71-20,946

UNGAR, Stephen Gene, 1937-
NON-LINEAR NUCLEAR ENERGIZED STELLAR PULSATIONS.

The City University of New York, Ph.D., 1971
Astrophysics

University Microfilms, A XEROX Company, Ann Arbor, Michigan

NON-LINEAR NUCLEAR ENERGIZED STELLAR PULSATIONS

by

STEPHEN G. (GENE) UNGAR

A dissertation submitted to the Graduate Faculty in Physics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1971

This manuscript has been read and accepted for the Graduate Faculty in Physics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/22/70
date

Richard Stothers
Richard Stothers
Chairman of Examining Committee

1/22/70
date

Marvin H. Mittleman
Marvin H. Mittleman,
Acting Executive Officer

Professor Chao-wen Chin

Professor Ferdinand Shore

Professor Harold Stolov

Professor Chi Yuan

Supervisory Committee

The City University of New York

ACKNOWLEDGEMENTS

Before giving the customary acknowledgements, I must express my deepest appreciation to those who have played a vital,albeit nonprofessional, role in making this work possible. I wish to express my profound gratitude to my parents for their ever-present encouragement, and their willingness to make available to me opportunities denied them by the times and circumstances under which they spent their early lives. My brothers, Harold and Marvin, were instrumental in setting an example for me to emulate which stressed academic achievement. My wife and children graciously made many sacrifices, materialistic and otherwise. I am particularly indebted to my wife, Susan, for encouraging me to leave the security and relative comforts of steady employment to pursue the means that would allow me to do what I really wanted to. I also wish to thank her for undertaking and coping with the tedious (and sometimes thankless) task of typing the manuscript for this thesis. I am grateful to my children who, with consideration far in excess of what should reasonably be expected from youngsters of their age, permitted their daddy to work on his thesis; and I am particularly grateful to my oldest child, Peter, who

ungrudgingly accompanied his daddy to the "Institute" on many Saturday afternoons just to be able to spend a few minutes with him between computer runs. Lastly, I wish to thank my in-laws for showing great patience and being helpful in many ways.

The author wishes to express his appreciation to Dr. Richard Stothers, his thesis advisor, for suggesting the problem, providing guidance, and most of all for using his uncanny ability to always pinpoint appropriate references relating to every aspect of the problem. He also wishes to thank Dr. Norman Simon for discussions on the physics of pulsation as well as the results of linearized theory and Dr. Arnold Lapidus for many useful discussions on numerical technique. This research was supported in part by the National Aeronautics and Space Administration under contract NsG 197-67. It is a pleasure to thank Dr. Robert Jastrow for his hospitality at the Institute for Space Studies.

TABLE OF CONTENTS

* computer generated

LIST OF TABLES

LIST OF FIGURES

* computer generated plots

CHAPTER 1

INTRODUCTION

This dissertation concerns itself with the generation of nuclear energized radial pulsations in main sequence population I stars. This is to be distinguished from previous non-linear model calculations where the pulsational driving sources were discontinuities in the stellar opacity (i.e. ionization zones). Until recently the only studies of nuclear energized pulsations were done with linearized theory (Schwarzchild and Harm, 1959; Stothers, 1968; Stothers and Simon, 1968; Simon and Stothers, 1969).

There are basically three areas in which a linear analysis of nuclear driven pulsations will supply useful information:

(i) whether small perturbations will grow or damp and their rate of growth or damping;

(ii) the pulsational period of the fundamental mode and, with some additional patience, the overtones;

(iii) a relative radius amplitude distribution for each of the modes.

Interpretation of the information supplied by a linear analysis is limited by the following considerations:

(i) If the analysis indicates the models are pulsationally unstable, it is not known whether, in fact, non-linear effects will quickly limit the pulsational amplitudes, allow them to grow or even invoke some mechanism (e.g. mass loss) that will cause

the pulsations to damp.

(ii) For large amplitude pulsations, the period may be other than that indicated by the linear theory.

(iii) The linearized relative amplitude distributions, are not necessarily adequate representations of the distributions for large pulsations. It is fruitless to attempt to relate a normalized spatial distribution to observational parameters.

(iv) The region in which linearized theory is valid may correspond to exceedingly small amplitude fluctuations (Simon, 1970a and 1970b), which in turn would correspond to nearly unobservable pulsations in real stars.

In addition to the unreliability of the information supplied, there is a great deal of important information unascertainable from the linear analysis. A detailed knowledge of factors such as surface velocity, surface luminosity and mass ejection rates are necessary in order to adequately compare stellar models to actual stellar objects. It is only with a complete non-linear treatment that one can substantiate proposals such as the recent one of Stothers and Simon (1969) indicating that β Cephei variable stars are undergoing nuclear energized pulsations.

In addition to the foregoing considerations, our

non-linear investigation of nuclear energized pulsations was prompted by an unpublished study of the luminosity functions of very young stellar clusters conducted in 1968. An abbreviated summary of this study is given in APPENDIX E. It indicates a very sharp cutoff of young, spectral type O, main sequence stars at about bolometric magnitude -10, corresponding to a mass of 70 $M_{\odot}$. The study indicates that at least 9 superluminous objects should be present out of some 280 stars studied in one of the samplings. In fact, none were present in any of the samplings. The lack of objects beyond the cutoff magnitude must be attributed to non-genetic effects such as pulsational instability. It is interesting to note that Schwarzchild and Harm (1959) found that main sequence stars above 60 $M_{\odot}$ would be unstable against nuclear energized pulsations. What is even more interesting, is that Schwarzchild and Harm's linear analysis employed electron scattering as the sole source of opacity, which leads to a cutoff mass that is too small by roughly a factor of two (Ziebarth, 1970). The discrepancy between revised theory and our luminosity functions may be resolved by recent OAO and rocket observations indicating that the bolometric corrections for the more massive stars in the 1968 study were underestimated by anywhere from ½ to 1 full magnitude.

Although in the above instance linear theory accounts for the observational cutoff, it gives us no

clue as to what sort of objects young supermassive stars become. This problem may only begin to be approached after developing non-linear techniques.

A full non-linear hydrodynamic treatment has been successfully employed in studying stellar pulsations driven by ionization layers in connection with models of classical Cepheid and RR Lyrae variable stars (Christy, 1964, 1966a, 1966b, and 1967). However, the problem posed by nuclear energized pulsations in massive main sequence stars differs markedly from that of the objects Christy studied. The approach developed in this dissertation is therefore quite divergent from that of Christy. We employ a fully explicit numeric scheme, as explained in chapter six, giving us exceedingly high resolution in terms of following the time development of our models. Several unique techniques were instituted in the course of developing the computer codes as explained later on in the text. In general the code was designed to take full advantage of the high speed and extensive core storage available on the IBM System 360 Model 95 computer at the NASA Institute for Space Studies.

TABLE 1.1

FREQUENTLY USED SYMBOLS

NOTE: Values of all constants are taken from ASTROPHYSICAL QUANTITIES
by C.W. Allen
second edition, 1963
Athlone Press, London

SYMBOL	QUANTITY	VALUE OF CONSTANTS (in CGS units)
a	radiation constant	$7.5641 \cdot 10^{-15}$
a_o	e^2/mc^2, classical electron radius	$2.81785 \cdot 10^{-13}$
β	ratio of gas pressure to total pressure	
c	speed of light	$2.997929 \cdot 10^{10}$
C_q	viscous coefficient	
ε	nuclear energy generation rate per unit mass	
G	universal gravity constant	$6.673 \cdot 10^{-8}$
γ	ratio of specific heats	
Γ_1, Γ_2, & Γ_3	adiabatic exponents, see Chandrasekhar (1957)	
H	proton mass	$1.67333 \cdot 10^{-24}$
k	Boltzman Constant	$1.38046 \cdot 10^{-16}$
κ	opacity	
κ_e	$4\pi/3H \cdot a_o^2(1+X)$, electron scattering opacity	
L_r	luminosity at radius r (rate of energy flow)	

$L_\odot$	solar luminosity	$3.90 \cdot 10^{33}$
M_r	mass with radius r	
$M_\odot$	solar mass	$1.989 \cdot 10^{33}$
μ	mean atomic weight	
P	total pressure	
Q	artificial viscosity	
r	radius	
$R_\odot$	solar radius	$6.9598 \cdot 10^{10}$
ρ	density	
S	entropy	
T	temperature	
τ	optical depth	
v	velocity	
X	hydrogen abundance	
X_{CNO}	C N O abundance	
Y	helium abundance	
Z	"metallic" abundance	

CHAPTER 2

STELLAR STRUCTURE

The laws governing the structure of a star in hydrostatic and thermal equilibrium may be expressed by the following equations (Schwarzchild, 1958, p. 96):

$$\frac{dP}{dr} = -\rho\frac{GM_r}{r^2} \tag{2.1}$$

$$\frac{dM_r}{dr} = 4\pi r^2\rho \tag{2.2}$$

$$\frac{dL_r}{dr} = 4\pi r^2\rho\epsilon \tag{2.3}$$

$$\left.\begin{aligned} \frac{dT}{dr} &= -\frac{3}{4ac}\frac{\kappa\rho}{T^3}\frac{L_r}{4\pi r^2} \quad \text{(radiative)} \\ \frac{dT}{dr} &= \left(1 - \frac{1}{\Gamma_2}\right)\frac{T}{P}\frac{dP}{dr} \quad \text{(convective)} \end{aligned}\right\} \tag{2.4}$$

The first two differential equations are simply the conditions for hydrostatic equilibrium. The third is the thermal equilibrium condition. The fourth and fifth differential equations are two alternative forms for the energy transport condition.

It is well known that if the actual temperature gradient of a region in a star's interior is less steep than the adiabatic temperature gradient, this region will be stable against small scale (convective) motions (Clayton 1968, p. 254; Schwarzchild 1958, p. 44).

$$\left|\left(\frac{dT}{dr}\right)_{star}\right| < \left|\left(\frac{dT}{dr}\right)_{ad}\right|$$

Since stellar temperature gradients are generally negative the condition for convective stability of a layer may be expressed as:

$$\left(\frac{dT}{dr}\right)_{star} > \left(1 - \frac{1}{\Gamma_2}\right) \frac{T}{P} \left(\frac{dP}{dr}\right)_{star} \qquad (2.5)$$

If a stellar layer is composed of a non-degenerate convectively stable gas the mode of energy transport through the layer will be by radiative diffusion (op. cit.). Thus the first energy transport equation would be the appropriate one.

If the assumption that the mode of energy transport at a layer is radiative transfer leads to a superadiabatic temperature gradient, we must surmise that the layer will become convectively unstable and convective energy transport becomes important. It has been shown (**Chandrasekhar**, 1957, p. 225) that in a convectively unstable core the temperature gradient may be considered approximately adiabatic to obtain a fairly accurate estimate of the energy transport. In this case it would be appropriate to use the second energy transport equation.

To solve the above mentioned system of differential equations it is necessary to introduce several subsidiary conditions:

$P = P(\rho, T, X, Y)$ equation of state (2.6)

$\varepsilon = \varepsilon(\rho, T, X, Y)$ a set of equations for energy generation

$\kappa = \kappa(\rho, T, X, Y)$ relations for determining opacity

It is to be noted that in general the above three parameters depend on the thermodynamic variables and the stellar composition.

All that remains to solve the above system of equations is the specification of boundary conditions.

Physical continuity dictates that at a star's center

$$r = 0,\ M_r = 0,\ L_r = 0$$

The surface boundary conditions may not be as readily fixed. Since we are not primarily interested in equilibrium models we will save our consideration of surface boundary conditions until we discuss dynamic stellar models.

CHAPTER 3

STELLAR PHYSICS

3.1 STATE OF STELLAR GAS

It is customary to treat the state of the interior of a massive main sequence star as that of a totally non-degenerate completely ionized gas. Since the temperature everywhere up to the surface is well above 10^4 °K and the density from the center out well below 10 gm/cm^3, the assumption of non-degeneracy is obviously justified.

At temperatures above 10^6 °K helium is double ionized even at densities well above those occurring in our models. There are two areas of consideration where our assumption of complete ionization must be examined. Firstly, we must consider the extent of ionization of our major constituents (hydrogen and helium) in the relatively low temperature region near the surface. Secondly, we must consider the degree of ionization of heavier ("metallic") elements throughout the star.

The lowest photospheric temperatures we achieve in any of our models is in the vicinity of $4 \cdot 10^4$ °K. However, the corresponding photospheric density is of order of magnitude 10^{-9}gm/cm^3. It has been shown (Simon, 1970a) that in this temperature density domain all hydrogen is completely ionized and in excess of 0.99 of the helium will be completely ionized, with the remainder being singly ionized.

In general, in the outer regions of our models the metallic elements are only partially ionized, but, because of their low abundance, assuming total ionization will not affect the thermodynamic properties of the gas as a whole. The fact that some of the metallic atoms are only partially ionized plays an important role in determining opacity, and will be treated later.

The foregoing discussion has convinced us that it is valid to treat the equation of state for a massive main sequence star as that of an ideal gas. However, since the radiative flux is so great, we must take into account pressure due to the radiation field. Therefore

$$P = P_{gas} + P_{rad} = \frac{k}{\mu H}\rho T + \frac{a}{3}T^4 \qquad (3.1)$$

and for a completely ionized gas:

$$\frac{1}{\mu} = 2X + \frac{3}{4}Y + \frac{1}{2}Z$$

(Schwarzchild, 1958, p. 54); substituting 1-(X+Y) for Z

$$\frac{1}{\mu} = \tfrac{1}{2}(1 + 3X + \tfrac{1}{2}Y) \qquad (3.2)$$

In computing equilibrium models, P is calculated by use of the structure equations with values of the variables in the preceding zone (See Sec. 4.1).

T is calculated in a similar fashion. The equation of state (3.1) is then used to calculate ρ.

$$\rho = \frac{\mu H}{k}\left(\frac{P}{T} - \frac{a}{3}T^3\right) \tag{3.3}$$

In computing pulsational models the equation of state is made use of in a different manner. The variables brought forward in the time integrations are r, v, and S. We now define ρ as:

$$\rho = \frac{1}{4\pi r^2}\frac{dM}{dr}$$

We have derived (APPENDIX B) the following expression for S.

$$S = \frac{k}{\mu H}\ln\left(\frac{T^{3/2}}{\rho}\right) + \frac{4}{3}\frac{aT^3}{\rho} \tag{3.4}$$

(where we have set const = 0).

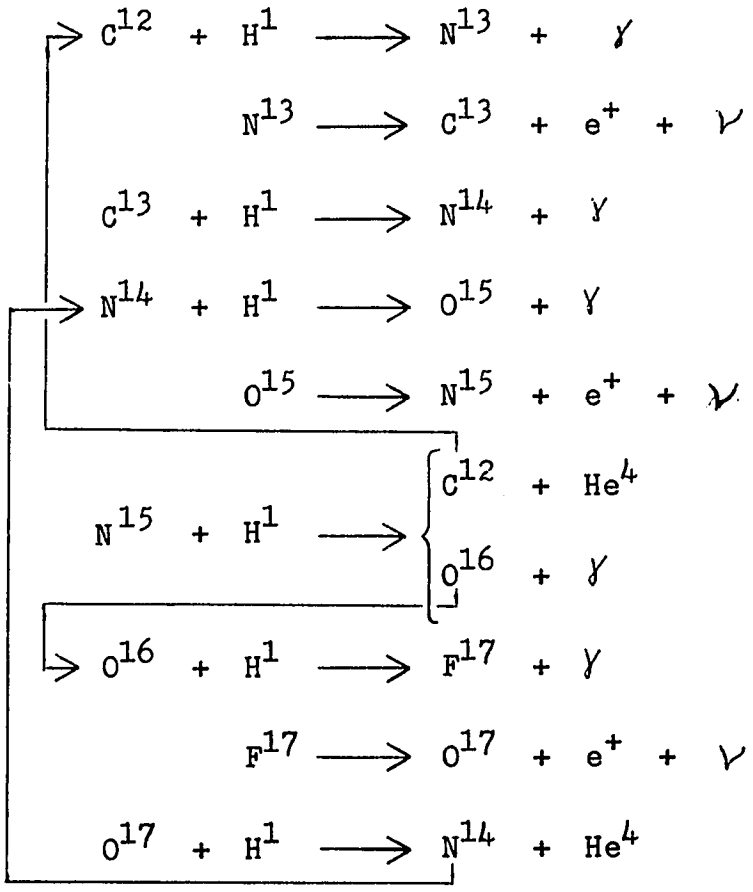
Equation (3.4) is now solved implicitly for T using a Newton-Raphson iterative procedure until $\delta T/T \leq 10^{-12}$. P may now be found directly from equation (3.1)

Summarizing, in the equilibrium schemes we use the equation of state to determine ρ, having determined P and T as variables of integration. In the dynamic schemes, we determine ρ and S from the variables of integration. Utilizing the temperature and density dependent expression for S along with the equation of

state we determine T and P.

3.2 NUCLEAR ENERGY GENERATION

The source of nuclear energy generation in massive ($>10M_{\odot}$) main sequence stars is assumed to be conversion of hydrogen to helium (hydrogen burning) by the Carbon-Nitrogen-Oxygen Bi-Cycle (Stothers, 1969). The reactions involved in this process may be represented as follows:



(3.5)

The following energy generation formula for the above

reactions is obtained from Reeves (1966)

$$\epsilon_{CNO} = (7.9\pm0.8)\cdot 10^{27} f_{14,1} g_{14,1} \rho X_{14} X_1 T_6^{-2/3} \exp(-152.31 T_6^{-1/3}) \text{ ergs/sec} \quad (3.6)$$

where: X_{14} = fractional mass abundance of N_{14}

$X_1 = X$ = hydrogen abundance

T_6 = temperature in units of 10^6 degrees Kelvin

$f_{14,1}$ is the electron screening factor and is given by:

$$f_{14,1} = 1 + 1.75\rho^{\frac{1}{2}}/T_6^{2/3} \quad (3.7)$$

and the correction term:

$$g_{14,1} = 1 + 10^{-3}(3T_6^{1/3} - 4T_6^{2/3}) \quad (3.8)$$

At temperatures and densities that typically occur in the interiors of the stars under consideration (i.e. $T_6 \sim 10$) it is readily seen that $f_{14,1}$ is slightly in excess of unity while $g_{14,1}$ is slightly less than unity. Their product is well within 10% of unity throughout the range of density and temperature values occurring in those regions of the stellar models considered where

significant contributions are made to the nuclear energy generation.

Since from the error figure indicated in the energy generation equation we see that the reaction rates are only determined to within 10%, it is questionable whether anything is to be gained by inclusion of the correction terms $f_{14,1}$ and $g_{14,1}$ in our calculations.

Furthermore, as indicated in Reeves (1965), $X_{14} \sim X_{CNO}$ for the conditions present in the prime energy producing regions of the stellar models under consideration. In all models calculated it was assumed that:

$$X_{14} = X_{CNO} = Z/2$$

where:

Z = the fractional mass abundance of all elements present other than hydrogen and helium (i.e. the "metal" abundance).

For initial calculations the following equation was used:

$$\varepsilon_{CNO} = 7.9 \cdot 10^{31} f_{14,1} g_{14,1} \rho X_{CNO}^{X} \cdot T^{-2/3} \exp(-15231 T^{-1/3}) \qquad (3.9)$$

where $f_{14,1}$ and $g_{14,1}$ were determined as indicated above. A representative sampling of the models calculated using

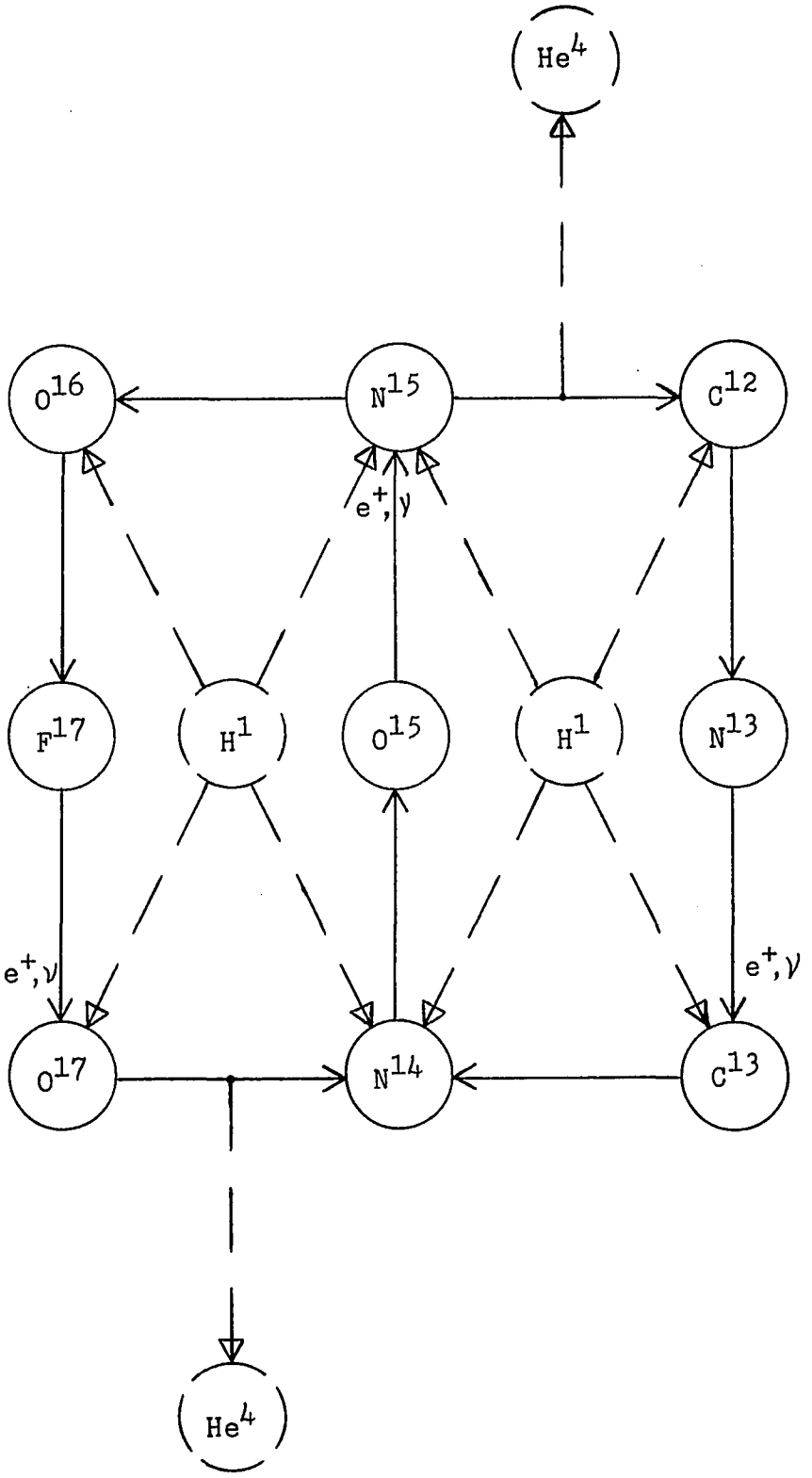
this energy generation equation were recalculated setting $f_{14,1} = g_{14,1} = 1$. Since no discernable differences were noted between the original and recalculated models, most of the later models were calculated with the energy generation equation:

$$\varepsilon_{CNO} = 7.9 \cdot 10^{31} \rho X \cdot X_{CNO} T^{-2/3} \exp(-15231 T^{-1/3}) \qquad (3.10)$$

X_{CNO} was taken to be $Z/2$ for all models discussed in this paper. Furthermore, since typically ε_{CNO} is down by six orders of magnitude from the value at a massive star's center when one reaches the boundary of the convective core, it was frequently assumed that $\varepsilon_{CNO} = 0$ throughout the envelope or some region of the envelope above some set mass fraction (e.g. the mass fraction corresponding to the equilibrium model's fitting point mass).

