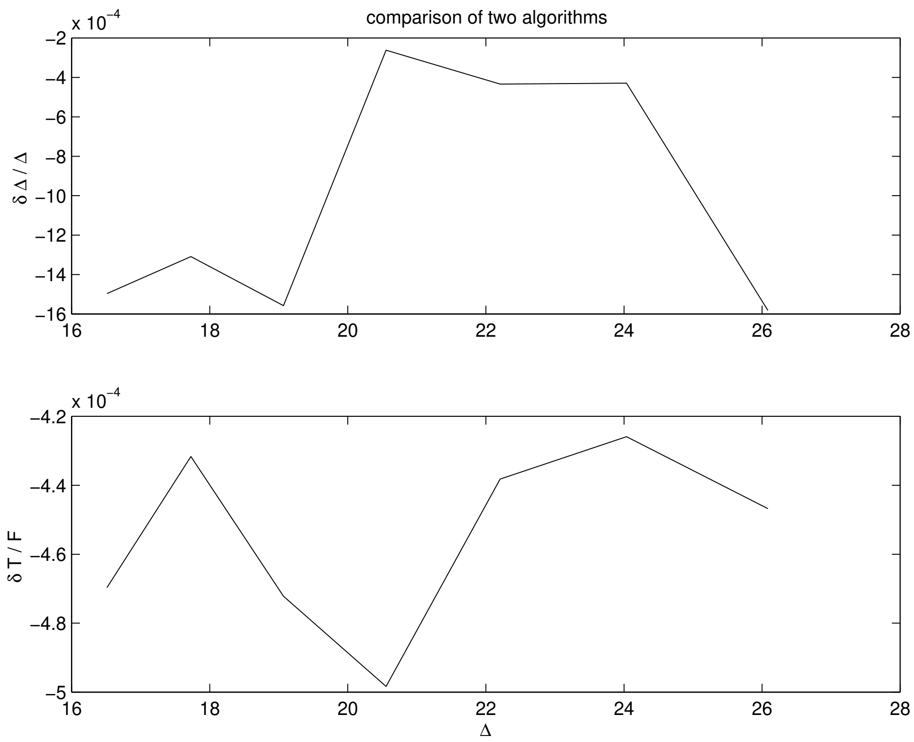

Figure 7: Fractional differences in group times (bottom panel) and first-bounce distances (top panel) from the `raypath.m` and `tau.m` codes. The horizontal axis in both cases is distance in degrees. The two codes give the same result to within 0.1 percent.

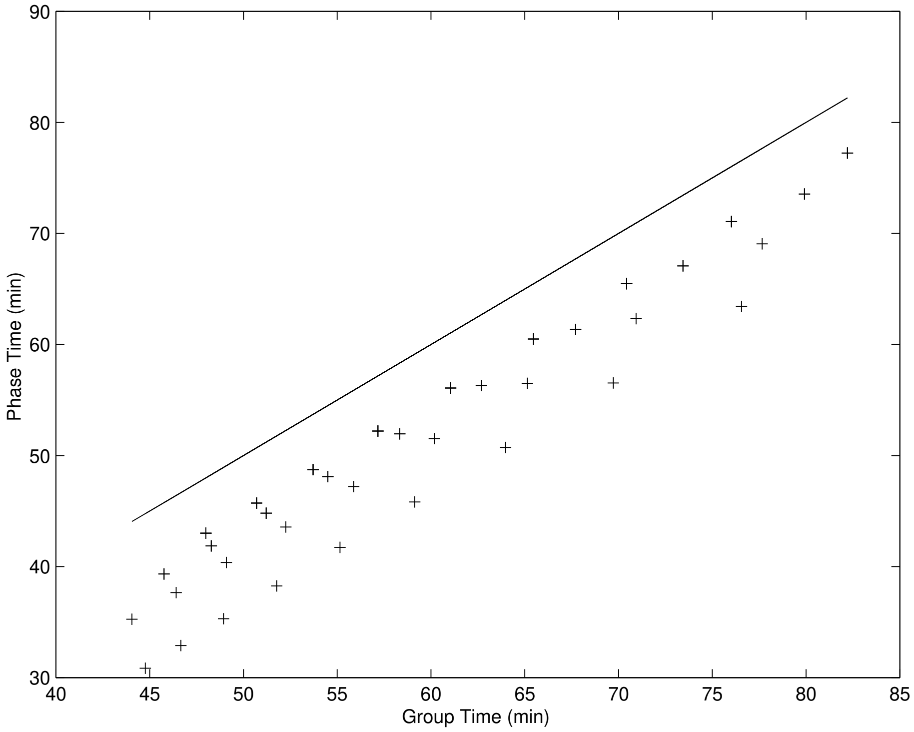

Figure 8: Comparison of the phase and group times for four different frequencies, 2.5, 2.833, 3.166, and 3.5 mHz. The solid line is where the phase time equals the group time. Notice that the phase time is always less than the group time. As the frequency increases the phase time increases up to the group time. The acoustic cutoff frequency becomes less important as the frequency increases.

References

Birch, A. C. 2002, PhD thesis, Stanford University

Christensen-Dalsgaard, J., Thompson, M. J., & Gough, D. O. 1989, MNRAS, 238, 481

D'Silva, S. & Duvall, Jr., T. L. 1995, ApJ, 438, 454

Giles, P. M. 1999, PhD thesis, Stanford University