1

ZAMS and PULS Computer Codes

This file contains documentation for the two `FORTRAN` computer codes called `ZAMS.FOR` and `PULS.FOR`, which are included as ASCII files in the directory `ZAMSPULS` on the CD attached to the back cover of this text. The first program makes ZAMS models which, in turn, may be analyzed for radial or nonradial pulsations in the second code. Also included on the CD are sample input and output files, executable versions of both codes for use on personal computers (PCs), and a `READ.ME` file which should be consulted first for further details.

1.1 The ZAMS Model Builder

The `FORTRAN` program `ZAMS.FOR` constructs homogeneous zero-age main sequence models using the fitting technique outlined in Chapter 7. It is not terribly sophisticated, but it will make very decent models. For those of you who would prefer to use either `C` or `PASCAL` the transition is not too difficult, but you will need a proficiency in `FORTRAN` to effect the translation. We have tested this program on personal computers (with a Microsoft compiler), UNIX workstations, and VAX VMS mainframes but cannot guarantee that your particular system will happily accept all that follows. We have made an effort to use standard (even old-fashioned) `FORTRAN` for portability. If you use an old PC, we suggest a fast one with a math coprocessor. On some machines you may have to work in extended precision (e.g., G–float in VAX FORTRAN). In addition, we have included what we hope is sufficient documentation within the code. Note that provision is made to write output quantities that may then be used for the radial and nonradial pulsation `PULS` code, which is discussed in the next section. `ZAMS` is written in double precision to provide a smooth and accurate model for the `PULS` code.

The running of the code is fairly straightforward but requires guesses of initial quantities. Some hints are given later. Because we use very simple photospheric conditions and the fit to opacity (c. 1970s) does not cover very cool temperatures, we do not recommend trying to construct models with low effective temperatures. Models with masses much below that of the sun will not be very successful. Also, do not try compositions that differ substantially from Pop I because the opacity fit does not cover a wide range of composition. You are, of course, welcome to substitute your own opacities, etc., including tabulated versions. Because the code assumes models in complete equilibrium, this code cannot be used to construct even warm white dwarfs (and the equation of state has no provisions for degeneracy). You are welcome to try to change things so that it might.

The program includes public domain routines from LINPACK that are used to solve systems of linear equations (in this case only a four-by-four system). If you are familiar with such routines and can identify the input and output quantities, you may substitute your own favorite programs. We also use a Runge-Kutta integrator called `RKF` to integrate all four stellar structure equations in this fitting program. (See the documentation before subroutine `RKF`.) `RKF`, and its associated routines, is a general-purpose integrator that works well with all but "stiff" systems of differential equations. You are again invited to use your own integrator (such as those in *Numerical Recipes* by Press et al., 1992).

There is quite a bit of output generated by this code and sample output is included on the diskette. As a function of $\mathcal{M}_r$ (actually $1 - \mathcal{M}_r/\mathcal{M}$) it will tabulate r, P, T, ρ, and $\mathcal{L}_r$, ε, κ, $\mathcal{L}_r(\text{conv})$, $\mathcal{L}_r(\text{conv})/\mathcal{L}_r(\text{tot})$, ∇, ∇_{ad}, and ∇_{rad} for 201 points in the model. With this information you should be able to figure out what makes the model tick.

Finally, we list guesses for central pressure, central temperature, total radius, and luminosity for ZAMS models of three different masses and two compositions (and see Tables 2.5 and 2.6). The compositions are $X = 0.74$, $Y = 0.24$, and $X = 0.70$, $Y = 0.28$. If the program has compiled properly (among other things), then these models should converge in one iteration of the fitting method. Note that poor guesses can be fatal. You are allowed `NTRY` attempts to converge and it is set to 15. If the code wants to take more iterations than this, then your initial guesses were probably not very good. If you are not satisfied with the composition used in Tables 2.5 and 2.6 then we advise you to creep up on a new composition by slowly adjusting X and Y away from that used in those tables and use graphical interpolation or extrapolation to obtain decent guesses.

Table 1.1. Some guesses for the `ZAMS` model builder: $X = 0.74$, $Y = 0.24$.

$\mathcal{M}/\mathcal{M}_\odot$	P_c	T_c	$\mathcal{R}$ (cm)	$\mathcal{L}/\mathcal{L}_\odot$
1	1.482×10^{17}	1.442×10^{7}	6.932×10^{10}	0.9083
3	1.141×10^{17}	2.347×10^{7}	1.276×10^{11}	89.35
15	2.769×10^{16}	3.275×10^{7}	3.289×10^{11}	1.960×10^{4}

Table 1.2. Some guesses for the ZAMS model builder: $X = 0.70$, $Y = 0.28$.