FIGURE 3.1

C N O BI-CYCLE


He^4
O^{16}
N^{15}
C^{12}
e^+, γ
F^{17}
H^1
O^{15}
H^1
N^{13}
e^+, ν
e^+, γ
O^{17}
N^{14}
C^{13}
He^4

3.3 OPACITY

Many of the models constructed were done so with the simplifying assumption that the opacity was due solely to electron (Thomson) scattering. The remaining models enjoyed a more thorough treatment of opacity and shall hereafter be referred to as the "total" opacity models.

ELECTRON SCATTERING OPACITY

As has been previously stated, the gas composing massive stars is ostensibly completely ionized. The only place where bound-free transitions may play a considerable role is in the outer regions where we do not have complete ionization of the metallic elements. For massive stars, therefore, it is not unreasonable to assume that the opacity is due primarily to Thomson scattering by free electrons (Stothers, 1969).

The cross section for Thomson scattering may be given as (Eisberg, 1961, pp. 493-98)

$$\nabla_e = \frac{8\pi}{3}\left[\frac{e^2}{mc^2}\right]^2 \qquad (3.11)$$

Since the opacity represents the total interaction cross section per unit mass

$$\kappa_e \rho = \nabla_e N_e \qquad (3.12)$$

and: $$\kappa_e = \nabla_e \frac{N_e}{\rho}$$

but: $N_e = \frac{1}{2}(1+X)\frac{\rho}{H}$ (Schwarzchild, 1958, p. 58)

therefore: $\kappa_e = \frac{4\pi}{3H}[\frac{e^2}{mc^2}]^2(1+X)$ (3.13)

Now $\frac{e^2}{mc^2} \equiv a_o$ is the classical radius of the electron and takes on the value $2.81785 \cdot 10^{13}$cm.

After evaluating all the constants

$$\kappa_e = 0.198766(1+X) \quad (3.14)$$

Thus the value for electron scattering opacity is independent of the physical state of the gas and is composition dependent only to the extent of being a function of the hydrogen abundance (i.e. it is independent of Z, the metallic abundance).

The use of electron scattering opacity obviously greatly simplifies the construction of stellar models. However, when constructing the pulsational models the region of greatest departure from electron scattering corresponds to the region of maximum pulsational damping. For main sequence stars, the effect of using a more realistic opacity is to raise the critical mass for sustained pulsations (Stothers and Simon, 1970).

"TOTAL" OPACITY

Christy (1966b) fitted the following analytic formula to some tables of opacity available at the time:

$$\kappa = P_e[[\,5.4\cdot10^{-13}V/T_4 + X\{T_4^{\frac{1}{2}}(2\cdot10^6/T_4^4 + 2\cdot1\,T_4^6)^{-1} + [4.5\,T_4^6 + T_4^{-1}(4\cdot10^{-3}/T_4^4 + 2\cdot10^{-4}V^{\frac{1}{4}})^{-1}]^{-1}\} + Y[(1.4\cdot10^3T_4 + T_4^6)^{-1} + 1.5(10^6 + 0.1\,T_4^6)^{-1}] + Z[T_4^{\frac{1}{2}}(20T + 5T_4^4 + T_4^5)^{-1}]\,]] \qquad (3.15)$$

where: T_4 is the temperature in units of 10^4 degrees K, and V is the specific volume. P_e is the electron pressure. Christy's formula, when compared to the widely accepted opacity tables of Cox and Stewart (1965) agrees rather well in the temperature-density domain corresponding to models of classical cepheids. However, when this formula is applied to our models, disagreement is marked, particularly in the high temperature range. This is probably due to the fact that at higher photon energies, Compton scattering must be used in place of Thomson scattering. The first term in the above formula reflects the temperature independent contribution of Thomson scattering and does not extrapolate to higher temperatures. In addition the form of the terms reflecting bound-free metal absorption is quite temperature dependent and is therefore only "locally" fitted in the formula.

In order to overcome these difficulties, several revisions to the above formula were proposed to the author by R. Stothers in 1969. The revised formulae were

tested against the Cox-Stewart Opacity Tables for a Cameron Mixture I (X = 0.739, Y = 0.240) gas. The following formula as revised by Stothers was adopted for use in our total opacity models:

$$\begin{aligned}\kappa = P_e\Big[\,& 4.85\cdot10^{-13}\left[\rho T_4(1 + 2.2\cdot10^{-5}T_4)\right]^{-1} \\ & + X\Big\{T_4^{\frac{1}{2}}(2\cdot10^{6}T_4^{-4} + 2.1T_4^{6})^{-1} \\ & + \left[4.5T_4^{6} + T_4^{-1}(4\cdot10^{-3}/T_4^{4} + 2\cdot10^{-4}/\rho^{\frac{1}{4}})^{-1}\right]^{-1}\Big\} \\ & + Y\left[(1.4\cdot10^{3}T_4 + T_4^{6})^{-1} + 1.5(10^{6} + 0.1T_4^{6})^{-1}\right] \\ & + Z\Big\{T_4^{\frac{1}{2}}\left[20T_4 + 35T_4^{4}\right. \\ & \left. + 380Z(2-Z)^{0.33}(1+X)^{0.33}\rho^{0.33}T_4^{4.71}\right]^{-1}\Big\}\,\Big]\end{aligned} \tag{3.16}$$

where the electron pressure:

$$P_e = N_e kT = \tfrac{1}{2}(1+X)\frac{\rho}{H}kT \tag{3.17}$$

The results of checking the above formula against the opacity tables for the region of interest are shown in Table (3.1). It is to be noted that the formula consistently gives rise to values that are well within the 30% accuracy range of the tables. A thorough investigation of the behavior of a slight variant of this formula has subsequently been made by Stothers and Simon (1970). In their version of the formula, the coefficient

TABLE 3.1

STOTHERS-CHRISTY OPACITY FORMULA

COMPARED TO COX-STEWART OPACITY TABLES

Values listed represent

$$\frac{\kappa_{Stoth} - \kappa_{Cox}}{\kappa_{Cox}}$$

T \ ρ	10	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
$5.80 \cdot 10^4$							-0.186	-0.063
$8.12 \cdot 10^4$						-0.501	-0.217	-0.038
$1.16 \cdot 10^5$						-0.352	-0.184	-0.007
$2.32 \cdot 10^5$				0.188	-0.271	-0.050		
$5.80 \cdot 10^5$				-0.097	0.153	0.002		
$8.12 \cdot 10^5$				0.017	0.239	-0.002		
$1.16 \cdot 10^6$				0.125	0.096	0.000		
$2.32 \cdot 10^6$			-0.135	0.057	0.035	0.005		
$4.64 \cdot 10^6$		-0.020	0.073	0.033	0.012			
$1.16 \cdot 10^7$		0.028	0.010	0.007				
$4.64 \cdot 10^7$	0.008	0.010						

18 replaces the expression $38OZ(2-Z)^{0.33}$. Strictly speaking, the two versions will produce identical results only for models where $Z \sim 0.038$, but the results do not differ significantly for other values of Z within the range tested by the author.

3.4 LUMINOSITY

In the region above the convective core the mode of energy transport is assumed to be entirely radiative and the luminosity may be calculated from the radiative transport equation (Schwarzchild, 1959 p. 96)

$$\frac{dT}{dr} = \frac{-3}{4ac} \frac{\kappa\rho}{T^3} \frac{L_r}{4\pi r^2} \tag{3.18}$$

We may state the luminosity in a form suitable for a Lagrange formulation by noting that $dr = dM_r/4\pi r^2\rho$

$$L_r = - (4\pi r^2)^2 \frac{ac}{3\kappa} \frac{dT^4}{dM_r} \tag{3.19}$$

The luminosity in the convective core is found in a unique manner. Since it was felt desirable to retain continuity of coding scheme between core and envelope, it was decided to solve for the core luminosity distribution explicitly. If this is done, all other quantities in the core and envelope may be found with identical equations. In fact, there is no reason to distinguish core from envelope, except when considering luminosity. For purposes of determining energy transport in convective cores it is customary to approximate the temperature distribution as adiabatic (Chandrasekhar, 1957, p. 125).

$$\frac{dT}{dM} = \left(1 - \frac{1}{\Gamma_2}\right) \frac{T}{P} \frac{dP}{dM} \qquad (3.20)$$

Assuming a more realistic temperature distribution will probably change luminosity of the models considered in this paper by less than 10% (Boury, Gabriel and Ledoux, 1964). It is felt that a more realistic treatment of convection is unlikely to qualitatively affect any of the reults herein. The not inconsiderable complications introduced by detailed description of convection shall await future consideration along with other planned refinements. In addition it is assumed that the boundary of the convective core retains its identity, an assumption that is justifiable when considering that typical mixing times are much greater than the pulsational periods for the models considered (Boury, Gabriel and Ledoux, 1964).

Equation (3.20) can be restated quite simply as

$$\frac{dS}{dM} = 0, \qquad M \leq M_{convec.} \qquad (3.21)$$

Taking the spatial derivative of each side of equation (5.2) we have

$$\frac{d^2S}{dMdt} = \frac{d}{dM}\left\{\frac{1}{T}\left(\varepsilon - \frac{dL}{dM}\right)\right\}$$

but in the convective core

$$\frac{d}{dM}\left(\frac{dS}{dt}\right) = \frac{d}{dt}\left(\frac{dS}{dM}\right) = 0$$

$$\therefore \quad \frac{1}{T}\left\{\frac{d\varepsilon}{dM} - \frac{d^2L}{dM^2}\right\} - \frac{1}{T^2}\frac{dT}{dM}\left\{\varepsilon - \frac{dL}{dM}\right\} = 0$$

$$\frac{d^2L}{dM^2} - \left(\frac{1}{T}\frac{dT}{dM}\right)\frac{dL}{dM} + \left[\frac{\varepsilon}{T}\frac{dT}{dM} - \frac{d\varepsilon}{dM}\right] = 0 \qquad (3.22)$$

At the model's center $M = 0$, $L_o = 0$. (3.23)

Since the luminosity must be continuous across the core boundary at $M = M_{con}$,

$$L_{con} = -(4\pi R_{con}^2)^2 \frac{ac}{3\kappa}\frac{dT^4}{dM}\Big|_{M_{con}} \qquad (3.24)$$

where M_{con} = mass boundary of convective core:

We may solve for ε and T as functions of M (and hence $\frac{d\varepsilon}{dM}$ and $\frac{dT}{dM}$) at any given time step before solving for L.

Hence, the problem of determining the luminosity distribution in the core reduces to solving a second order linear differential equation between two specified boundaries; the luminosity at the outside boundary being determined by the radiative transport condition.

There are many standard ways of solving such equations. We chose to use a method outlined by McCormick and Salvadore (1964). The finite difference scheme utilized equal intervals in M which effectively reduced the whole problem to the solution of a band diagonal matrix with three elements per row.

3.5 STELLAR BOUNDARIES

For equilibrium models it is convenient to consider the photosphere as a boundary. Then,

$$T_R = T_e = \left(\frac{4}{ac}\frac{L}{4\pi R^2}\right)^{\frac{1}{4}} \tag{3.25}$$

As discussed in section (4.1), the photosphere may be considered as being at optical depth 2/3

$$\tau_R = \int_R^\infty \kappa\rho dr = \frac{2}{3} \tag{3.26}$$

If we equate the photospheric pressure to the weight of the overlying atmosphere

$$P_R = \int_R^\infty g\rho dr \tag{3.27}$$

Since the mass of the atmosphere is concentrated near R (see APPENDIX C) because of the small density scale height,

$$P_R \simeq \frac{GM}{R^2}\int_R^\infty \rho dr \tag{3.28}$$

Assuming $\kappa \sim \kappa_R$ throughout atmosphere and combining equations (3.26) and (3.28)

$$P_R = \frac{2}{3}\frac{GM}{\kappa_R R^2} \tag{3.29}$$

Thus we may determine the photospheric temperature and pressure from equations (3.25) and (3.29). If we are not interested in the stellar atmosphere, the photosphere serves as the outer boundary of our model. If we wish to calculate an approximate atmosphere, the photosphere is used as an inner boundary and after we compute our model between center and photosphere the atmosphere is constructed by outward integration from the photosphere.

We will now consider in somewhat more detail the outermost layers of massive stars. If we substitute $d\tau$ for $\kappa\rho dr$ in equation (3.18) we get:

$$\frac{dT^4}{d\tau} = \frac{3}{4\pi r^2}\frac{L_r}{ac} \qquad (3.30)$$

If we restrict our discussion to the region above the photosphere (r>R), we may utilize the fact that even after many density scale heights $(r-R)/r \gg 1$ and $L_r/L \simeq 1$.

$$\frac{dT^4}{d\tau} \simeq \frac{3}{4\pi R^2}\frac{L}{ac} \qquad \text{for} \quad r \geq R \qquad (3.31)$$

From equation (3.25) we now get

$$\frac{dT^4}{d\tau} \simeq \frac{3}{4}T_e^4 \qquad (3.32)$$

Assuming the radiation field is isotropic throughout the region $r > R$ we find the photosphere is at optical depth $\tau = 2/3$, which in conjunction with the above considerations gives us the so-called Eddington approximation (see Eddington, 1926, pp. 322-30)

$$T_e^4 - T_o^4 = \frac{3}{4} T_e^4 \int_o^{2/3} d\tau = \frac{1}{2} T_e^4$$

where: $T_o \equiv$ boundary temperature

(i.e. temperature at $\tau = 0$)

Thus, $T_o^4 = \frac{1}{2} T_e^4$ or $T_e = 1.189\, T_o$ (3.33)

substituting this back into equation (3.32)

$$\frac{dT^4}{d\tau} \simeq \frac{3}{2} T_o^4 \qquad \text{for } \tau < 2/3 \tag{3.34}$$

The above equation is often referred to as the Eddington approximation and, as indicated, is equivalent to placing the photosphere at optical depth 2/3.

In the same spirit as the Eddington approximation we may consider a boundary pressure due to the presence of the radiation field

$$P_o \simeq \frac{a}{3} T_o^4 = \frac{a}{6} T_e^4 \quad ,$$

leading to a modification of equation (3.29),

$$P_R = \frac{2}{3} \frac{GM}{\kappa_R R^2} + \frac{a}{6} T_e^4 \tag{3.35}$$

The pulsational problem amounts to the integration in time of a set of first order differential equations from an initial distribution between specified boundaries. At the inner boundary $m = 0$, $r = v = \frac{dS}{dm} = 0$. If we choose the photosphere as the outer boundary we may set up Neuman type boundary conditions relating $\frac{dT}{dm}\Big|_M$ to P_R.

Substituting $d\tau = -\frac{\kappa\rho}{4\pi r^2}dm$ into equation (3.32), gives us:

$$T_e^4 = -\frac{16\pi R^2}{3\kappa_R}\frac{dT^4}{dm}\Big|_M \qquad (3.36)$$

Generalizing equation (3.35) we must add the dynamic acceleration to the gravitational acceleration term:

$$P_R = \frac{2}{3\kappa_R}\left(\frac{GM}{R^2} + \frac{\partial^2 R}{\partial t^2}\right) + \frac{aT_e^4}{6}$$

but the expression in parenthesis is nothing more than the surface pressure gradient:

$$P_R = \frac{a}{6}T_e^4 - \frac{8\pi R^2}{3\kappa_R}\frac{dP}{dm}\Big|_M \qquad (3.37)$$

To equations (3.36) and (3.37) we may add the following two relations:

$$\rho_R = \frac{\mu H}{kT_e}\left(P_R - \frac{a}{3}T_e^4\right) \qquad (3.38)$$

$$\kappa_R = \kappa(\rho_R, T_e) \tag{3.39}$$

Since in any differencing scheme $\frac{dT^4}{dm}\Big|_M$ and $\frac{dP}{dm}\Big|_M$ will be expressible in terms of T_e, P_R and known values of T and P evaluated at other mass points, we may in principle solve equations (3.36), (3.37), (3.38) and (3.39) simultaneously. The exact approach to this problem will depend on the numerical scheme employed and its discussion is deferred until section (6.1).

The above method of describing the stellar boundary will henceforth be referred to as the "Photospheric Boundary Conditions". A more general method was employed in specifying the "outer" boundary for some of the earlier models. It was assumed that the outer boundary was somewhere above the photosphere, and expressing equation (3.36) in terms of the boundary temperature

$$T_R^4 = - \frac{8\pi R^2}{3\kappa_R} \frac{dT^4}{dm}\Big|_M \tag{3.40}$$

This is identical to the temperature boundary condition employed by Christy (1964, and 1966a) in his dynamic calculations of classical Cepheids. Christy imposed a zero pressure boundary on his models. Because of the strong radiation present in our models it is assumed

$P \sim T^4$ in the outermost layers leading to

$$P_R = - \frac{8\pi R^2}{3k\epsilon_R} \left. \frac{dP}{dm} \right|_M \qquad (3.41)$$

Equations (3.40) and (3.41) comprise what we refer to as the "Eddington Boundary Conditions". For testing purposes, some models were constructed with equation (3.41) replaced by the simple condition $P_R = 0$. These are referred to as the "Zero Pressure Boundary Conditions" models.

CHAPTER 4

EQUILIBRIUM MODELS

4.1 GENERAL APPROACH

The hydrodynamical equations are most conveniently written down in Lagrange form (i.e. with respect to a coordinate system fixed to the star). A useful spatial coordinate is M which represents the mass interior to the shell where the hydrodynamic variables P(M,t), T(M,t), L(M,t), R(M,t), etc. are being considered. In order to construct equilibrium models consistent with the dynamic code it is necessary to recast the structure equations of Chapter 2 in a form with M as the independent variable. From Equation (2.4) we find:

$$dM \equiv dM_r = 4\pi r^2 \rho \, dr$$

The stellar structure equations may now be stated as:

$$\frac{dP}{dM} = -\frac{GM}{4\pi r^4} \tag{4.1}$$

$$\frac{dL}{dM} = \mathcal{E} \tag{4.2}$$

$$\frac{dT}{dM} = \begin{cases} \left(\frac{dT}{dM}\right)_{RAD}, & \text{if } -\left(\frac{dT}{dM}\right)_{RAD} \leq -\left(\frac{dT}{dM}\right)_{AD} \\ \left(\frac{dT}{dM}\right)_{AD}, & \text{if } -\left(\frac{dT}{dM}\right)_{RAD} > -\left(\frac{dT}{dM}\right)_{AD} \end{cases} \tag{4.3}$$

where:
$$\left(\frac{dT}{dM}\right)_{RAD} = -\frac{3\kappa}{4ac} \frac{L}{(4\pi r^2)^2} \frac{1}{T^3} \tag{4.4}$$

$$\left(\frac{dT}{dM}\right)_{AD} = -\left(1 - \frac{1}{\Gamma_2}\right)\frac{T}{P}\frac{GM}{4\pi r^4} \tag{4.5}$$

$$\frac{dr}{dM} = \frac{1}{4\pi r^2 \rho} \tag{4.6}$$

From the Equation of State (see Section 3.1) we may find ρ for a given P and T:

$$\rho = \frac{\mu H}{kT}\left[P - \frac{a}{3}T^4\right] \tag{4.7}$$

we may also specify

$$\Gamma_2 = 1 + \frac{(4-3\beta)(\gamma-1)}{\beta^2+3(\gamma-1)(1-\beta)(4+\beta)}$$

(see Chandrasekhar, 1957, p. 57)

where: $\beta = P_{gas}/(P_{gas} + P_{RAD})$ (4.8)

and since for a monatomic ideal gas $\gamma = 5/3$

$$\Gamma_2 = \frac{8/3-2\beta}{\beta^2+2(1-\beta)(4+\beta)} \tag{4.9}$$

Section (3.3) discusses the form of

$$\kappa = \kappa(\rho, T) \tag{4.10}$$

Section (3.2) discusses the form of

$$\varepsilon = \varepsilon(\rho, T) \tag{4.11}$$

For completeness we may compute

$$S = S(\rho, T) \tag{4.12}$$

(See APPENDIX B). Although specification of S is not necessary for solution of the foregoing structure equations, it will be used to initialize the dynamic models.

At the star's center (i.e. at M = 0) L = 0, r = 0, $P \equiv P_c$, $T \equiv T_c$.