$\mathcal{M}/\mathcal{M}_\odot$	P_c	T_c	$\mathcal{R}$ (cm)	$\mathcal{L}/\mathcal{L}_\odot$
1	1.683×10^{17}	1.536×10^{7}	7.411×10^{10}	1.175
3	1.088×10^{17}	2.387×10^{7}	1.281×10^{11}	1.072×10^{2}
15	2.693×10^{16}	3.316×10^{7}	3.300×10^{11}	2.232×10^{4}

1.2 Adiabatic Pulsation Code

The program `PULS.FOR` does an adiabatic pulsation analysis of either nonradial modes in the Cowling approximation or radial modes for the full second order system (see Chapter 8). Most of the input is from `ZAMS` where the first block of data is, in order and usual notation, $x = \ln(r/P)$, r, g, and ρ. The quantity x is the independent variable and replaces r in the pulsation equations. It is used because it "stretches" the mesh at the surface and center to allow for better zoning in integration. (See Osaki and Hansen 1973, ApJ, 185, 277.) The second block of data from `ZAMS` consists of x, g/r, $(g/r)\ell(\ell+1)/S_\ell^2$, $(r/g)N^2$, U (of 7.52), and $(1+V)^{-1}$ (also of 7.52). These data are used as is in the nonradial part of the code but is modified (see below) for the radial portion.

The method of solution is by shooting from the center to the surface using the same `RKF` integrator used in `ZAMS`. The solution is started by guessing a value of the period, Π (in seconds), setting one eigenfunction to unity (and this is arbitrary because one normalization is arbitrary), and applying the central boundary condition to yield the value of the second eigenfunction at the center. With these starting values, the pulsation equations are then integrated until the surface is reached. Unless the eigenvalue has been guessed correctly, the outer boundary condition is not satisfied. A Newton's method is used to satisfy that boundary condition by iteration on the eigenvalue.

The non–`ZAMS` input, which you must provide for the code, is a guess for the period and an integer choice for ℓ. An ℓ of zero yields the radial case. The variables for the radial case are $y_1 = \delta r/r$ and $y_2 = \delta P/P$.

The nonradial case is a bit more complicated and uses the "Dziembowski variables" y_1 and y_2 discussed in Unno et al. (1989), §18.1 (but in the Cowling approximation). These are more stable for numeric computation then the horizontal and transverse displacements discussed in Chapter 8.

The output of `PULS` consists of information on convergence and, if desired, the eigenfunctions (called y_1 and y_2 in both radial and nonradial problems). For the radial case, y_1 and y_2 are printed out directly. The nonradial output is different. To obtain y_1 and y_2, you must multiply the tabulated output by $(r/\mathcal{R})^{\ell-2}$. This is not done in the code itself because for very high ℓ you get overflow problems. (The computational variables for the code are really

y_1 and y_2 divided by that factor.) In any case, the overall normalization is $y_1(\mathcal{R}) = 1$.

For guesses you might try the following. Both are for the $X = 0.74$, $Y = 0.24$ ZAMS sun listed in the model guesses given for the `ZAMS` program. For the fundamental radial mode ($\ell = 0$ and no nodes in the y_1 eigenfunction), try a period guess of 3583 seconds. (This period, of roughly an hour, corresponds to the dynamic or pulsation time scale for the sun as discussed in Chapter 1.) For a high ℓ nonradial mode try $\Pi = 254$ and $\ell = 100$. `PULS` should respond by saying this mode has a "phase diagram mode" of -10. The minus sign means that you have found a p-mode and the 10 implies that the order of the mode is 10; i.e., a p_{10} mode. The phase diagram method of classifying modes is discussed in Unno et al. (1989), §17. The results for p-modes you obtain should agree to within a few seconds of those of the present-day sun. You will, however, not get *excellent* agreement using `PULS` because it is a Cowling code and does not take into account gravitational perturbations. Note also that very high overtone modes may give you some difficulty because they have many wiggles in their eigenfunctions. Experiments you might wish to try include using models of different masses and compositions and looking for modes of low frequency (periods longer than the radial fundamental). The latter are gravity modes and tend to be concentrated in the inner portions of the model. The phase diagram mode for a gravity mode is signaled by a + sign. You should, for example, find a g_1, $\ell = 1$, mode at about 35,120 seconds period in the $15\,\mathcal{M}_\odot$ model generated by the guesses given for `ZAMS`. A first overtone radial mode (one node in y_1) is at 6814 seconds for this model. Other periods for upper ZAMS models may be found in Aizenman, Hansen, and Ross (1975), ApJ, 201, 387.