Therefore, if we are able to specify P_c and T_c, in principle the foregoing system of equations may be solved as an initial value problem in four simultaneous first order ordinary differential equations. We have specified the four variables of integration at M = 0 and all we need do is integrate forward in M until we reach $M = M_{star}$.

Of course the fatal drawback to this scheme is that there is no way of exactly specifying P_c and T_c a priori. However, if we choose to integrate only up to the star's photosphere we know that:

$$L = 4\pi R^2 \frac{ac}{4} T_R^4 \qquad (\text{i.e. } T_R = T_{eff}) \qquad (4.13)$$

from the definition of photosphere, and

$$P_R = \frac{2}{3\kappa_R} GM/R^2 \qquad (4.14)$$

from the assumption that the photosphere is at optical depth 2/3 (see Schwarzchild, 1958 pp 89-90).

Thus it seems that by choosing the best ad hoc values of P_c, T_c we may make a series of trial integrations

up to the photosphere seeing which trial values of P_c, T_c lead to the solutions which best satisfy equations (4.13) and (4.14). We might then, either by trial and error or through some error matrix eigen-value scheme, try to find the correct P_c, T_c by successive approximations. Unfortunately, it has been shown (op. cit. p. 102) that minute changes in P_c or T_c lead to large divergences of the solution in the envelope region. This is due to the fact that the envelopes of massive stars are in radiative equilibrium, and as we approach the surface, T becomes smaller and smaller in the denominator of the radiative Equation (4.4).

An alternative initial value problem approach would be to consider the initial boundary as the photosphere, $M = M_{star}$; choose L and R; from equations (4.13) and (4.14) calculate T_R and P_R; and now integrate backward to M = 0.

Integrating inwards we first find good convergence of the solutions, but as we approach the center r vanishes and a divergence occurs because of the occurrence of r in the denominator of several equations.

The approach used was a combination of the two preceding ones. Trial values of P_c and T_c were chosen and integrations were made outward to some fitting point where, typically, M_f was chosen to be something like $0.9\,M_{star}$. Then integrations inward to the fitting point

were made with several trial values for L and R. On the basis of the differences between the results at the fitting point of the inward and outward integrations, new values of P_c, T_c, L and R were chosen. This process was continued until the differences were considered negligible. The exact method used is outlined in the next section.

Since any of the integrations described above may be considered the solution to a set of simultaneous first order ordinary differential equations with initial values specified, the problem lends itself well to a numerical scheme such as that of Runge-Kutta. In particular we employed the fourth order Runge-Kutta scheme outlined immediately below.

Let y represent either P,T,L or r. Then any one of the Equations (4.1) to (4.6) may be written in the generalized form:

$$\frac{dy}{dM} = f_y(M,P,T,L,r)$$

If the current values of our variables are

$y_o = (P_o, T_o, L_o, r_o)$ at M_o, and we wish to find

$y = (P,T,L,r)$ at $M = M_o + \Delta M$, fourth order Runge-Kutta leads to the following procedure:

define $F_y(i) \equiv 0$, for $i = 0$

$$F_y(i) = f_y(M_i, P_i, T_i, L_i, r_i), \quad i = 1,2,3,4$$

where: $M_i = M_o + A_i \Delta M$

$y_i = y_o + A_i F_y(i-1)$

and $A_i = 0, \frac{1}{2}, \frac{1}{2}, 1,$ for $i = 1,2,3,4$

then: $y = y_o + [F(1) + 2F(2) + 2F(3) + F(4)]/6$

at $M = M_o + \Delta M$

It is obvious that this powerful method may not be directly applied to the outward integrations since $f_y(0, P_c, T_c, L_c, r_c)$ is singular for y = P,T and r. In order to surmount this difficulty we start the outward Runge-Kutta integration at some very small distance from the center. We may obtain the values of P,T,L, and r at this point from a simple series expansion; ignoring the higher order terms:

$$M \simeq \Delta M$$

$$r \simeq \left(\frac{3}{4\pi} \frac{\Delta M}{\rho_c}\right)^{1/3}$$

$$P \simeq P_c - \frac{2\pi}{3} Gr^2 \rho_c^2$$

$$T \simeq T_c - \frac{2\pi}{3} Gr^2 \rho_c^2 \left[1 - \frac{1}{(\Gamma_2)_c}\right] \frac{T_c}{P_c}$$

$$L \simeq \varepsilon_c \Delta M$$

The above approximations are readily obtained from inspection of the Structure Equations.

No comparable difficulty arises for the inward integrations.

4.2 FITTING POINT SCHEME

Ezer and Cameron (1963) initiated a fitting point procedure utilizing successive solutions to an eigen-value problem in the study of the early contracting phase of solar evolution. The following represents a major modification of this procedure. A total of six integrations per step is required as opposed to seven in the Ezer-Cameron method. In general, a difference of less than 1 part in 10^{12} between determination of the variables at the fitting point by outward and inward integrations is achieved within eight steps.

We will perform outward and inward integrations to a specified point M_f with the following sets of trial eigen-values:

(1) P_c, T_c, L, R

(2) $P_c+\delta P_c$, T_c, L, R

(3) P_c, $T_c+\delta T_c$, L, R

(4) P_c, T_c, $L+\delta L$ R

(5) P_c, T_c, L, $R+\delta R$

Note that outward integrations with set 4 or 5 will produce the same fitting point values as outward integrations with set 1. Similarly sets 2 and 3

produce the same fitting point values as set 1 for the inwards integrations. Therefore, in order to compare the results of inward and outward integrations with each set of trial eigenvalues on the variables P_f, T_f, L_f r_f at fitting point M_f we need only perform 6 integrations.

Let us define:

$$\left.\begin{aligned} \delta_i P_f &= P_{f_{out}} - P_{f_{in}} \\ \delta_i T_f &= T_{f_{out}} - T_{f_{in}} \\ \delta_i L_f &= L_{f_{out}} - L_{f_{in}} \\ \delta_i r_f &= r_{f_{out}} - r_{f_{in}} \end{aligned}\right\} \text{ for the } i^{th} \text{ set of trial eigenvalues}$$

Now consider P_c', T_c', L', R' as the "true" values of their unprimed counterparts, resulting in $\delta P_f = \delta T_f = \delta L_f = \delta r_f = 0$.

We may now define: $\Delta_i y = y' - y_i$

where: $y' = (P_c', T_c', L' \text{ or } R')$

and, y_i is the i^{th} trial eigenvalue corresponding to the quantity represented by y'.

We may express the deviations of the trial eigenvalues from their "true" values as functions of the fitting point differences between inwards and outwards integrations:

$$\Delta_i y = f_y(\delta_i P_f, \delta_i T_f, \delta_i L_f, \delta_i r_f)$$

If we have chosen our eigenvalues such that they are close to the "true" values, we should find that:

$$f_y(0,0,0,0) = 0$$

In any event, we shall shortly see that making this assumption leads to a determination of some set of "true" values which we may choose to be set # 1 of a new group of trial eigenvalues.

A simple first order Taylor series expansion leads to:

$$\Delta_i y = \left.\frac{\partial f_y}{\partial \delta P_f}\right|_i \delta_i P_f + \left.\frac{\partial f_y}{\partial \delta T_f}\right|_i \delta_i T_f + \left.\frac{\partial f_y}{\partial \delta L_f}\right|_i \delta_i L_f + \left.\frac{\partial f_y}{\partial \delta r_f}\right|_i \delta_i r_f$$

It is useful to abbreviate the derivatives such that:

$$\Delta_i y = f_y^P \delta_i P_f + f_y^T \delta_i T_f + f_y^L \delta_i L_f + f_y^r \delta_i r_f$$

Finally, we may recast our fitting point formulas into more useable form by introducing the following definitions:

$$\Delta y \equiv \Delta_1 y = \begin{cases} P_c' - P_c \\ T_c' - T_c \\ L' - L \\ R' - R \end{cases}$$

$$\nabla_i y \equiv \Delta_i y - \Delta y ,$$

leading to the set of equations:

$$\nabla_i y = f_y^P \delta_i P_f + f_y^T \delta_i T_f + f_y^L \delta_i L_f + f_y^r \delta_i r_f - \Delta y$$

The above set of equations may be handily represented by the matrix relationship;

$$\underleftrightarrow{\nabla} = \underleftrightarrow{\delta} \cdot \underleftrightarrow{f}$$

where:

$$\underleftrightarrow{\nabla} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\delta P_c & 0 & 0 & 0 \\ 0 & -\delta T_c & 0 & 0 \\ 0 & 0 & -\delta L & 0 \\ 0 & 0 & 0 & -\delta R \end{bmatrix}$$

$$\overleftrightarrow{\delta} = \begin{bmatrix} \delta_1 P_f & \delta_1 T_f & \delta_1 L_f & \delta_1 r_f & -1 \\ \delta_2 P_f & \delta_2 T_f & \delta_2 L_f & \delta_2 r_f & -1 \\ \delta_3 P_f & \delta_3 T_f & \delta_3 L_f & \delta_3 r_f & -1 \\ \delta_4 P_f & \delta_4 T_f & \delta_4 L_f & \delta_4 r_f & -1 \\ \delta_5 P_f & \delta_5 T_f & \delta_5 L_f & \delta_5 r_f & -1 \end{bmatrix}$$

$$\overleftrightarrow{f} = \begin{bmatrix} f_P^P & f_T^P & f_L^P & f_r^P \\ f_P^T & f_T^T & f_L^T & f_r^T \\ f_P^L & f_T^L & f_L^L & f_r^L \\ f_P^r & f_T^r & f_L^r & f_r^r \\ P'_c - P_c & T'_c - T_c & L' - L & R' - R \end{bmatrix}$$

Examination of the fitting point matrix $\overleftrightarrow{\delta}$ shows us that it is nonsingular, and since we have determined all its components we may find its inverse $\overleftrightarrow{\delta}^{-1}$. Our original matrix relation may now be written as:

$$\overleftrightarrow{f} = \overleftrightarrow{\delta^{-1}} \cdot \overleftrightarrow{\nabla}$$

and $P_c' = \overleftarrow{f}_{51} + P_c$

$T_c' = \overleftarrow{f}_{52} + T_c$

$L' = \overleftarrow{f}_{53} + L$

$R' = \overleftarrow{f}_{54} + R$

P_c', T_c', L', T' now form the basis for a new set of trial eigenvalues, and the process is repeated until $\delta_1 P_f$, $\delta_1 T_f$, $\delta_1 L_f$ and $\delta_1 r_f$ are less then some prescribed minimum.

EMPLOYING THE SCHEME

For massive main sequence models, initial trial values of P_c, T_c, R and L were taken from the electron scattering models of Stothers (1966). Simple linear interpolation was used to determine the values for models with masses intermediate to those of Stothers. In a similar manner, Stothers and Simon (1969) served as the source of initial trial values for our β Cephei models.

In all cases, it was found that rapid convergence occurred with: $\delta P_c = 10^{-6} P_c$, $\delta T_c = 10^{-6} T_c$, $\delta L = 10^{-6} L$, $\delta R = 10^{-6} R$.

The criterion for a good fit was that $\delta P_f \leq 10^{-12} P_f$, $\delta T_f \leq 10^{-12} T_f$, $\delta L_f \leq 10^{-12} L_f$, and $\delta r_f \leq 10^{-12} r_f$. This criterion was usually met within 8 iterations.

4.3 MASS ZONING SCHEMES

The manner in which each stellar model was divided into concentric shells of incremental mass varied within the framework of the five schemes outlined below. Note, that each of the schemes involves the zoning of the model into shells of equal mass increment from the center to some point well above the boundary of the convective core. This feature facilitated determination of the core luminosity distribution by the method outlined in section (3.4).

4.4 UNIFORM ZONING

The model is divided into shells of uniform mass from center to surface (i.e. $M_i - M_{i-1} = \Delta M$, where ΔM is constant). If we consider terms up through second order we find that the derivative at any point in the model may be expressed as (see APPENDIX D)

$$\left.\frac{df}{dM}\right|_{M=M_i} = \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{M_{i+\frac{1}{2}} - M_{i-\frac{1}{2}}} + \left.\frac{d^2f}{dM^2}\right|_{M_i} [M_{i+\frac{1}{2}} + M_{i-\frac{1}{2}} - 2M_i]$$

NOTE: that in the uniformly zoned model the expression in the brackets of the second term of the above equation disappears. Therefore one strong advantage of the uniform zoning scheme is that the approximation:

$$\left.\frac{df}{dM}\right|_{M_i} = \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{\Delta M}$$

is in fact good to second order, leading to second order spatial integration schemes.

A strong disadvantage of the uniform zoning scheme is that in order to get good resolution at a model's surface (where most physical quantities are changing rapidly) an inordinately large number of zones must be considered, slowing down calculation time.

4.5 PARABOLIC ZONING

The model is divided into shells of uniform mass from the center to some fitting point above the core boundary. The fitting point mass, M_f, is chosen to satisfy the following relation:

$$M_f = \frac{2i_f}{N + i_f} M \qquad (4.15)$$

where: i_f = number of mass zones up to fitting point

N = total number of zones model is to have

M = total stellar mass

Above the fitting point, the stellar model is divided into shells of varying mass in accordance with the following formula:

$$M_i = Ai^2 + Bi + C \qquad i_f \leq i \leq N \qquad (4.16)$$

where: $A = -M/(N^2 - i_f^2)$

$B = -2A \cdot N$

$C = M + A \cdot N^2$

Figure (4.1) shows this mass distribution.

NOTE: that $\frac{dM_i}{di}$ is continuous at the fitting point and

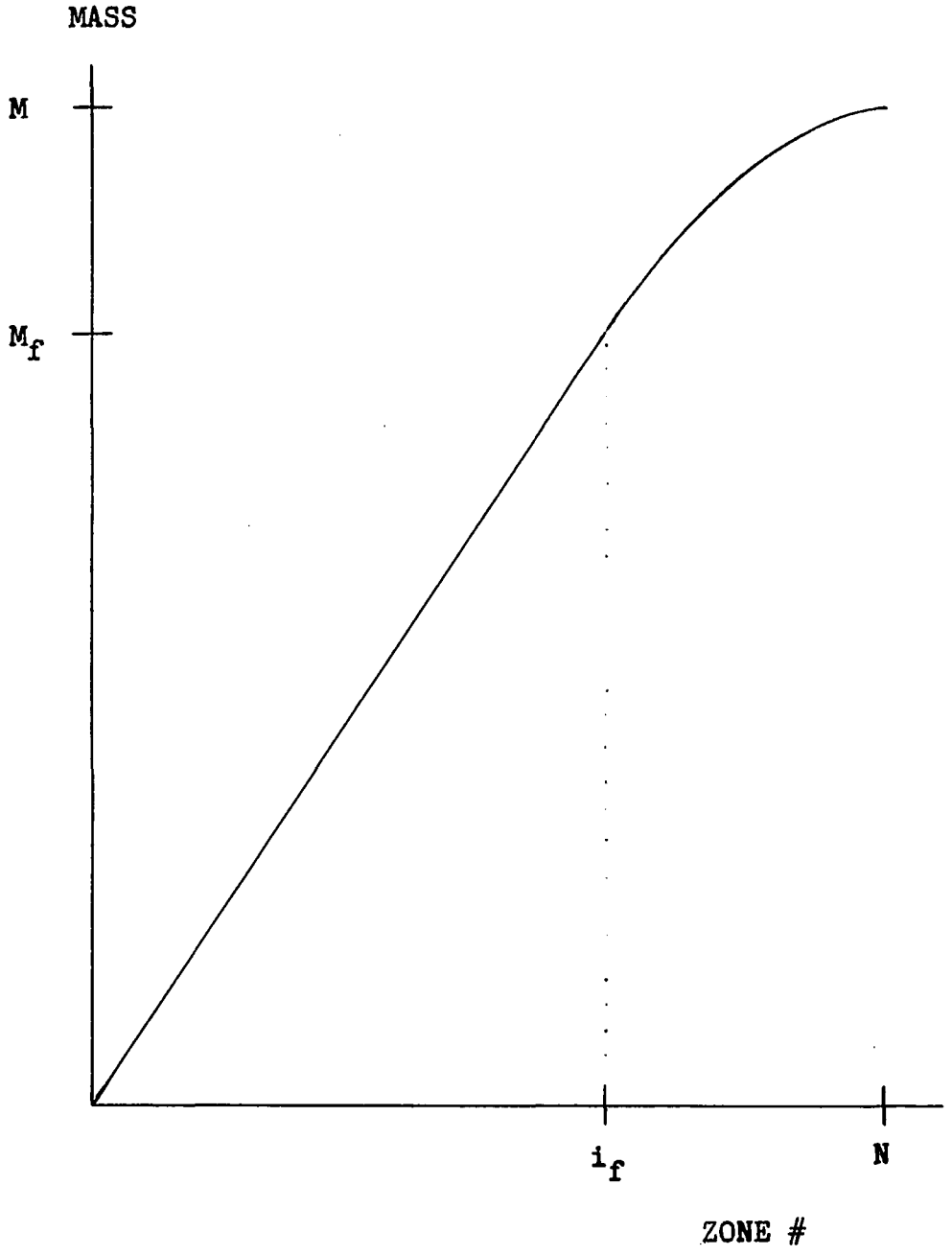
that $\frac{dM_i}{di} = 0$, at $M_i = M_N = M$.

This zoning scheme allows smaller increments of mass near the surface than the equivalent uniform zoning scheme, with only a moderate increase in the number of zones. The continuity of the derivative at the fitting point assures a minimal disruption due to changing mass content of shells. Since the mass content is not constant from shell to shell, spatial integration schemes employed in the main code are no longer equivalent to second order schemes.

In practice this zoning scheme was implemented in the following fashion: a value for i_f and an approximate value for M_f were chosen. On the basis of these values N was calculated from equation (4.15). Since an arbitrary choice of M_f does not guarantee an integer value for N, the result had to be truncated to the nearest integer. Using this truncated (integer) value of N in Equation (4.15) an exact value for M_f was calculated. The final value of M_f was always within a percent or two of the initial value specified.

FIGURE 4.1

PARABOLIC MASS ZONING SCHEME


MASS
M
M_f
i_f
N
ZONE #

Defining $\Delta M \equiv M_i - M_{i-1}$, for $i \leq i_f$, we find $\Delta M = M_f/i_f$. Now let us find the value of mass in the surface shell (i.e. $M - M_{N-1}$)

$$M - M_{N-1} = A[N^2 - (N-1)^2] + B[N - (N-1)]$$

$$= (2N - 1)\,A + B = \frac{M}{(N^2 - i_f^2)}$$

Because of the parabolic nature of the mass distribution, the mass shells inward from the surface to the fitting point may be calculated from the following relation:

$$M_{N-k} - M_{N-(k+1)} = (2k + 1)M/(N^2 - i_f^2)$$

for: $0 \leq k \leq N - i_f$

4.6 EXPONENTIAL ZONING

As in the Parabolic scheme the zoning is uniform in mass up to some fitting point M_f. Then the mass distribution satifies the following relation:

$$M_i = M_T - (M_T - M_f) \exp\left[\frac{-M_f}{n(M_T-M_f)} \left(\frac{i^n}{i_f^n} - 1\right)\right] \qquad (4.17)$$

$$i_f \leq i < \infty$$

where: $M_T = M_{(star)} + M_{Atmo}$

(See APPENDIX C for calculating M_{Atmo})

The above exponential distribution displays the following behavior:

The mass and rate of change of mass are continuous across the fitting point.

The mass goes to M_T as $i \longrightarrow \infty$

The rate of change of mass goes to 0 as $i \longrightarrow \infty$

The above scheme is implemented as follows: A value for i_f and an initial value for M_f are chosen. M_i is set equal to M_{star} and equation (4.17) is solved for i. The value of i is truncated to an integer

$i \rightarrow i_{surf} \cong N$. Now N is substituted back into equation (4.17), which is solved for the adjusted value of M_f.

Now Equation (4.17) may be solved for all integer values, $i_f \leq i \leq N$, giving the mass distribution from fitting point to surface. We may now choose any number of zones in the atmosphere N_{Atmo} with full confidence that our scheme adjusts zone sizes so as not to over-run M_T. The mass distribution in the atmosphere is then found by solving for M_i for $N \leq i \leq N + N_{Atmo}$.

The role of the parameter n is to adjust the steepness of the distribution (i.e. to control number of zones from fitting point to surface and zone size within the atmosphere). Values of n are allowed to take on positive integer values. As n is increased the number of atmospheric zones necessary to reach a certain optical depth decreases.

This zoning scheme enjoys all the advantages of the Parabolic scheme with the increased flexibility of being able to adjust the mass within the outer shells down to arbitrarily small values in a chosen number of zones, thus allowing for the inclusion of an atmosphere.

In practice a star zoned so as to include even only one zone in an atmosphere will be thermally

limited as opposed to Courant limited, thereby increasing markedly the time for calculating a pulsational period (See section 6.2). This scheme, as the Parabolic scheme, leads to a first order solution of all integrations in space.

4.7 CHRISTY TYPE ZONING

This scheme is an adaptation of the mass distribution used by R. Christy (1964) for investigating Cepheid variables. The star is uniformly divided into shells of equal mass up to some fitting point M_f above the convective core boundary, then each succeeding shell has a mass which is some constant fraction of the mass for the shell before.

$$(M_{i+1} - M_i) = B\cdot(M_i - M_{i-1}) \quad i_f \leq i \leq N \quad \text{where} \quad B \lesssim 1 \qquad (4.18)$$

The closer B is to unity, the less disruptive is the effect of mass zone size change on the difference scheme. However, we must take more zones and move the fitting point closer to the convective core boundary to achieve the same minimum zone size as B approaches unity. We are ultimately limited by the restriction that $M_f \geq M_{convec}$.

Assuming that the mass above the fitting point is distributed in an integer number of zones, $N_{env} = N - i_f$, the following relations must be obeyed:

$$M = M_f + \Delta M \sum_{n=1}^{N_{env}} B^n \tag{4.19}$$

where ΔM is the sub-fitting point zone size.

multiplying both sides by B

$$B \cdot M = B \cdot M_f - B\Delta M + \Delta M \sum_{n=1}^{N_{env}} B^n + B^{N_{env}+1} \Delta M \tag{4.20}$$

subtracting (4.19) from (4.20)

$$(B - 1)(M - M_f) = B(B^{N_{env}} - 1)\, \Delta M$$

$$\frac{B(B^{N_{env}} - 1)}{(B-1)} = \frac{M-M_f}{\Delta M} \tag{4.21}$$

In practice N_{env} is chosen and B is found then the mass is zoned in accordance with equation (4.18). The advantages of this scheme are similar to those of the Exponential scheme. The disadvantages are the same as that of the two previously described schemes.

4.8 SUBDIVIDED ZONING

The model is divided into a predetermined number of uniform shells as in the uniformly zoned scheme. The last shell is then subdivided into N_{sub} uniform subzones, etc. N_{sub}, for the equilibrium models, is always twice some odd number. This implies that the dynamic models will always have zones subdivided into an odd number of shells. (See Figure 4.2). The reason for this is to allow for a differencing scheme that results in second order space integrations. Consider evaluating the derivative of a quantity at a point within a mass shell. Then:

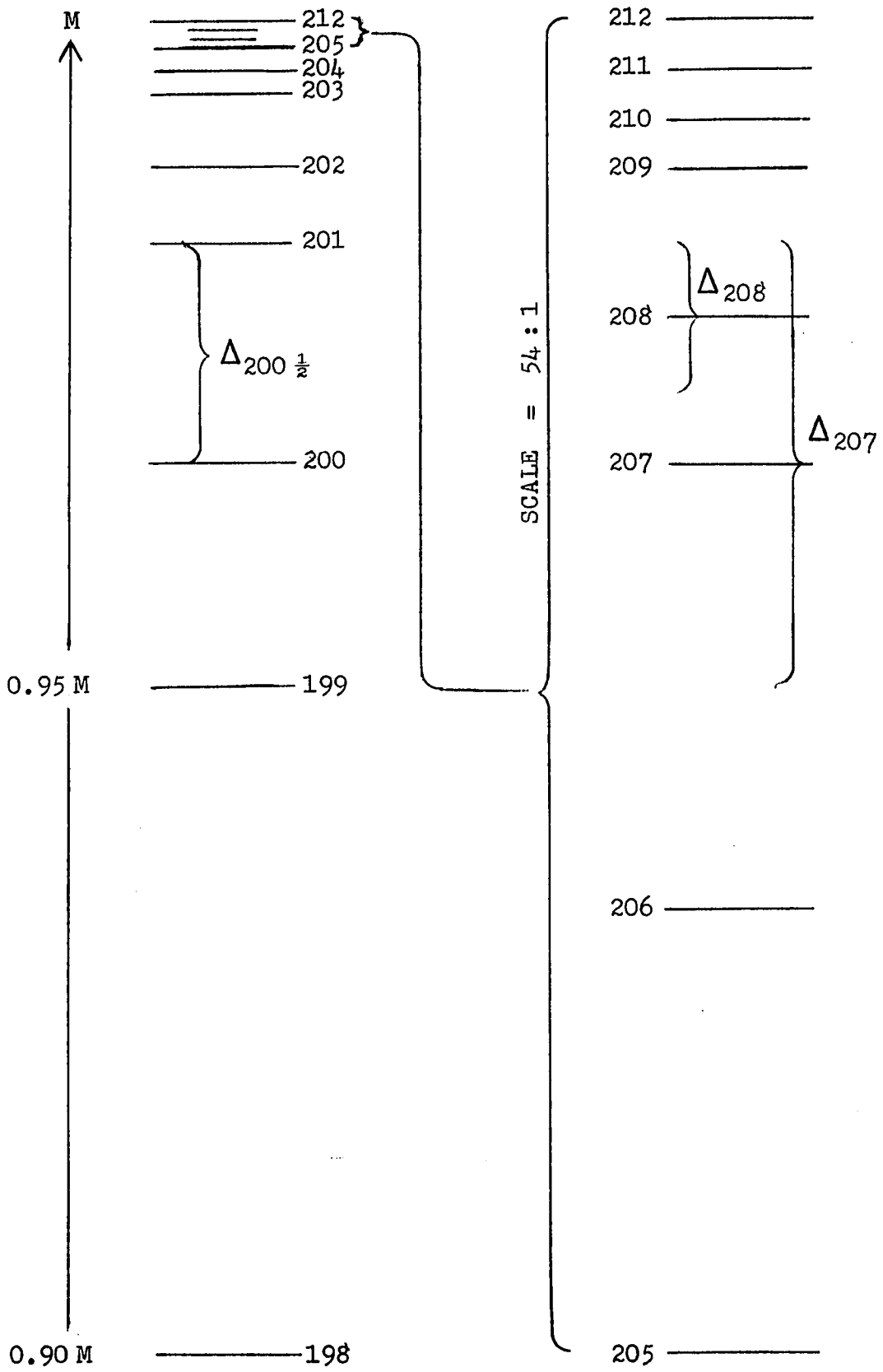
$$\left.\frac{dr}{dM}\right|_{M_{i+\frac{1}{2}}} \longrightarrow \frac{r_{i+1} - r_i}{M_{i+1} - M_i}$$

where the values of r and M are those on the shell boundaries and the above scheme guarantees centering of these derivatives in the dynamic models. On the other hand, when we consider derivatives at shell boundaries we run into the following two cases:

FIGURE 4.2

SUBDIVIDED MASS ZONING SCHEME

Numbers refer to zone boundary numbers from center. At the surface N = 212, in this case. Zone sizes are uniform up through zone 199, then what would be the last zone in a Uniform Zoning Scheme is successively divided in accordance with the text.


M
212
205
204
203
202
201
200
$\Delta_{200\frac{1}{2}}$
0.95 M
199
0.90 M
198
SCALE = 54 : 1
212
211
210
209
208
207
206
205
Δ_{208}
Δ_{207}

Case Number 1: Boundary considered is between shells of equal mass. The derivatives at this boundary are guaranteed to be centered with respect to mass

(i.e. $\left.\frac{dP}{dM}\right|_{M_i} \rightarrow \frac{P_{i+\frac{1}{2}} - P_{i-\frac{1}{2}}}{M_{i+\frac{1}{2}} - M_{i-\frac{1}{2}}}$ is a second order

approximation if $M_{i+\frac{1}{2}} - M_i = M_i - M_{i-\frac{1}{2}}$)

Case Number 2: Boundary considered is between shells of unequal mass (i.e. at a point where a subdivision occurs). the derivatives, if computed as above will not be centered. Therefore we propose to approximate these derivatives as follows:

$$\frac{dP}{dM_i} \rightarrow \frac{P_{j+\frac{1}{2}} - P_{i-\frac{1}{2}}}{M_{j+\frac{1}{2}} - M_{i-\frac{1}{2}}}$$

where: $j = i + (N_{sub} - 1)/2$

Note, that from the definition of N_{sub},

$M_{j+\frac{1}{2}} - M_i = M_i - M_{i-\frac{1}{2}}$ is guaranteed and therefore the

derivative as defined above will be mass centered (See Figure 4.2).

Whenever the subdivision zoning scheme is used the main code has been so modified as to take derivatives at boundaries between shells of unequal mass in the manner indicated above. The subdivision zoning scheme allows for second order space integrations as in the uniform zoning scheme while at the same time providing for decrease in the mass zone sizes in the outer regions of the model where better resolution is required.

CHAPTER 5

STELLAR DYNAMICS

5.1 THE DYNAMICAL EQUATIONS

The dynamic system may be expressed by the following generalization of the stellar structure equations (4.1) and (4.2).

$$\frac{\partial^2 r}{\partial t^2} = -\frac{GM}{r^2} - 4\pi r^2 \frac{\partial P}{\partial M} \tag{5.1}$$

$$\frac{\partial S}{\partial t} = \frac{1}{T}(\varepsilon - \frac{\partial L}{\partial M}) \tag{5.2}$$

where: $r = r(M,t)$, etc.

If P, T, ε, and L can be expressed in terms of r, S and M, we may in principle solve equations (5.1) and (5.2).

To this end let us consider the continuity equation

$$\rho = \frac{1}{4\pi r^2}\frac{\partial M}{\partial r} \tag{5.3}$$

Then from the expression for entropy (equation 3.4), we may find T for a given S, ρ.

Knowing T, it is a simple matter to get P from the equation of state and ε from the energy generation formulas of section (3.2). In the radiative envelope, L may be gotten from the radiative transport equation. In the convective region we get the distribution of L by the spatial integration indicated in section (3.4).

There are two factors that prevent an attempt of an analytic substitution for P, T, $\mathcal{E}$ and L into equations (5.1) and (5.2). Firstly, the transcendental behavior of the entropy equation (3.4) makes it impossible to solve for T explicitly. Secondly, in the convective region L is obtained only after a spatial integration of a second order equation and therefore can only be expressed in integral form, which could not preserve the simple differential nature of equation (5.2).

A summary of the system of equations to be solved is presented in Table (5.1). Equation (5.1) has been converted into two first order equations by introducing the velocity as an additional variable. This serves a two fold purpose:

Limiting our considerations to a set of first order differential equations allows us to employ a powerful Runge-Kutta scheme for the numerical solution. The explicit appearance of velocity allows us to conveniently perturb our static models by imposing an initial velocity distribution.

In order to solve the preceding set of equations we must specify initial values of the primary variables r, V, and S, as well as boundary conditions. The considerations that go into choosing central and outer boundary conditions are discussed in considerable detail in section (3.5). It is assumed that the pulsating model

TABLE 5.1 STELLAR DYNAMICS

BASIC EQUATIONS	$\frac{\partial r}{\partial t} = v$ $\frac{\partial v}{\partial t} = -\frac{GM}{r^2} - 4\pi r^2 \frac{\partial P}{\partial M}$ MOMENTUM CONSERVATION $\frac{\partial S}{\partial t} = \frac{1}{T}(\varepsilon - \frac{\partial L}{\partial M})$ ENERGY CONSERVATION
CONTINUITY CONDITION	$\rho = \frac{3}{4\pi} \frac{dM}{d(r^3)}$
TEMPERATURE CONDITION	$S = \frac{1}{\mu} \frac{k}{H} \ln(\frac{T^{3/2}}{\rho}) + \frac{4}{3} \frac{aT^3}{\rho} + K$ where: $\frac{1}{\mu} = \frac{1}{2}(1+3X+\frac{Y}{2})$
PRESSURE CONDITION	$P = \frac{1}{\mu} \frac{k}{H} \rho T + \frac{a}{3} T^4$ EQUATION OF STATE
ENERGY CONDITION	$\varepsilon = 7.9 \cdot 10^{27} f_{14,1} g_{14,1} \rho X_{14}^X T_{14}^x T_6^{-2/3}$ $\exp(-152.31 T_6^{-1/3})$ where: $f_{14,1} = 1 + 1.75\rho^{\frac{1}{2}}/T_6^{3/2}$ $g_{14,1} = 1 + 10^{-3}(3T_6^{-1/3} - 4T_6^{2/3})$
LUMINOSITY CONDITIONS	$L = -(4\pi r^2)^2 \frac{ac}{3} \frac{dT^4}{dM}$ (RAD) $\frac{d^2L}{dM^2} - (\frac{1}{T}\frac{dT}{dM})\frac{dL}{dM} + (\frac{\varepsilon}{T}\frac{dT}{dM} - \frac{d\varepsilon}{dM}) = 0$ (CON) where: $\kappa = \kappa(\rho, T)$ or $\frac{4\pi}{3c}[\frac{e^2}{mc^2}]^2 (1 + X)$

will pass through an equilibrium state that may be approximated by the static model distributions. Therefore, the initial values for the distributions r(M,0) and S(M,0) are taken as those of the static models. Several schemes are used for determining initial velocity distributions and are discussed in the following section.

SECTION 5.2 - INITIAL VELOCITY DISTRIBUTION

The simplest initial velocity distribution to impose on a dynamic model would be $v(M,0) = 0$ for all M. At first it might appear that such a velocity distribution would result in a static model. This is not the case. The equilibrium model solutions (See section **4.1**) do not represent static solutions of the dynamic model equation set. Parameters, such as ρ, are calculated in a different manner for the dynamic models than for the equilibrium models. Hence, the zero velocity initial condition results in a starting model that is very slightly perturbed in a pseudo-random fashion. The major drawback of employing such a starting model is that the amplitude of pulsations will take a great many periods to grow (Schwarzchild and Harm, 1959 have demonstrated that the e-folding time for a massive star is about 10^6 periods). In addition numerical devices introduced to facilitate computation produce effects comparable to the perturbation amplitudes.

A useful method for obtaining a starting velocity distribution is to scale up the **advanced velocity** distribution from a zero starting velocity model as it is passing through an equilibrium configuration. This scheme was employed in a majority of the models we considered.

A third method of obtaining a reasonable

initial velocity distribution is to allow $v(M,0)/r(M,0)$ to be proportional to $\frac{\delta r}{r}$, where $\frac{\delta r}{r}$ represents the radius amplitude computed for models using linearized pulsation theory (See APPENDIX A). In the models we considered, $\frac{\delta r}{r}$ was supplied by N. Simon (1969) and corrected for departures from adiabaticity near the surface. It is apparent that choice of such a starting velocity distribution should optimize stimulation of the fundamental mode of pulsation. In practice this choice of starting velocity does not lead to results substantially different from the scaled up velocity method, previously discussed, providing that the scaling factors are adjusted so that the initial surface velocities are comparable.

Still another choice of starting velocity distribution is one that would reflect a homologous perturbation, i.e. $v(M,0) \propto r(M,0)$. This simplistic distribution was used in most of the later models, after it was discovered that it led to results not substantially different from those obtained using the two methods outlined above.

It seems that all of the initial velocity distributions discussed will produce nearly identical results provided they are chosen so that the surface velocities are the same. This feature could have been anticipated, in part, by considering that δr for the linearized models is nearly homologous (i.e. $\frac{\delta r}{r} \sim$ const).

Since the zero starting velocity models are all very small amplitude models, they too will give rise to scaled up velocities that are approximately homologous.

CHAPTER 6

PULSATIONAL MODELS

6.1 THE NUMERICAL SCHEME

To solve the equations of section 5.1 numerically, we must subdivide the model into spatial zones based on one of the mass zoning schemes outlined in section 4.3 . We will use integer subscripts to refer to the value of variables at zone boundaries. The subscript i is appended to a variable evaluated at the outer boundary of the i^{th} zone. The inner boundary of the i^{th} zone is of course the outer boundary of the $(i-1)^{th}$ zone and variables evaluated at this point will be appended with the subscript (i-1). The following variables are generally evaluated at zone boundaries: M, r, v and L. The thermodynamic variables represent the mass averaged values for each zone and are presumed to be evaluated at some mass point within each zone, designated by $M_{i-\frac{1}{2}}$. The thermodynamic variables are ρ, T, P, S, $\mathcal{E}$ and κ. Note that variables evaluated within zones are subscripted with half integers; $i-\frac{1}{2}$ corresponding to the i^{th} zone.

The numerical approximation corresponding to the 3 basic dynamic equations follows:

$$\frac{dr_i}{dt} = v_i \tag{6.1}$$

$$\frac{dv_i}{dt} = -\frac{GM_i}{r_i^2} - 4\pi r_i^2 \frac{P_{i+\frac{1}{2}}-P_{i-\frac{1}{2}}}{\Delta M_i} \tag{6.2}$$

$$\frac{dS_{i+\frac{1}{2}}}{dt} = \frac{1}{T_{i+\frac{1}{2}}} \left[\varepsilon_{i+\frac{1}{2}} - \frac{L_{i+1}-L_i}{\Delta M_{i+\frac{1}{2}}}\right] \tag{6.3}$$

$$\text{where:} \quad \Delta M_i \equiv M_{i+\frac{1}{2}} - M_{i-\frac{1}{2}}$$

$$\Delta M_{i+\frac{1}{2}} \equiv M_{i+1} - M_i \tag{6.4}$$

We now have a set of 3N simultaneous first order ordinary differential equations, where N represents the total number of zones into which the model is divided. This situation lends itself to a method of solution similar to that employed in solving the equilibrium models. We may use the same fourth order Runge-Kutta scheme outlined in section 4.1. Of course, instead of dealing with 4 equations we must contend with 3N equations, and our integration is carried forward in time as opposed to space. The necessary supplemental conditions to equations (6.1-6.3) may be presented as follows:

$$\rho_{i+\frac{1}{2}} = \frac{3}{4\pi} \frac{\Delta M_{i+\frac{1}{2}}}{r_{i+1}^3 - r_i^3} \tag{6.5}$$

$$S_{i+\frac{1}{2}} = S(\rho_{i+\frac{1}{2}}, T_{i+\frac{1}{2}}) \Rightarrow T_{i+\frac{1}{2}} \tag{6.6}$$

$$P_{i+\frac{1}{2}} = \frac{k}{\mu H} \rho_{i+\frac{1}{2}} T_{i+\frac{1}{2}} + \frac{a}{3} T_{i+\frac{1}{2}}^4 \tag{6.7}$$

$$\varepsilon_{i+\frac{1}{2}} = \varepsilon(\rho_{i+\frac{1}{2}}, T_{i+\frac{1}{2}}) \tag{6.8}$$

$$\kappa_i = \frac{\kappa(\rho_{i+\frac{1}{2}}, T_{i+\frac{1}{2}}) + \kappa(\rho_{i-\frac{1}{2}}, T_{i-\frac{1}{2}})}{2} \tag{6.9}$$

$$L_i = -(4\pi R_i^2)^2 \frac{ac}{3\kappa_i} \frac{T_{i+\frac{1}{2}}^4 - T_{i-\frac{1}{2}}^4}{\Delta M_i} \tag{6.10}$$

where: $i \geq N_{con}$ (convec. core boundary)

L_i for $i < N_{con}$ comes from equation (3.22) solved between the boundaries $L_o = 0$ and $L_{N_{con}}$ from equation (6.10) (6.11)

A casual perusal of equations (6.1) through (6.10) indicates that difficulties arise when trying to determine the value of some of the variables at the boundaries ($i = 0$ and $i = N$). This difficulty is circumvented by applying the boundary conditions discussed in section (3.5). At the star's center, $r_o = v_o = L_o = 0$, and it is unnecessary to calculate any of the other variables in order to evolve the model in time. The surface boundary poses a more complex problem. In order to calculate v_N using equation (6.2), we must specify $P_{N+\frac{1}{2}}$ and M_N. Since this amounts to evaluating P and M at points physically not in the star, we resort to the following technique which allows us to maintain the same indexing scheme right up to the stellar surface. First, we choose ΔM_N in some convenient fashion such as $\Delta M_N \equiv M_N - M_{N-\frac{1}{2}}$ or $\Delta M_N = 2(M_N - M_{N-\frac{1}{2}})$. Then we specify

$P_{N+\frac{1}{2}}$ so that

$$\frac{P_{N+\frac{1}{2}} - P_{N-\frac{1}{2}}}{\Delta M_N} = \left.\frac{dP}{dM}\right|_{M_N} \tag{6.12}$$

where the right side of equation (6.12) is evaluated in one of the ways indicated in section (3.5). With the help of procedures outlined in APPENDIX D we may calculate $P_{N+\frac{1}{2}}$ from physically tenable variables such as $P_{N-\frac{1}{2}}$ and $T_{N-\frac{1}{2}}$. We may find L_N in a similar fashion by substituting

$$\frac{T^4_{N+\frac{1}{2}} - T^4_{N-\frac{1}{2}}}{\pi_N \Delta M_N} = \frac{1}{\kappa}\left.\frac{dT^4}{dM}\right|_{M_N} \tag{6.13}$$

or equivalently, by calculating the effective temperature when using photospheric boundary conditions.

Equations (6.1 - 6.13), together with the initial distribution $r_i(0)$ and $S_i(0)$ taken from the equilibrium models and $v_i(0)$ imposed as indicated in section (5.2), completely defines the numerical problem.

In actual computation, additional factors had to be taken into account. Realistic initial values for ρ and T had to be supplied in order to guarantee rapid convergence for some of the code's routines (e.g. the Newton-Raphson scheme employed in finding T from the entropy). Expression (6.4) had to be modified so that ΔM_i corresponds to the correct mass interval when using

the subdivided mass zoning scheme of section (4.8). In addition, when this zoning scheme was employed, the expressions involving spatial derivatives had to be modified so that variables were evaluated at the correct points as indicated in figure (4.2). Yet another complication was introduced when solving models of non-homogenous chemical composition. The variables defined by equations (6.6) through (6.9) had to be treated as functions of chemical composition. Some of the other "nitty-gritty" considerations that were involved in actual computation will be discussed in sections (6.2) and (6.3).

At this point it is useful to make some general observations concerning the numerical integration scheme outlined in this chapter. It is a completely "explicit" scheme. That is, all equations are integrated forward in time, and the new values of r, v, and S are determined solely on the basis of the value of variables evaluated during the previous time step. Thus we may consider the problem as being redefined each time step so that we have the integration of the same set of differential equations with new initial conditions. Obviously, in terms of computer usage, this expedites restarting the code so that calculations for a particular model may be readily segmented. It also has the further advantage of minimizing the propagation of certain types of error.

6.2 THE TIME STEPSIZE

In performing the integrations involved in constructing the equilibrium models of Chapter 4, we imposed no restriction on the mass integration stepsize. In general, if too large a stepsize is chosen, the only consequence is that the solution to the finite difference scheme is not an accurate representation of the solution to the corresponding differential equations. However when dealing with the dynamic models, we are concerned with two dimensional equations, the space and time stepsizes in the numerical scheme defining a two dimensional mesh. When using an explicit scheme, it is necessary to impose conditions on the stepsize to guarantee stability of the numeric scheme.

Equations (6.1) and (6.2) are numerical representations of a simplified hydrodynamic equation with the gravitational potential included. Numerical representations of the hydrodynamic equation have been extensively studied. To guarantee stability of explicit representations, it is sufficient to employ the well know "Courant" condition (Courant, Friedrichs and Lewy, 1928; Courant, Issacson and Rees, 1952).

In essence, the Courant condition restricts the time stepsize so that a disturbance propagating at the local sound speed is unable to traverse a spatial zone

within one time step. For the scheme described in section (6.2), this condition may be written as follows (see Richtmyer and Morton, 1967, p. 298):

$$\frac{c\Delta t}{r_{i+1}^{j} - r_i^{j}} \leq 1 \qquad \text{for all } i,\ j \tag{6.14}$$

where the superscript denotes that r_i and r_{i+1} are being evaluated at the j^{th} time step. We may put this in a form more suitable for our Lagrangian scheme by noting that, $\Delta M \sim 4\pi r^2 \rho \Delta r$, leading to

$$4\pi (r_i^j)^2 \rho_i^j \frac{c_i^j \Delta t^j}{\Delta M_i^j} < 1 \quad \text{for all } i,\ j$$

We are at liberty to revaluate the time stepsize at the end of each time step. Therefore we have appended the superscript j to Δt, indicating that Δt^j is the new stepsize determined at time step j. We may now drop the superscripts entirely, with the understanding that the procedures derived below are to be applied at each time step to determine the stepsize to the next time step.

$$\Delta t \lesssim \frac{\Delta M_i}{4\pi r_i^2} \frac{1}{\rho_i c} \quad \text{for all } i \tag{6.15}$$

but $$c \equiv \sqrt{\frac{dP}{d\rho}} = \sqrt{\frac{(\Gamma_1)_i P_i}{\rho_i}}$$

$$\Delta t \lesssim \frac{\Delta M_i}{4\pi r_i^2} [(\Gamma_1)_i P_i \rho_i]^{-\frac{1}{2}} \tag{6.16}$$

where the adiabatic exponent Γ_1 for an ideal gas plus radiation is shown by Chandrasekhar (1957, p. 57) to be

$$\Gamma_1 = \beta + \frac{(4-3\beta)^2 (\gamma-1)}{\beta + 12(\gamma-1)(1-\beta)}$$

which in our case reduces to

$$\Gamma_1 = \beta + \frac{\frac{2}{3}(4-3\beta^2)}{\beta + 8(1-\beta)} \tag{6.17}$$

To insure stability of the hydrodynamic equation, the righthand side of equation (6.16) is evaluated for each mass zone and the minimum value is retained. Δt is then chosen as some fraction (usually 0.8) of this value.

We must now find an analogous condition to insure the stability of the thermal equation (6.3). We shall refer to the restriction imposed on the time step by requiring stability of equation (6.3) as the thermal stepsize, to differentiate it from the Courant stepsize discussed in the preceding paragraph. If we examine equation

(6.3) in its original differential form (equation 5.2) we see that it may be rewritten as

$$\frac{\partial T}{\partial t} = \sigma \frac{\partial^2 T}{\partial r^2} + a\frac{\partial T}{\partial r} + b \cdot T$$

where we have expressed L and $\mathcal{E}$ in terms of T and its mass derivative. Of course, σ, a , and b will be functions of ρ and T. On pages 195-198, Richtmyer and Morton (1957) discuss equations of this form. We come to the conclusion that

$$\frac{\sigma \Delta t}{(\Delta r)^2} \lesssim 1/2 \qquad (6.18)$$

guarantees stability in all situations. Detailed stability analysis for the above equation is quite involved, and stability criteria depend on the actual values of the diffusion equation coefficients in a complex way. It is readily seen that:

$$\sigma = \frac{K}{c_v} \qquad (6.19)$$

where K is the effective thermal conductivity (i.e. the energy transport coefficient) and c_v is just $\left(\frac{\partial U}{\partial T}\right)_\rho$.

The radiative flux may be written as

$$\frac{L}{4\pi r^2} = -K\frac{dT}{dr}$$

referring to equation (3.19) we find

$$K = \frac{4ac}{3\kappa\rho}T^3 \qquad (6.20)$$

For an ideal gas plus radiation,

$$U = \frac{3}{2}\frac{k\rho}{\mu H}T + aT^4$$

where: U is the internal energy per <u>unit volume</u>

therefore:

$$c_v = \frac{3}{2}\frac{k\rho}{\mu H} + 4aT^3 \qquad (6.21)$$

finally leading to

$$\sigma = \frac{4acT^3/3\kappa\rho}{\frac{3}{2}\frac{k\rho}{\mu H} + 4aT^3} \qquad (6.22)$$

The thermal time condition may now be stated as

$$\Delta t \lesssim \frac{1}{2}\frac{(\Delta M_i)^2}{(4\pi r_i^2)^2 \rho_i^2 \sigma_i} \qquad (6.23)$$

where σ_i is defined by equation (6.22)

Once again, in practice, the righthand side of equation (6.23) is evaluated for each mass zone and the minimum value multiplied by a fraction is chosen as the thermally limited time stepsize.

Finally, the Courant stepsize determined through equation (6.16) is compared to the thermal stepsize determined through equation (6.23). The time stepsize used to advance the model is determined as the minimum of these two.

Quite fortunately most of our models are Courant limited. When the models are thermally limited the time stepsize goes as the square of the mass zone size, and any tightening in zoning causes an amplified reduction in stepsize, thereby increasing the computation time for a pulsational period substantially. The optimum zoning scheme would set the mass zone sizes so that the equations would be on the verge of being thermally limited. In terms of a Courant limited mesh, the Christy mass zoning scheme has the interesting property of decreasing the mass zone size roughly at the same rate as the sound velocity decreases. This causes the Courant stepsize determined for each zone to be roughly constant throughout the outer regions of the models zoned in this fashion. The Christy zoning scheme is therefore very efficient in terms of packing into a model a maximum number of zones with a minimal increase in computation time.

6.3 ARTIFICIAL VISCOSITY

When dealing with non-viscous hydrodynamics, it is possible to generate shock fronts that cause discontinuities or near discontinuities in some of the dynamic variables. The occurrence of these fronts represent perfectly valid solutions to differential equation (5.1). However, the development of such features in the finite difference equations (6.1) and (6.2) cause trouble, such as the crossing of mass zones and other destabilizing effects. Ultimately, even if the numeric scheme supports the development and propagation of near discontinuities, we will run into trouble during computation. If we are required to take spatial derivatives of one of the variables such as pressure, we may find the slope of the front exceeds the largest number representable on the computer.

Another problem encountered when replacing equation (5.1) by equations (6.1) and (6.2) is that there exist solutions to the numeric equations that do not correspond to solutions of the original differential equation (i.e. non-physical solutions). These solutions usually consist of fluctuations which span space and time intervals on the order of the mesh size. In some instances they are referred to as "numeric

shocks".

It is desirable to introduce some device that will spread out the real shocks without overtly changing the physics of the problem, while at the same time damping out the numerical shocks. Such a device has been extensively investigated by Lax and Wendroff (1960), following up on a study by Lax and Richtmyer (1956).

The Lax-Wendroff form of artificial viscosity has recently been used in a number of astrophysical investigations involving solution of the hydrodynamical equations (Arnett, 1966; Cocke and Cohen, 1969; Colgate and White, 1966; Christy, 1964). The exact form of Lax-Wendroff artificial viscosity we employed is that given by Richtmyer and Morton (1967, p. 318).

The artificial viscosity enters into equation (6.2) in the form of what one might loosely interpret as a turbulent pressure term to be added to the total pressure such that,

$$P^{j}_{i+\frac{1}{2}} \longrightarrow P^{j}_{i+\frac{1}{2}} + Q^{j}_{i+\frac{1}{2}} \tag{6.24}$$

$$\text{where: } Q^{j}_{i} = \begin{cases} \dfrac{C_q (v^{j}_{i+1} - v^{j}_{i})^2}{(1/\rho^{j-1}_{i+\frac{1}{2}} + 1/\rho^{j}_{i+\frac{1}{2}})} & , \quad v_{i+1} < v_i \\ 0 & , \quad v_{i+1} \geq 0 \end{cases} \tag{6.25}$$

The viscosity coefficient, C_q, determines the number of zones a shock front will spread over. It also determines how rapidly the numeric shocks are dissipated. To minimize its effect on the physical meaning of the solution, it is desirable to keep C_q relatively small.

It is difficult to establish a good compromise value of C_q a priori. Many of the more massive models were run with varying values of C_q in order to determine an optimum value. A survey of values used is given in the next chapter.

In order for equations (6.2) and (6.3) to retain their identity as conservation laws after equation (6.2) has been altered by condition (6.24), we must alter (6.3) to take into account the thermal effect of viscous turbulence that might generate a pressure gradient of the form $\frac{dQ}{dM}$. We will call the term we must add to the thermal equation (6.3) the thermal viscous term. The thermal viscous term must take the form:

$$\left(\frac{dS}{dt}\right)_Q = -\frac{Q}{T}\frac{d\left(\frac{1}{\rho}\right)}{dt} \qquad (6.26)$$

giving rise to the following term which may be added to equation (6.3) :

$$\frac{dS^{j}_{i+\frac{1}{2}}}{dt} \longrightarrow \frac{dS^{j}_{i+\frac{1}{2}}}{dt} - \frac{Q^{j}_{i+\frac{1}{2}}}{T^{j}_{i+\frac{1}{2}}} \frac{(1/\rho^{j}_{i+\frac{1}{2}} - 1/\rho^{j-1}_{i+\frac{1}{2}})}{\Delta t^{j-1}} \qquad (6.27a)$$

Although equation (6.27a) was used in constructing several models, most of the models which included the thermal viscous term used it in a slightly modified form. In order to reference the change in entropy to variables evaluated only during the current time step we find:

$$\frac{1}{\rho} = \frac{d}{dM}(\tfrac{4}{3}\pi r^3)$$

leads to

$$\frac{d}{dt}(\frac{1}{\rho}) = 4\pi \frac{d}{dM}(r^2 v)$$

allowing us to state the thermal viscous term as

$$\frac{dS_{i+\frac{1}{2}}}{dt} \longrightarrow \frac{dS_{i+\frac{1}{2}}}{dt} - 4\pi \frac{Q_{i+\frac{1}{2}}}{T_{i+\frac{1}{2}}} \frac{(r^2_{i+1} v_{i+1} - r^2_i v_i)}{\Delta M_{i+\frac{1}{2}}} \qquad (6.27b)$$

Examination of equation (6.25) reveals

$$Q \neq 0 \Rightarrow \frac{dv}{dM} < 0 \quad \text{or} \quad \frac{d}{dt}(\frac{dr}{dM}) < 0$$

In most instances when $\frac{d}{dt}(\frac{dr}{dM}) < 0$ we find $\frac{d}{dt}(\frac{1}{\rho}) < 0$. Referring to equation (6.26), we see that when the thermal viscosity contributes to the entropy (i.e. when Q is non-zero), it usually does so in such a way as to raise the entropy. The effect of a passing shock front is to also increase the entropy. We may therefore think of one of the roles of the artificial viscosity as spreading the shocks while (if the thermal viscous term is included in the analysis) thermalizing the shocks' energy in such a way that the energy balance is not grossly affected by the spreading.

When the thermal viscous term was incorporated into the computer code it was found that a value for C_q of 5 worked well, spreading the shocks over three or four zones, independent of shock strength.

6.4 RESULTS OF CALCULATION

The models constructed by the methods outlined in the previous sections, may be divided into two major categories. First we have the Zero Age Main Sequence models of homogeneous chemical composition. These models are of mass 30 $M_\odot$, 100 $M_\odot$ and 200 $M_\odot$. In all cases the abundances are chosen as X = 0.70, Y = 0.27, X_{CNO} = Z/2 = 0.015. Models in this category will be referred to simply as the "massive star" models. Next we have the models of inhomogeneous composition with helium enriched envelopes. All of these models are of Mass 15 $M_\odot$. Up to mass fraction 0.66 M, their composition is identical to that of the massive star models. Above 0.66 M the abundances are X = 0.35, Y = 0.62, X_{CNO} = 0.015. Since this mass and composition corresponds to a model of a β Cephei star proposed by Stothers and Simon (1969a), the models in this category will be referred to as the "β Cephei star" models.

The massive star models played a major role in the development, perfection, and testing of the computer codes. Features, such as the pulsational period and small perturbation damping rates, were well established for these models from the linearized theory and served as a check on the validity of the dynamic scheme.

Using the massive star models to find the

optimum configuration for running the code was a complex and tedious task. Each of the five mass zoning schemes described in chapter 4 was employed. Furthermore, models of varying minimum zone size were constructed within the context of each zoning scheme; the exponential zoning scheme being used to construct models with and without atmospheres. Some of the above models used an electron scattering opacity while others used the Christy-Stothers opacity (see section 3.3). In addition, each of the three sets of boundary conditions enumerated in section (3.5) had to be utilized in the construction of these models. To compound the complexity, the models were subject to each of the four types of initial velocity distributions outlined in section (5.2), runs being made with several values of velocity scaling factors for each class of distribution. Finally, the manner in which artificial viscosity affects the calculations was studied by running the models with several different values of C_q, the viscous coefficient. Some of the models were run with a varying value of C_q. None of the massive star models included a viscous term in the thermal equation.

The attributes and effects of the various zoning schemes, opacity laws, boundary conditions, initial velocity distributions, and forms of the artificial viscosity are discussed in the sections dealing with these options. However it is useful to point out the

following general features. Normally, the time stepsize for a model was Courant limited at some point well within the convective core. Inclusion of a stellar atmosphere in a model caused the time step to be thermally limited (see section 6.3). For the models considered, the atmospheric mass is about 10^{-10} of the total mass. Except for the very surface, the models were not strongly dependent on choice of boundary conditions. Nor were they greatly dependent on the form of the initial velocity distribution. They were, however, quite sensitive to the size of the velocity perturbation. If the artificial viscosity was too small or too large the models were destabilized. A value for C_q of 200 ± a half magnitude seemed to be the most effective range for the massive star models.

More than thirty massive star models were constructed. Most of these were run only for a short time, with the end in mind, of observing the effect of varying the computational parameters. Several common features of some physical significance were noted for these models. All the low initial velocity models seemed to be relaxing toward sinusiodal light and velocity curves, in agreement with linear theory. Larger initial velocity perturbations produced structured light curves which generally consisted of a major peak with one or two side peaks. Corresponding to these light curves

were surface velocity plots indicating a single main peak with the presence of a minor peak, usually trailing.

The β Cephei models, which were run after the development of most of the computational techniques, represent the main thrust of this dissertation. In all, sixteen models were run. Their characteristics are summarized in table (6.1) and figures (6.1) through (6.30). The first column of table (6.1) identifies the model; gives general information about the run; and where pertinent, indicates the radius amplitude of the pulsation. The second column indicates features of the code employed such as; the type of boundary conditions employed, what form the artificial viscosity takes if it is employed in the thermal term, and the type of initial velocity distribution imposed along with the value of surface velocity it produces. Column three indicates the type of mass zoning used (as explained in Chapter 4), the total number of zones, and the mass content of the smallest zone (invariably, the surface zone). The fourth column indicates the value of C_q, the artificial viscosity coefficient; and if C_q is changed during a run, the times of these changes. Columns five and six, which appear on the second page of the table, indicate schematically the shapes of the light and velocity curves respectively. They contain information as to what

interval in the time development the curves represent, information about the period, and information concerning the changes in amplitude and average value. Units of time are either seconds or number of pulsational periods from inception of the model. The last column contains information about the subsurface luminosity, as well as radius amplitude curves and other noteworthy information. The standard subsurface luminosity and radius are generally considered to be those at ten zones from the surface.

All the models utilize opacities as determined by the Stothers-Christy opacity formula of section (3.3). Furthermore, it has been assumed that the sole mode of energy transport in the envelope region above the convective core is by radiative diffusion. Stothers and Simon (1969) indicate that such an assumption may give rise to a small region in the vicinity of the composition discontinuity where the temperature gradient is slightly superadiabatic. In order to formally satisfy the condition for convective neutrality, it would be necessary to introduce a thin semiconvective zone (Schwarzchild and Harm, 1958). However, such a refinement would scarcely affect the energy balance near the surface, and is totally unmerited in light of the crude assumption of discontinuity in composition.

Before evaluating our results, it is useful to comment on those features of each of the β Cephei models which pertain to that model's credibility.

Model # 1: Exclusion of the thermal viscous term and incorporation of Eddington boundary conditions (section 3.5) employing a surface opacity derived by averaging the opacity one zone in with electron scattering opacity, leads us to categorize the "physics" used in synthesizing this model as only "moderately" realistic. The frequent changes in C_q make this model useful for testing the effects of artificial viscosity, but make it difficult to establish reliable trends. We will not consider model 1 in our analysis.

Model # 2: Physics and zoning are similar to that of model 1, but a thermal viscous term of the form indicated in table (6.1) is included. From the viewpoint of the differencing scheme, this is not the preferred form for this term. C_q is reduced to 5 during course of run. The only definitive observation to be made is that at low C_q, the surface seems to be expanding while undergoing pulsations.

Model # 3: Physics is the same as for model 2, but the surface zoning is much cruder ($\sim 10^4 \times$ larger). We will not consider model 3 in our analysis since its zoning makes sinusoidal surface response highly unreliable.

Model # 4: A machine error occurred early in this

model's development. We will disregard this model.

Model # 5: (Rerun of Model # 4) Physics is similar to that of model 2, but the surface zone size is down by an order of magnitude. C_q goes from high value of 20 down through 5 to value of 1. Results are to be accepted with caution because of inadequate physics coupled to moderate zone size.

Model # 6: Inclusion of the preferred form of thermal viscous term and use of photospheric boundary conditions, leads us to categorize the physics used in synthesizing this model as "good" (i.e. more realistic than in previous models). The surface zone size is an order of magnitude smaller than that of model 1. Model 6 was run for less than 4 pulsational periods, because of misinterpretation of the results at the time of computation. The results of this model are highly reliable.

Models # 7 through # 10: All of these models contain the same moderately realistic physics and utilize the same mass zoning as model 1. They all use a value of 200 for C_q. This turns out to be optimum when there is no thermal viscous term. The models differ from each other with respect to their initial velocity distribution. The fairly small surface zone size makes these models fairly reliable indicators of what good physics models would be like if the shock energy could be

radiated away.

Model # 11: This model has the same good physics and zoning as model 6. C_q is at its optimum value of 5 (see section 6.3). Therefore, the results of this calculation should be highly reliable.

Model # 12: This model is identical to model 11 except for initial velocity distribution, and its results are therefore highly reliable.

Model # 13: This model is very similar to model 12, except for the exclusion of the artificial viscosity term in the thermal equation. This makes both the physics and zoning somewhat superior to that of models 7 through 10. This model should be a quite reliable indicator of the behavior of the proposed β Cephei system with the shock energy totally dissipated.

Model # 14: This model runs less than 3 periods because of some non-physical difficulties with one of the computational schemes. It is not used in the analysis of the results.

Model # 15: Here we employ the same good physics as that of models 6, 12 and 13. The surface zone size is a full order of magnitude smaller than that of these models. The results of this model are extremely reliable.

Model # 16: This model is identical to model 15 except for initial velocity distribution and its results are therefore extremely reliable.

We may divide the usable models into two classes: those which include a thermal viscous term, and those which do not. In the first category we find the highly reliable models 6, 11, 12, 15 and 16; as well as the questionably reliable models 2 and 5. Confining our attention to this first category, we note that model 12 is essentially a rerun of the aborted model 6. It is clear that the outer shells of these models are blowing off and that mass loss is taking place. Model 11 differs from models 6 and 12 by its low initial velocity distribution. It is anticipated that this model which is apparently "quivering" at the surface would show a behavior similar to that of model 12 if it were run long enough to allow the underlying pulsations to build up. Models 15 and 16 represent more finely zoned versions of models 11 and 12. Model 15 which has an initial velocity distribution somewhat higher than that of model 11, essentially exhibits the behavior of model 11. Model 16 which has an initial velocity distribution higher than that of model 15, but not as high as that of model 12, acts like model 15 for a while, but then develops an expanding shell as in model 12. This tends to confirm our hypothesis concerning the low initial velocity models. It is to be noted that the relatively high initial velocity marginally reliable models 2 and 5 tend to indicate mass loss by their expanding surface

radius. In addition, model 5 suggests an increase in the period of the light and radial velocity curves to several pulsational periods as the surface expands. This is borne out by figures (6.7) and (6.8) which display a computer plot for the light curves of models 12 and 16 respectively.

Model 16 represents our most reliable and physically realistic β Cephei model. Its coding would permit us to detect a mass loss at the stellar surface as little as 10^{-8} $M_{\odot}$. The model's time resolution is approximately 1/2000 of the pulsational period. Restrictions on real and computer time, allowed us to run model 16 out for only about 8 pulsational periods. This was adequate to show that qualitatively it was approaching the behavior of model 12. Because of its finer zoning, we used model 16 to estimate the quantitative picture of mass loss presented in chapter 7. Figures (6.1) through (6.10) represent the time development of selected features for models 12 and 16. We include model 12 because of the relatively short duration of the more accurate model 16. Figures (6.11) through (6.16) are computer plots of the luminosity curves for model 12 at the indicated masses below the surface. The deepest plot is for mass fraction 0.9998. Figures (6.17) through (6.23) are computer plots of similar curves for model 16. These overlap, in mass, the curves for model 12, but extend to the much greater depth of mass fraction 0.96. The

deeper plots illustrate the strictly sinusoidal behavior of the pulsations below the outermost regions of the star. Note that the deep light curves vary with exactly the pulsational period. One should also note the manner in which the period lengthens as we approach the surface, as well as the time development of this lengthening for both models 12 and 16. Figures (6.24) through (6.30) represent detailed computer plots of the luminosity curves for model 12 corresponding to the time interval between underlying pulsations number 19 and 38. These curves allow us to follow the phase relations between the luminosity fluctuations at various depths at a time well after the surface zone is blown off the model. Examination of figures (6.22) and (6.23) reveals that there is no apparent luminosity phase shift below mass fraction ~ 0.98. Comparison of figure (6.25) with figure (6.26) and subsequent plots reveals a luminosity curve phase shift with depth in a luminosity curve phase shift with depth in the models outermost regions indicative of an outwardly propagating supersonic disturbance. This is born out when considering the behavior of other model parameters. All of this is suggestive of shock wave production. Our model is in many respects analogous to a piston driven shock tube. Detailed analysis leads to

a shock build up cycle of several pulsational periods, which is comparable to the characteristic time for surface luminosity fluctuations as displayed in figure (6.7). As table (6.1) indicates, the mass content of the surface zone of model 12 is $\sim 8 \cdot 10^{-8}$ as compared to $\sim 8 \cdot 10^{-9}$ for model 16. Moving inward, successive zone sizes increase by approximately 50 per cent for both models. We can follow any shock effects on the outer zones of either model by comparing zone velocities obtained from our detailed printed output with calculated escape velocities. The significance of the results from models 12 and 16 is summarized in Chapter 7.

We now must examine the results of our usable models which do not utilize the thermal viscous term. The models to consider are models 7 through 10 and model 13. A cursory perusal of table (6.1) indicates not the slightest inclination toward mass loss for any of these models. The inclusion of artificial viscosity solely in the hydrodynamic equation acts as a damping mechanism for shocks without providing a means for pumping the shock energy back into the envelope. The effect of this is to remove the mass loss mechanism. These models may, therefore, reflect the manner in which an under-

lying β Cephei object should pulsate after undergoing considerable mass loss. Their light and velocity curves may be more suggestive of observational features than are mass loss models. Referring to table (6.1), we find that, with two reservations, the sequence of models 7 through 10 give rise to light and radial velocity curves that have common features. The light curves display a major peak with a relatively well defined secondary peak trailing by about 1/3 of a period. The radial velocity curves are composed of a main peak with a closely trailing sideband. The first reservation to the above account comes when considering model 7 after C_q has been raised to 400. This increase in artificial viscosity produces a smoothing effect which obliterates the structure for both light and velocity curves. Our second reservation concerns the zero initial velocity distribution model, which as time goes on seems to be relaxing to sinusoidal light and radial velocity curves. All the above curves show a consistent skewness as evidenced by an examination of table (6.1). It is important to note that the more reliable model 13 produces results that are in substantive agreement with those outlined above. The significance of these results is discussed in Chapter 7.

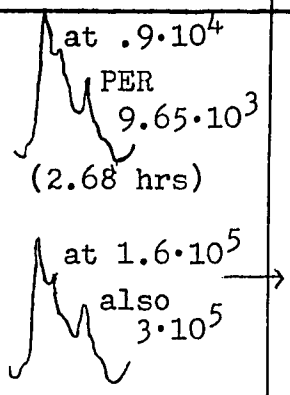
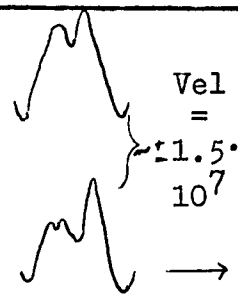
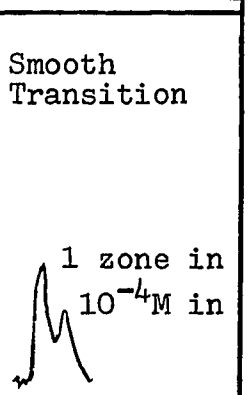
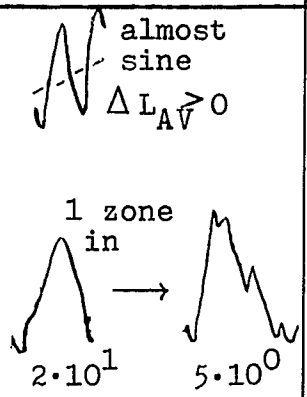
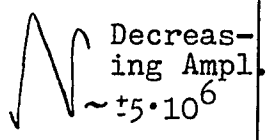
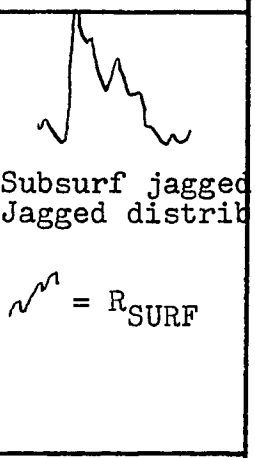
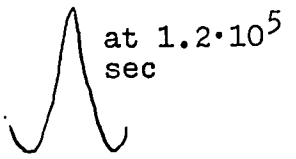
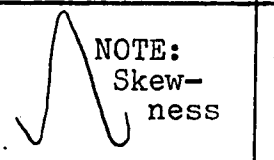
In closing, we make the observation that all the β Cephei models produce nearly identical results as we

move a small mass fraction inward from the surface. Furthermore, points of maximum intensity for each model's light curve correspond to minimums of surface radius. If a periodic velocity curve can be established for a model, it invariably leads the light curve by 90^{o}.

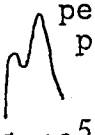
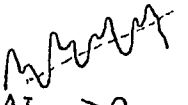
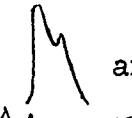
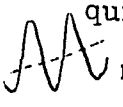
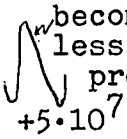
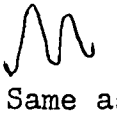
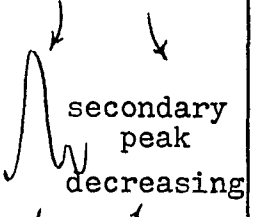
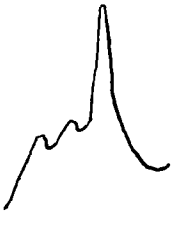
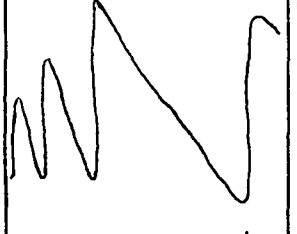
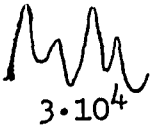
TABLE 6.1

SUMMARY OF β CEPHEI MODELS

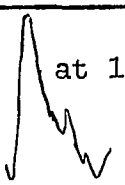
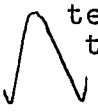
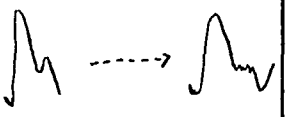
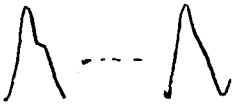
	CODE FEATURES	ZONING	VISCOSITY
β CEPHEI MODEL # 1 $\delta R/R \sim 0.1$	Viscosity in hydro. eq. only Eddington B.C.'s, Centered deriv's Opac. at surf avg of subsurf and Elec Scat $V_i(0) \propto V_i^{100M_\odot}$	$\Delta M = 0.005M$ Last Zone subdiv. into 3 the last of which is subdiv. into 3 nested 8 deep N = 216 $(\Delta M)_{min} \backsimeq$ $7.6 \cdot 10^{-7} M$	10^2 at 0.0sec 50 at $\sim 9^+ \cdot 10^4$ sec. 100 at $1.647 \cdot 10^5$ 50 at $2.411 \cdot 10^5$
MODEL # 2 $\delta R/R \sim 0.09$ at $2.26 \cdot 10^5$ sec.	same as # 1 BUT: $Q_i^k \frac{\left(\frac{1}{\rho_i^k} - \frac{1}{\rho_i^{k-1}}\right)}{T_i^k}$ added to $\frac{dS}{dt}$ term.	same as # 1	20 at 0.0sec. 5 at $2.26 \cdot 10^5$ End at $3.38 \cdot 10^5$
MODEL # 3 $\delta R/R \sim 0.06$	same as # 2	UNIFORM $\Delta M = 0.005M$ N = 200	20 at 0.0sec. 5 at $5.45 \cdot 10^4$

	SURF LUM	SURF VEL	SUBSURF & DISTRIB
MODEL # 1			
MODEL # 2			
MODEL # 3			ALL SINUSOIDAL

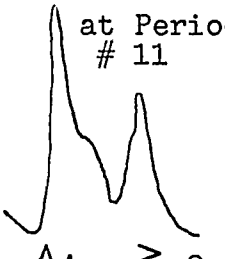
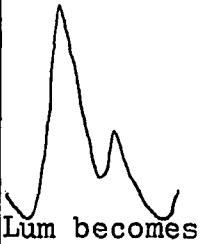
	CODE FEATURES	ZONING	VISCOSITY
MODEL # 4 NOTE: MACHINE ERROR AT 10^5 blows model $\delta R/R \sim 0.01$	Same as # 2 but: non-cent derivs NOTE*: DM(J) used in region $M(J) > 0.9M_{tot}$ slightly larger than correct value ($\lesssim 0.05$)	$\Delta M = 0.005M$ to 0.9M, then Christy Zoning total N = 213 $(\Delta M)_{min} = 6.2 \cdot 10^{-6} M$	50 thruout
MODEL # 5 $\delta R/R \sim .025$ at $2.7 \cdot 10^5$ sec $\delta R/R \sim 0.01$ at $4.6 \cdot 10^5$ sec $\delta R/R$ indeterm at end	Same as # 4 NOTE*: same error	Same as # 4	20 at 0.0sec 5 at $2.76 \cdot 10^5$ 1 at $4.69 \cdot 10^5$ End at $6.06 \cdot 10^5$ sec
MODEL # 6 $\delta R/R$ indeterm* NOTE: 1st run for Model 6 made with same errors as 4 and 5 {RESULTS substantially same as presented at right for 2nd run	$-4\pi Q_i \frac{R_{i+1}^2 V_{i+1} - R_i^2 V_i}{DM_{2i} T_i} DZ$ Added to thermal term photospheric B.C.'s with true opacity Non-cent diff.	$\Delta M = 0.005M$ to 0.9M then Christy Zoning N = 223 $(\Delta M)_{min} = 7.14 \cdot 10^{-8} M$	50 thruout

	SURF LUM	SURF VEL	SUBSURF & DISTRIB.
MODEL # 4	Reversed from peak 2 per prior  $1 \cdot 10^5$ sec  $\Delta L_{AV} > 0$	 and $\Delta A_{vel} \ll 0$	NOTE: After machine error perturb all subsurf quantities $\longrightarrow$ sine
MODEL # 5	 $\Delta L_{AV} > 0$ 2nd peak getting smaller  quite sine (sym-metric) SINE with ampl. var of 3 per freq.	 becoming less pronounced $+5 \cdot 10^7$ to $-8 \cdot 10^7$  Same as LUM See graphs	basically same  secondary peak decreasing similar
MODEL # 6	3 spikes leading to 	 etc	*Surf. Rad. blows up subsurface erratic down to 0.9998 then LUM last 2 per.  $3 \cdot 10^4$

	CODE FEATURES	ZONING	VISCOSITY
MODEL # 7 $\delta R/R \sim 0.05$ and falling	Same as # 1	Same as # 1	200 at 0.0sec 400 at $1.364 \cdot 10^5$sec
MODEL # 8 $\delta R/R \sim 0.002$	Same as # 1 but $V_i(0) = 0.0$	Same as # 1	200 thruout
MODEL # 9 $\delta R/R \sim 0.08$	Same as # 1 but $V_i(0) \propto R_i(0)$ such that V_n same as for Model # 1	Same as # 1	200 thruout
MODEL # 10 $\delta R/R \sim 0.04$	Same as # 9 but $V_n(10) = \frac{1}{2} V_n(9)$	Same as # 1	200 thruout

	SURF LUM	SURF VEL	SUBSURF & DISTRIB.
MODEL # 7	 at $1.3 \cdot 10^5$  tended toward at high visc.	 	 SINE
MODEL # 8	 per 5 #13		Subsurf Lum sinusoidal alternating peaks of dif. ampl. but $\Delta A \to 0$ (Same for surf. rad.)
MODEL # 9			Similar to #7
MODEL # 10			

	CODE FEATURES	ZONING	VISCOSITY
MODEL # 11 $\delta R/R \sim 0.018$	same as # 6 but: $V_i(0) \propto R_i(0)$ such that $V_n(0) = -10^6$	same as # 6	5 thruout
MODEL # 12 surf. radius blows out as in # 6	same as # 6 but: $V_i(0) \propto R_i(0)$ such that V_n same as model # 6 ($V_n = -8.7 \cdot 10^6$)	same as # 6	5 thruout
MODEL # 13 $\delta R/R \sim 0.09^+$ and falling slowly	same as # 12 but viscosity <u>not</u> included in thermal equation	same as # 6	100 thruout

	SURF. LUM.	SURF. VEL.	SUBSURF & DISTRIB.
MODEL # 11	 3 peak structure at $\sim 10^6$ sec.	Erratic, possible modulation at 3 period interval	Erratic down past 0.9998M
MODEL # 12	sine of period $\sim 6 \cdot 10^4$ sec. at $3 \cdot 10^5$ sec (period decreasing)	constant	See Separate plots
MODEL # 13	 $\Delta A_{Lum} \gtrsim 0$	 $\Delta A_{Vel} \sim 0$	 Lum becomes sine at 0.998M similar to # 6

	CODE FEATURES	ZONING	VISCOSITY
MODEL # 14 Bombs out after 3 periods/ Newt-Raphson develops log of neg. no. because of small zone size	same as # 13	$\Delta M = 0.005M$ to .9M then Christy Zoning total N = 228 $(\Delta M)_{min} \simeq$ $7.7 \cdot 10^{-9} M$	100 thruout
MODEL # 15 $\delta R/R \sim 0.046$	same as # 11 but: $V_n(0) = -3 \cdot 10^6$	same as # 14	5 thruout
MODEL # 16 surf. radius blows out as in # 12	same as # 11 but: $V_n(0) = -6 \cdot 10^6$	same as # 14	5 thruout

	SURF. LUM.	SURF. VEL.	SUBSURF & DISTRIB.
MODEL # 14	Very similar to model # 13 (1st 3 periods). Slight departures but model has not settled	almost identical to 1st 3 periods of Model # 13	almost identical to 1st 3 periods of Model # 13 (in all quantities including radius).
MODEL # 15	similar to Model # 11	similar to Model # 11	Similar to Model # 11
MODEL # 16	Eratic to point at which surf. blows off, then like Model # 12	Similar to Model # 11 until surf. blows off, then like Model # 12	Similar to Model # 11 Up to point at which surface blows off(subsurf lum. looks like surf lum. for Model # 11) Then like Model # 12

FIGURE 6.1

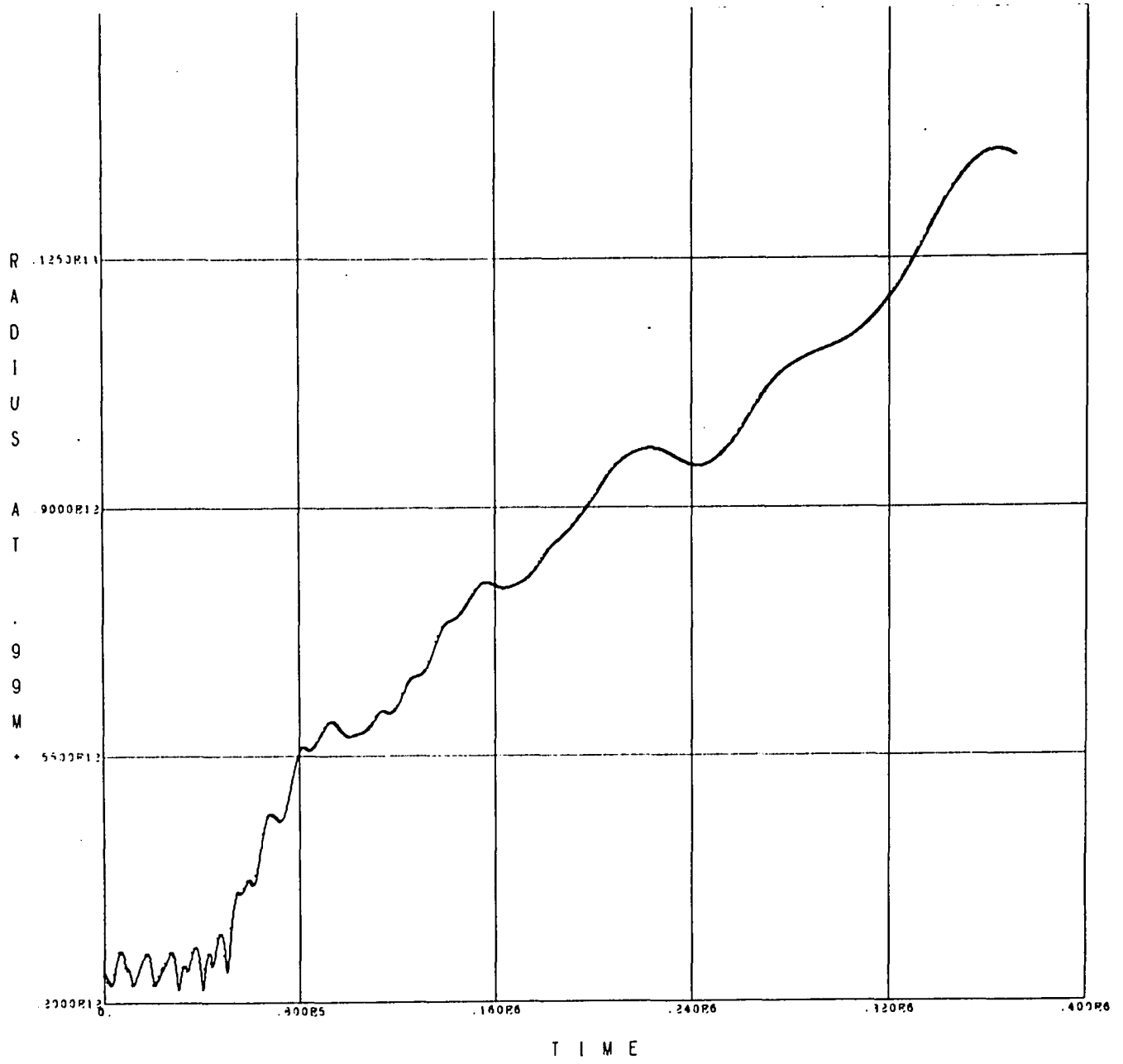
β CEPHEI MODEL # 12 *

RADIUS AT SUBSURFACE

ZONE # 10

$(M-M_i = 10^{-5}M)$

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

223 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/28/70

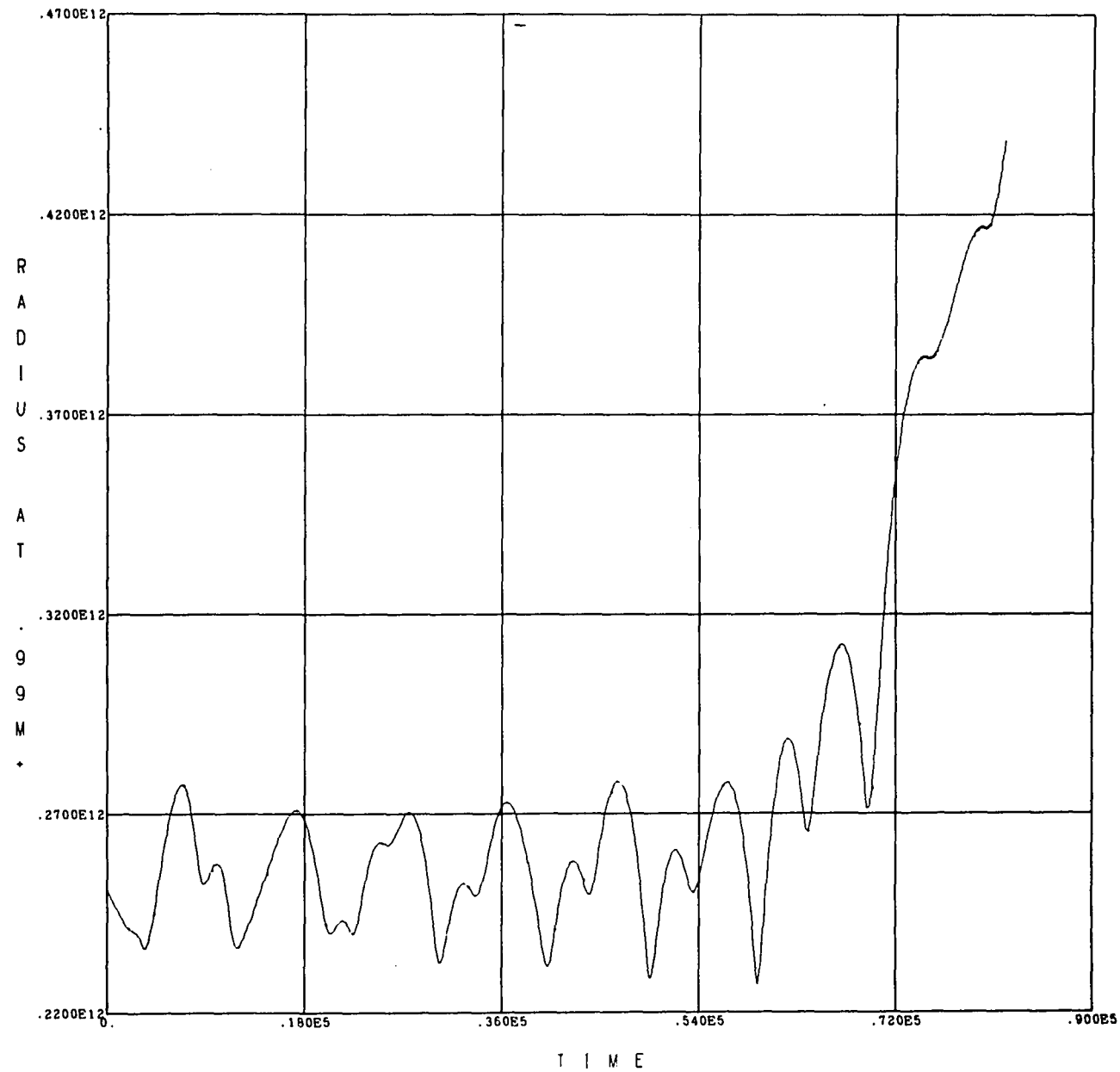
FIGURE 6.2

β CEPHEI MODEL # 16 *

RADIUS AT SUBSURFACE
ZONE # 10

$(M-M_i = 10^{-6}M)$

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

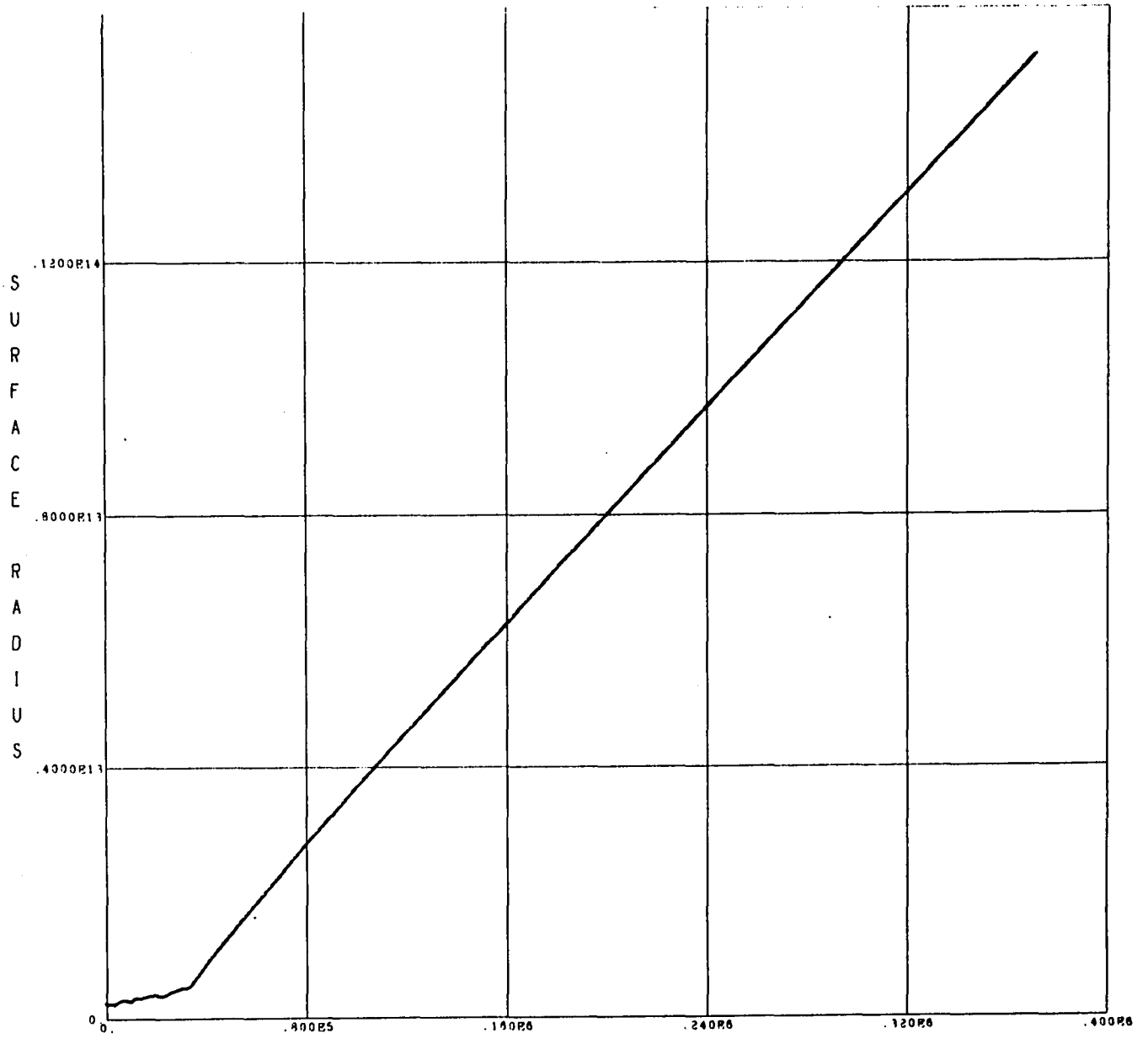
228 ZONES - MASS = 2.984D 34 GM - A CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY 12/03/70

FIGURE 6.3

β CEPHEI MODEL # 12 *

RADIUS AT SURFACE

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

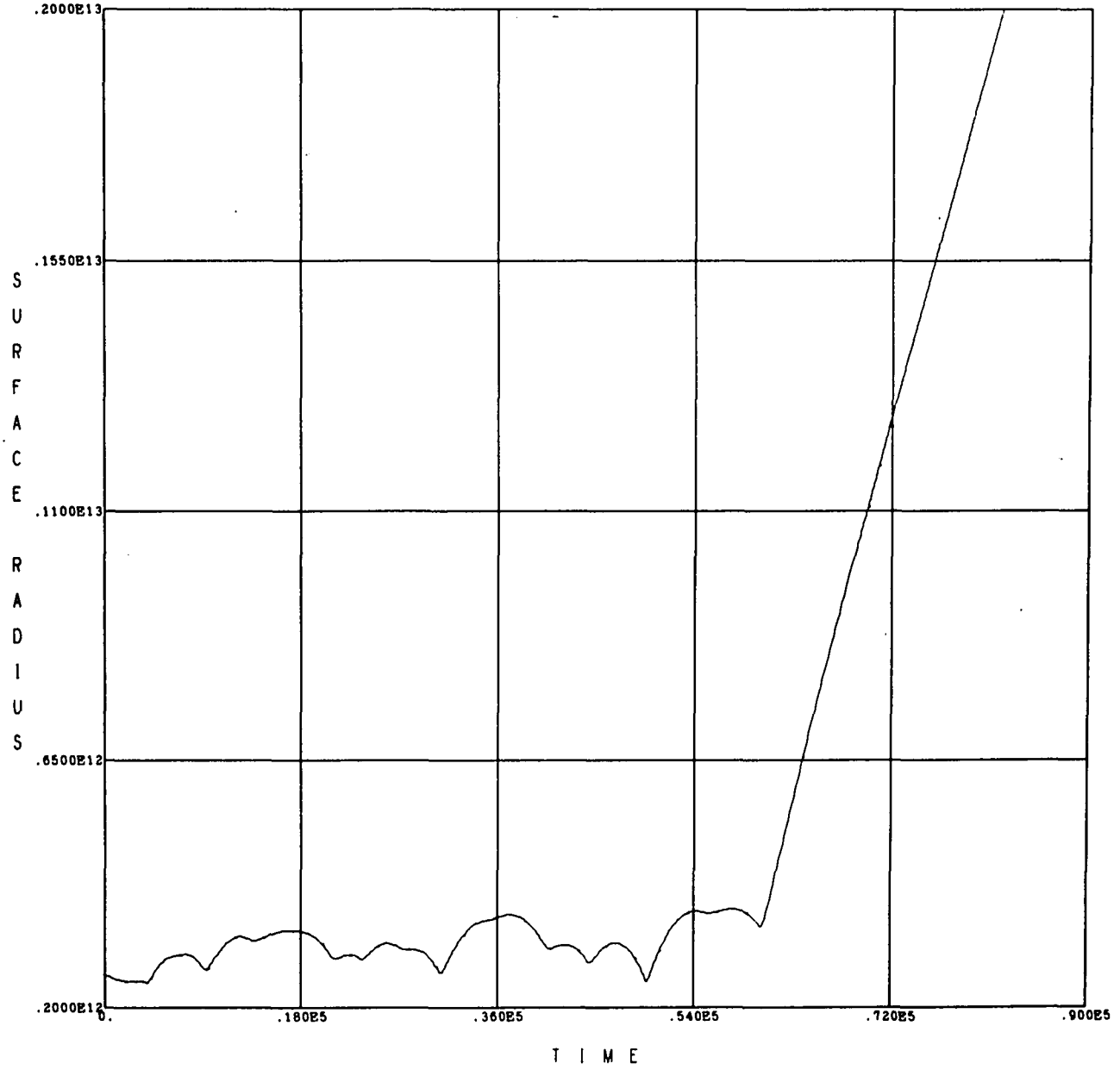
223 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/28/70

FIGURE 6.4

β CEPHEI MODEL # 16 *

RADIUS AT SURFACE

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

228 ZONES - MASS = 2.984D 34 GM - A CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY 12/03/70

FIGURE 6.5

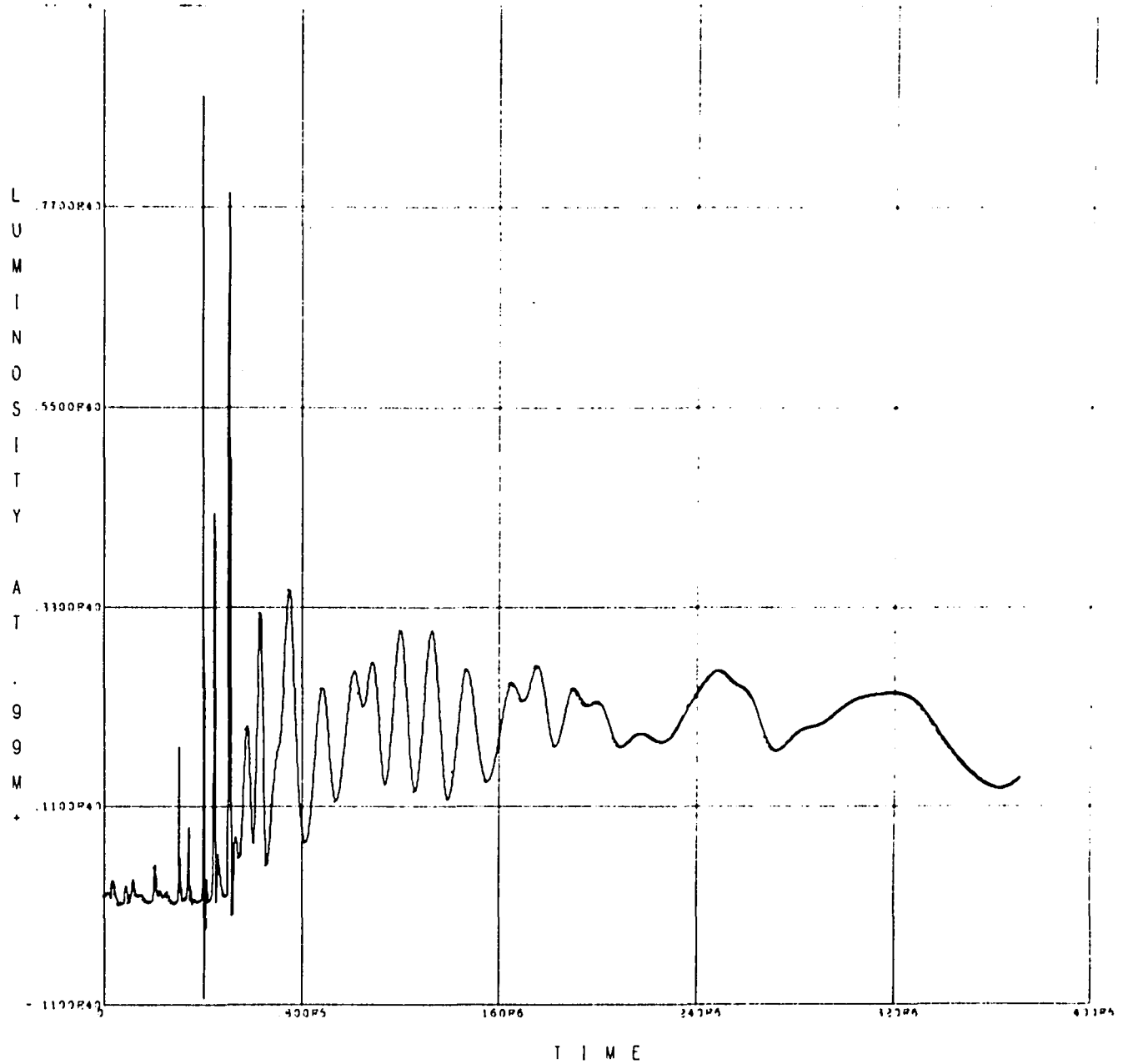
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 10

$(M-M_i = 10^{-5}M)$

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

221 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPUTER OPACITY 08/24/70

FIGURE 6.6

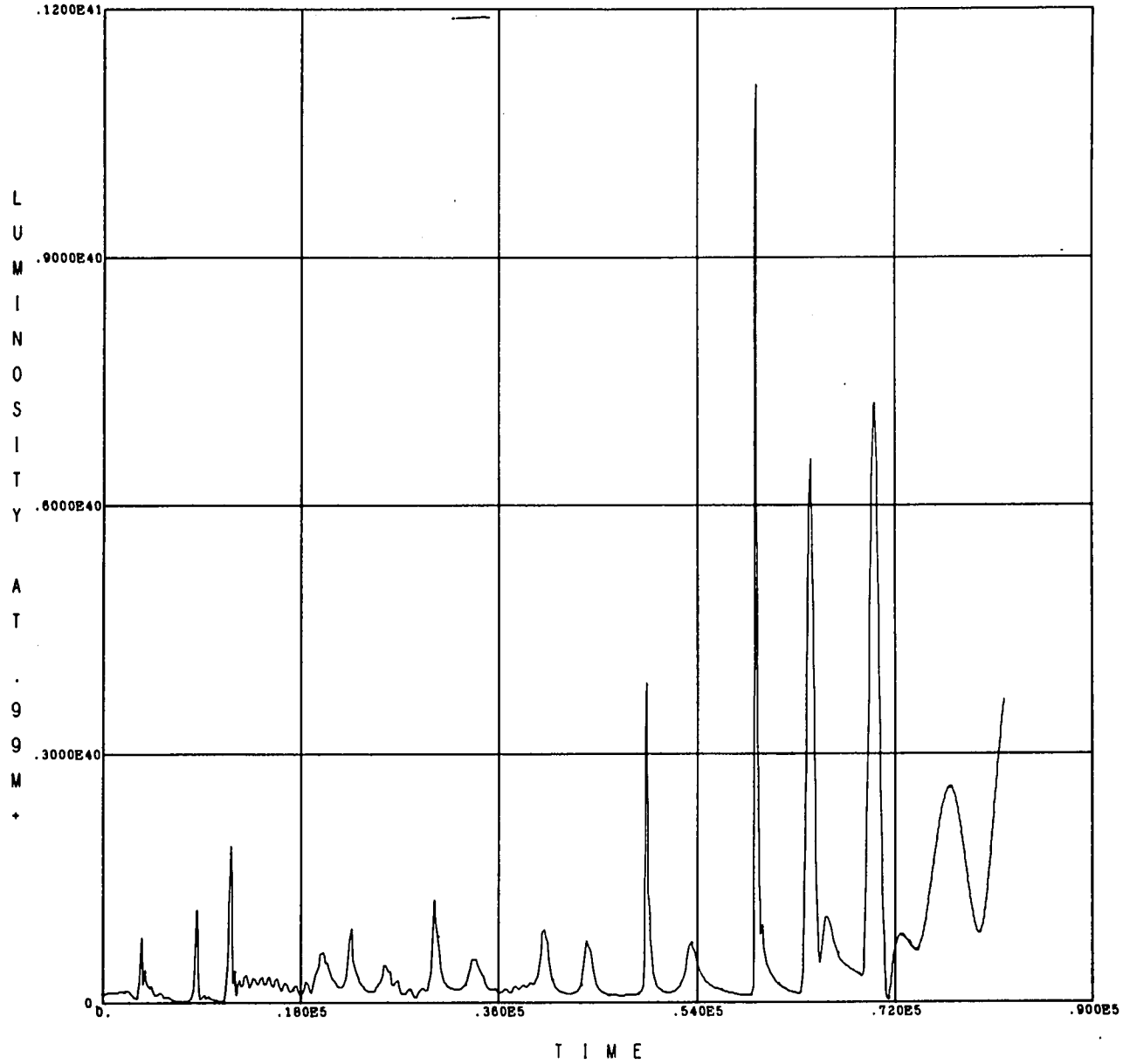
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 10

$(M-M_i = 10^{-6}M)$

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

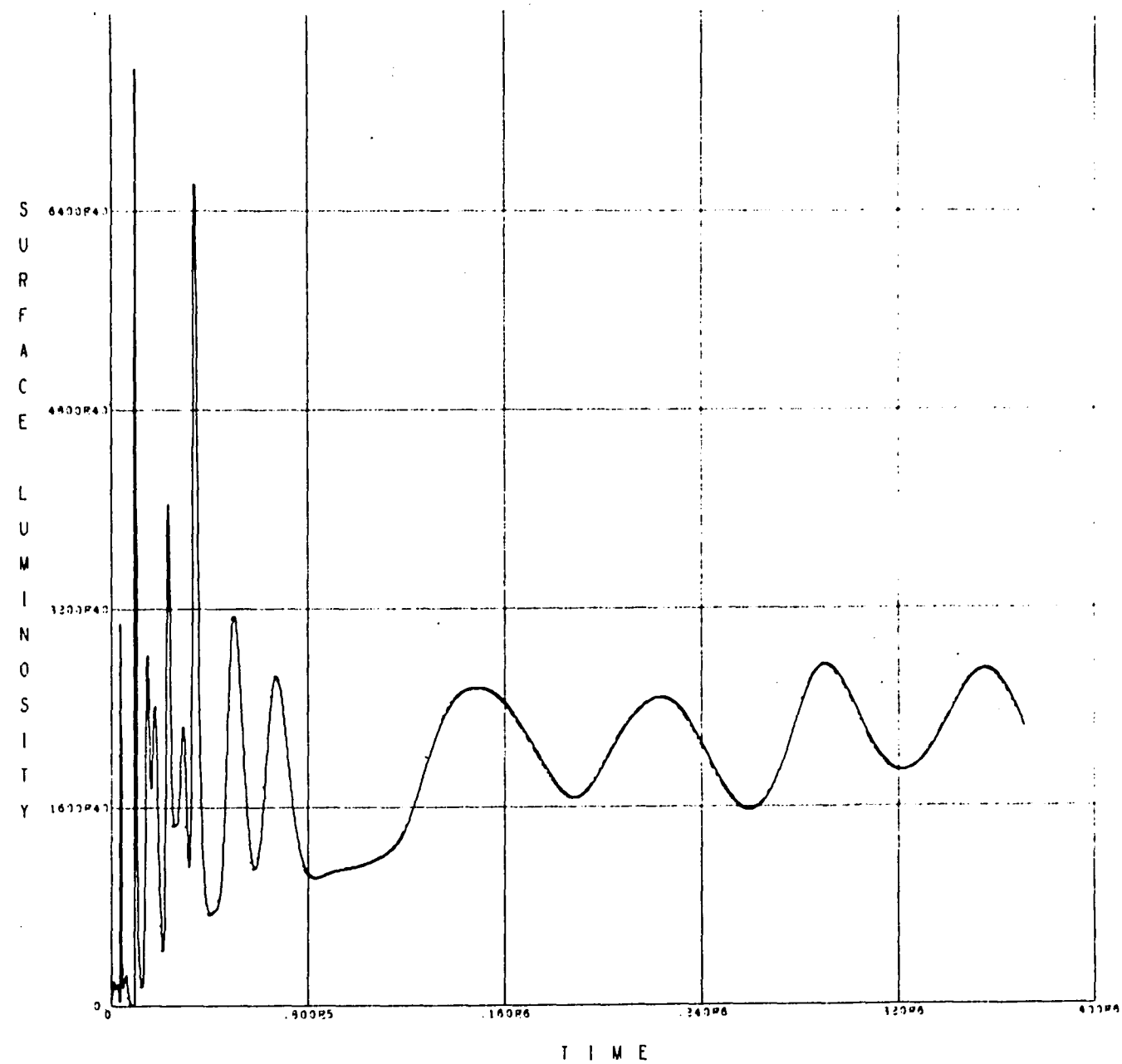
228 ZONES - MASS = 2.984D 34 GM - 8 CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY 12/03/70

FIGURE 6.7

β CEPHEI MODEL # 12 *

LUMINOSITY AT SURFACE

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

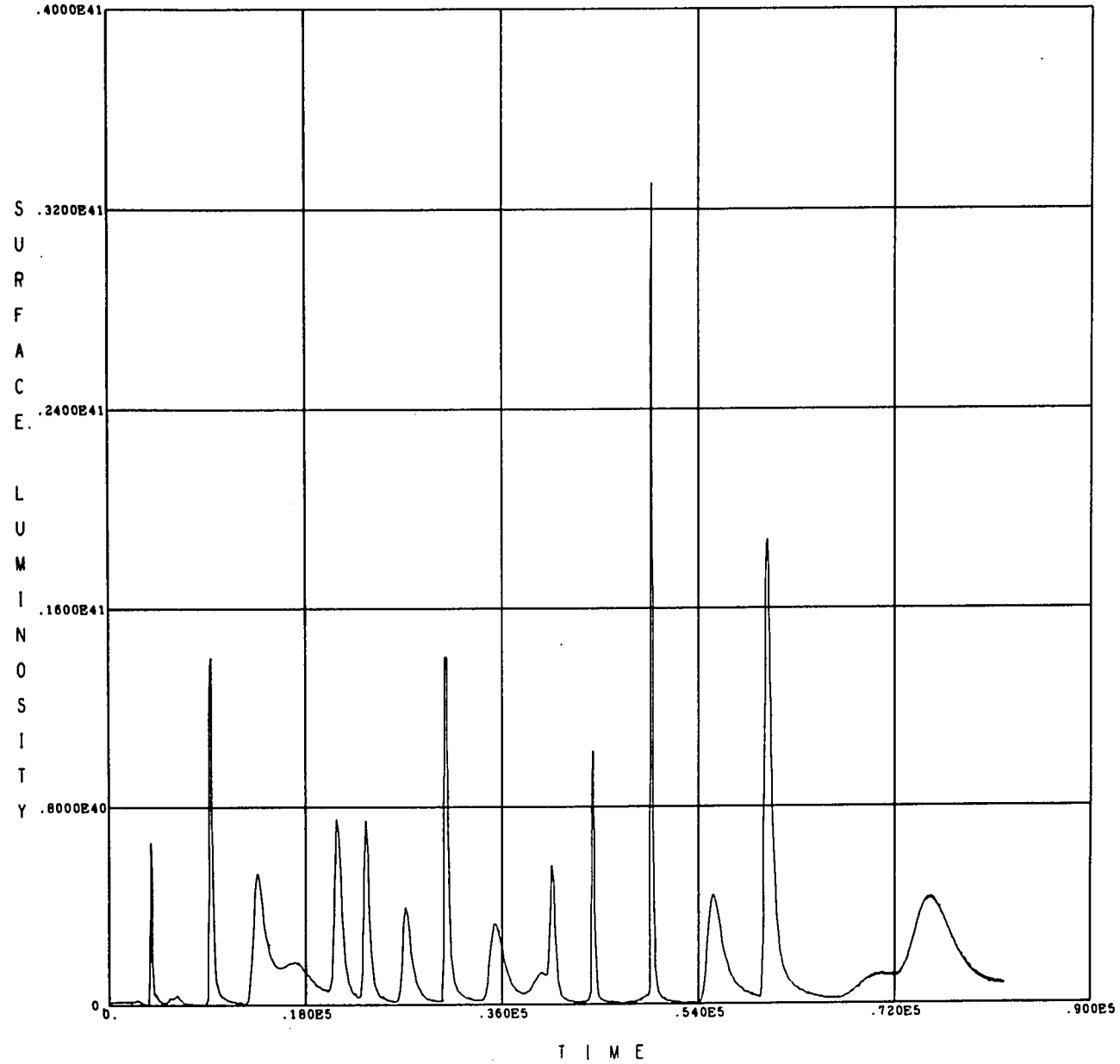
223 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/25/70

FIGURE 6.8

β CEPHEI MODEL # 16 *

LUMINOSITY AT SURFACE

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

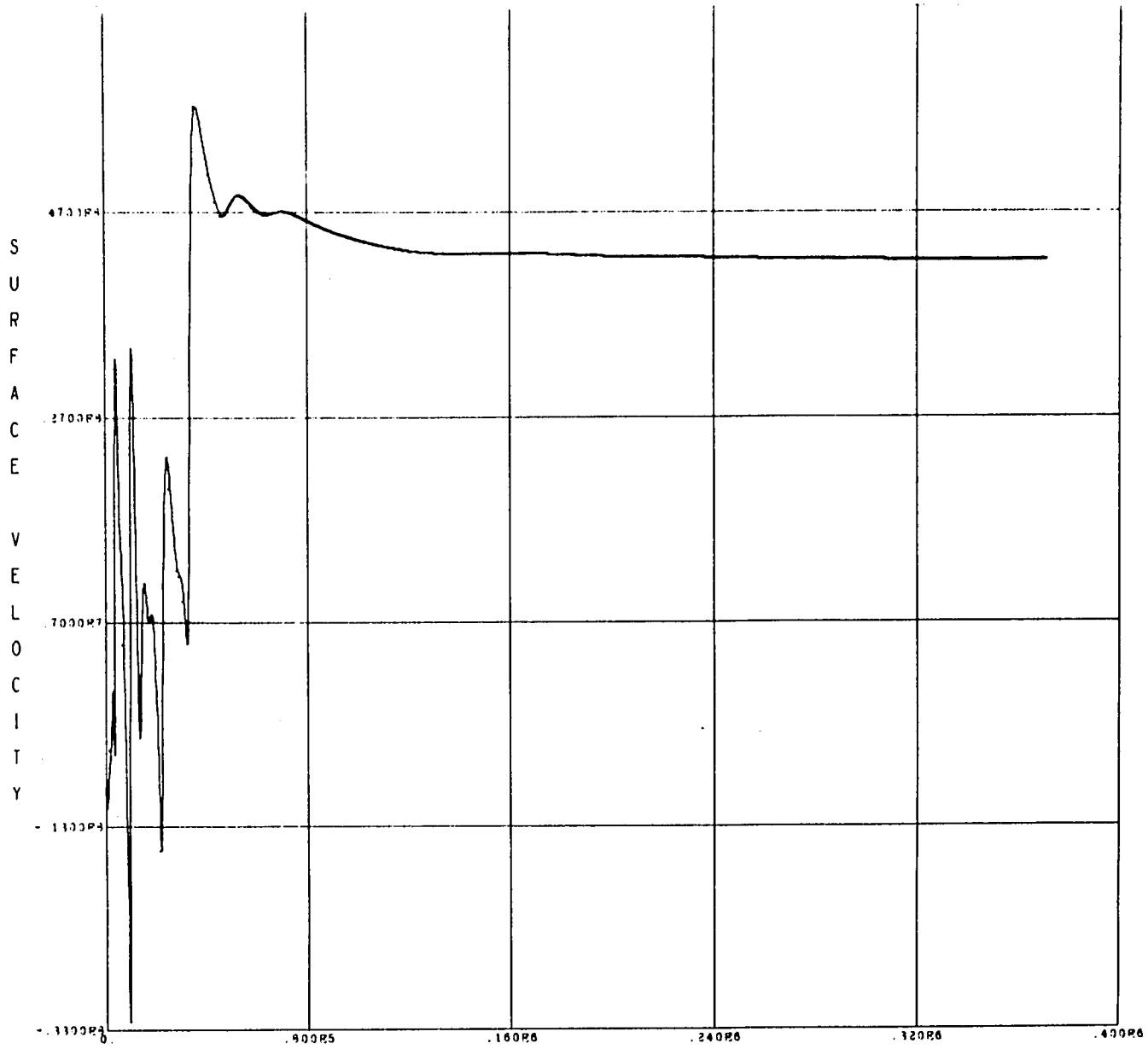
228 ZONES - MASS = 2.984D 34 GM - # CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY 12/03/70

FIGURE 6.9

β CEPHEI MODEL # 12 *

VELOCITY AT SURFACE

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

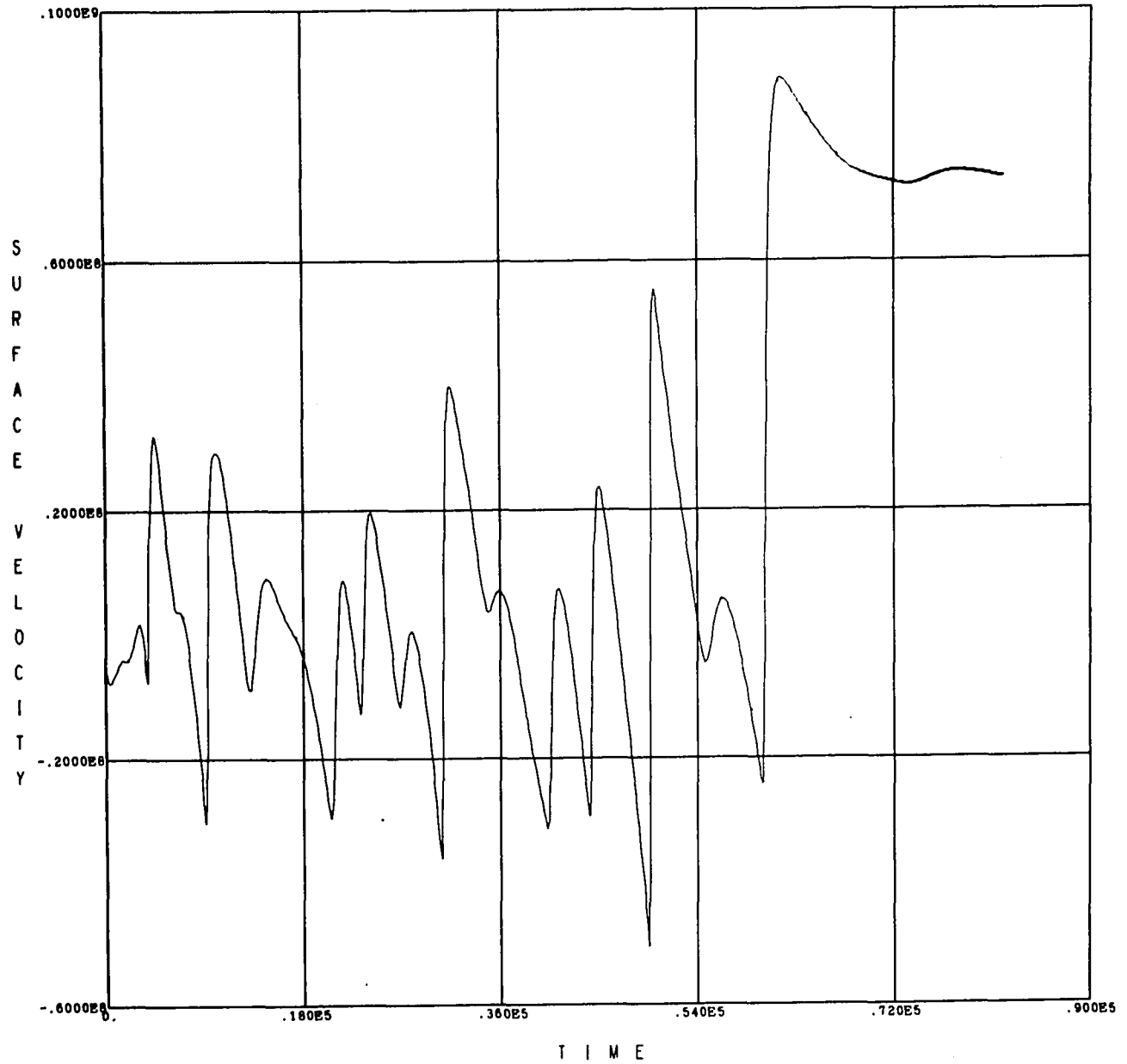
223 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/28/70

FIGURE 6.10

β CEPHEI MODEL # 16 *

VELOCITY AT SURFACE

* computer generated plot



PULSATING STAR - 15 SOLAR MASSES

226 ZONES - MASS = 2.984D 34 GM - A CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY 12/03/70

FIGURE 6.11

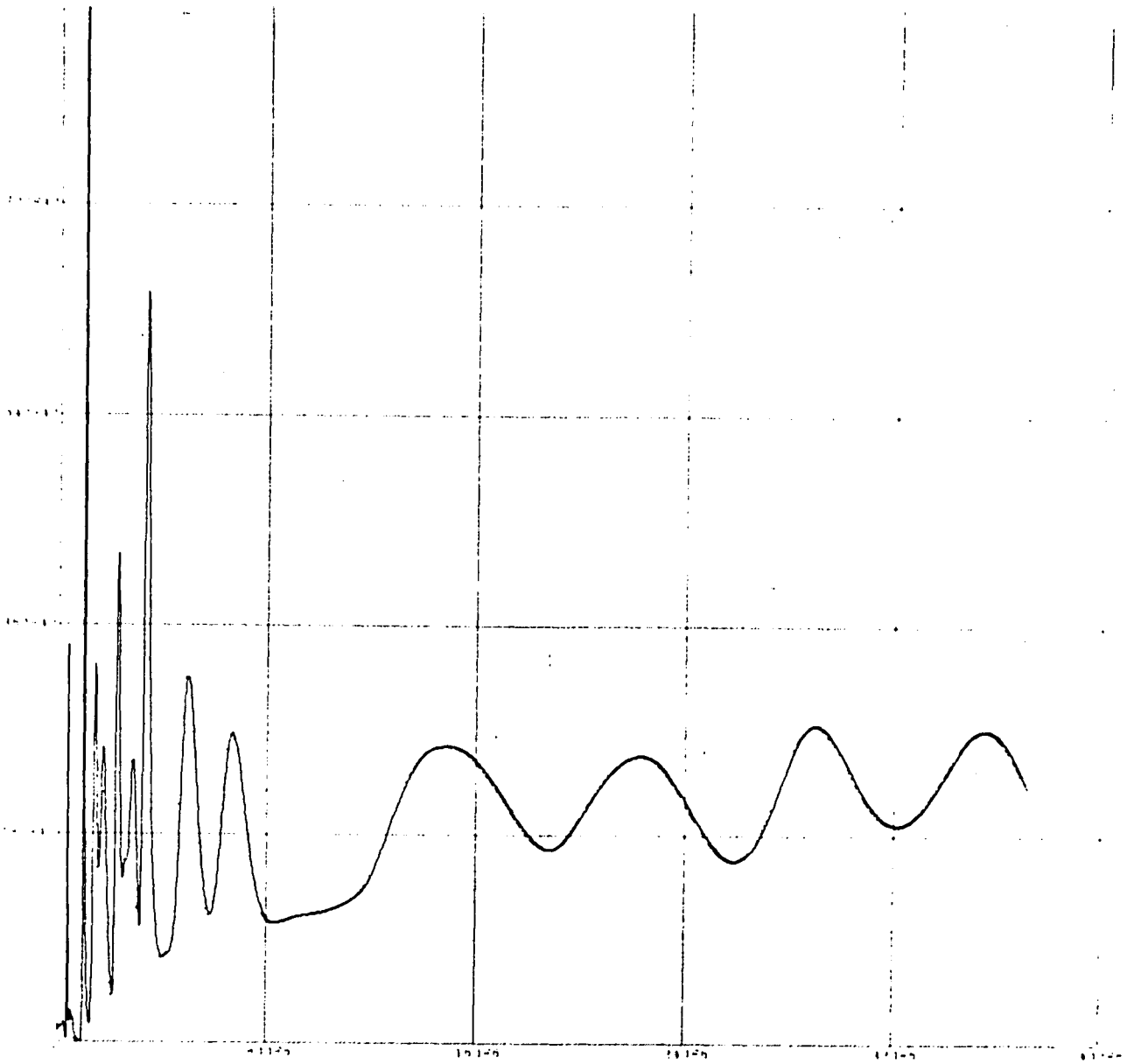
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 1

$(M-M_i = 8.0 \cdot 10^{-8} M)$

* computer generated plot



LUM(2) VS TIME (0=1.0000)

FIGURE 6.12

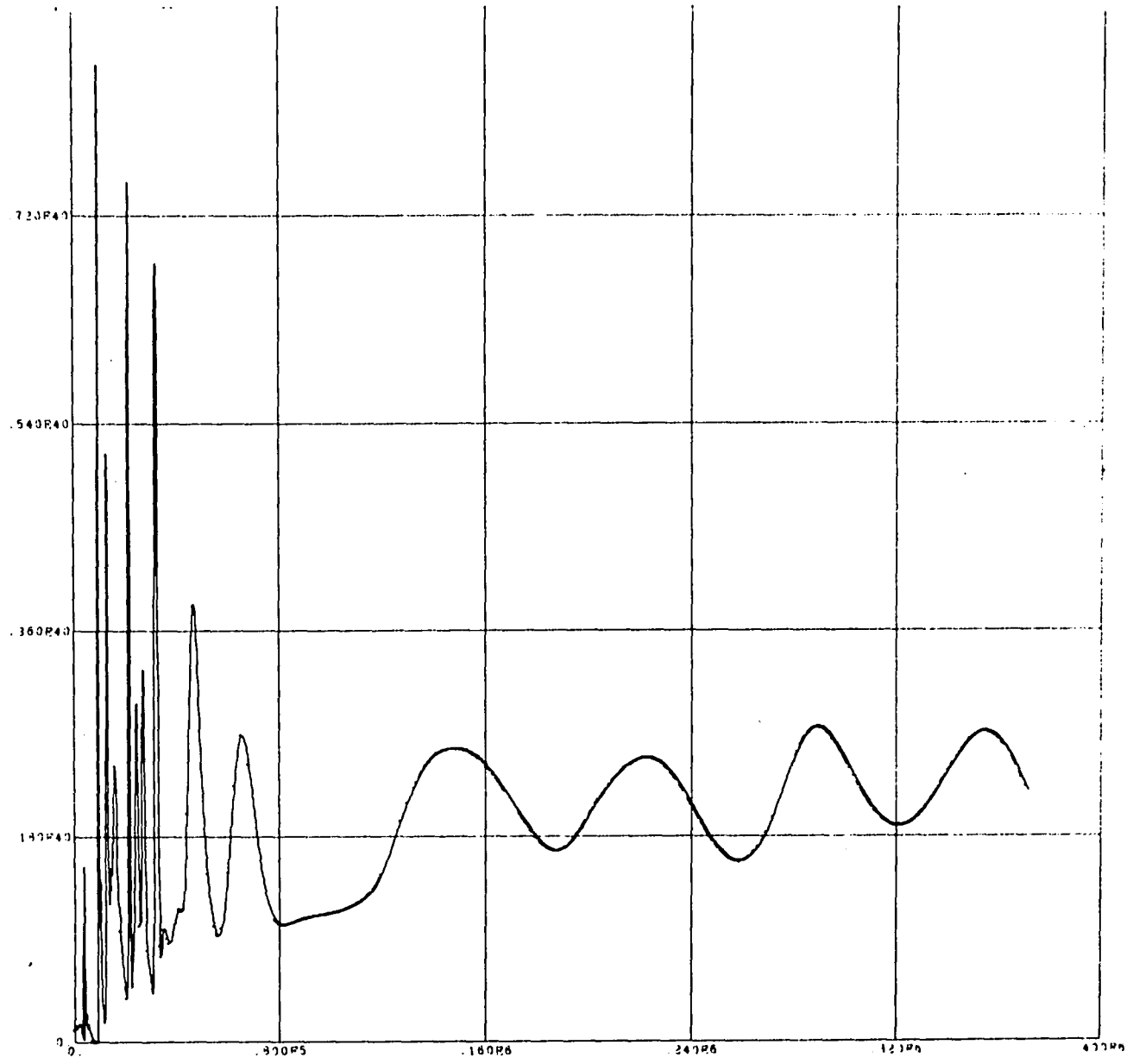
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 3

$(M-M_i = 4.0 \cdot 10^{-7} M)$

* computer generated plot



LUM(4) VS TIME (Q=1.0000)

223 ZONES - MASS = 2.994E 34 GM - A GRPHRID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 04/24/70

FIGURE 6.13

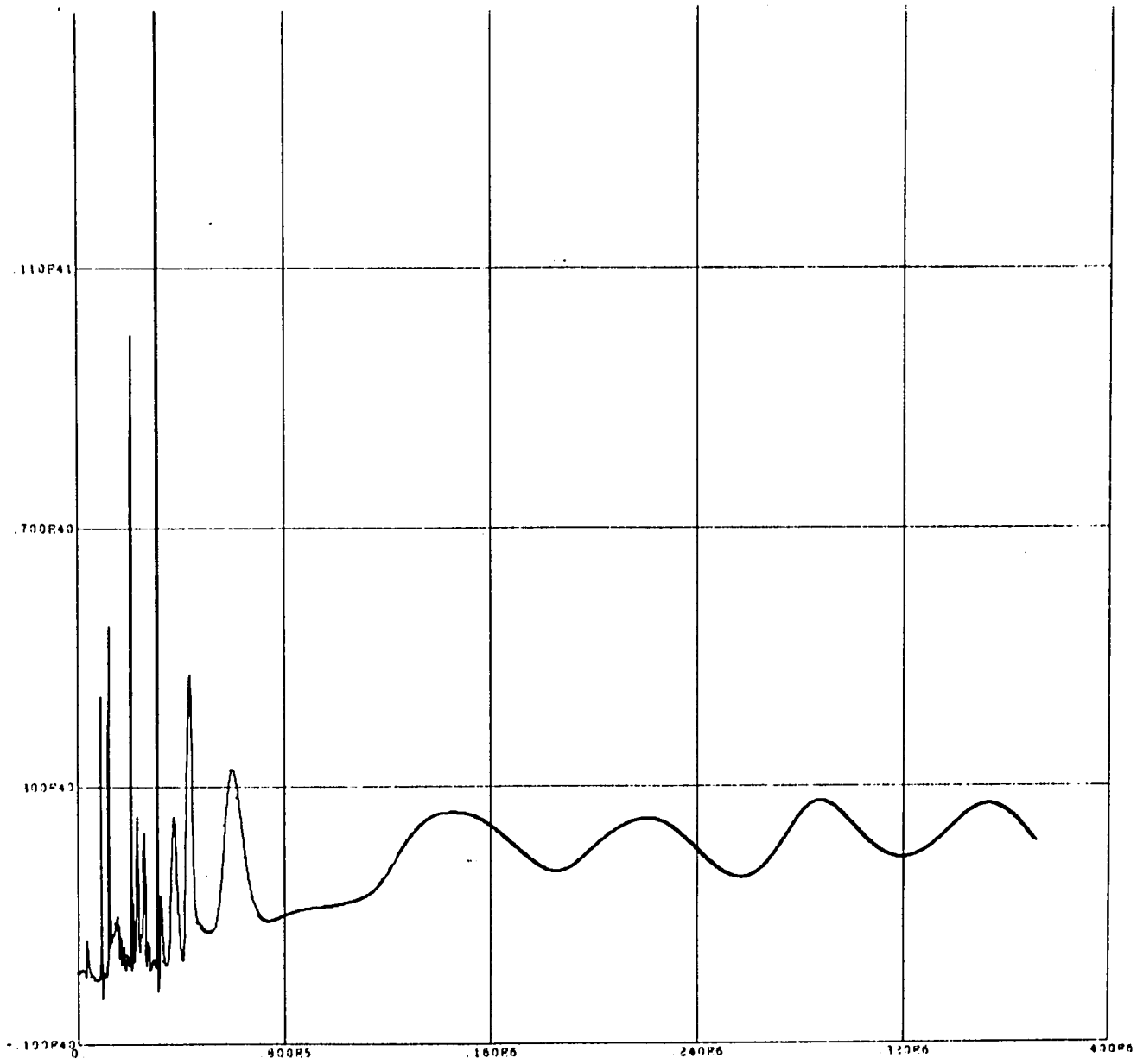
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 5

$(M-M_i = 1.2 \cdot 10^{-6} M)$

* computer generated plot



LUM(6) VS TIME (Q=1.0000)

223 ZONES - MASS = 2.994E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 06/29/70

FIGURE 6.14

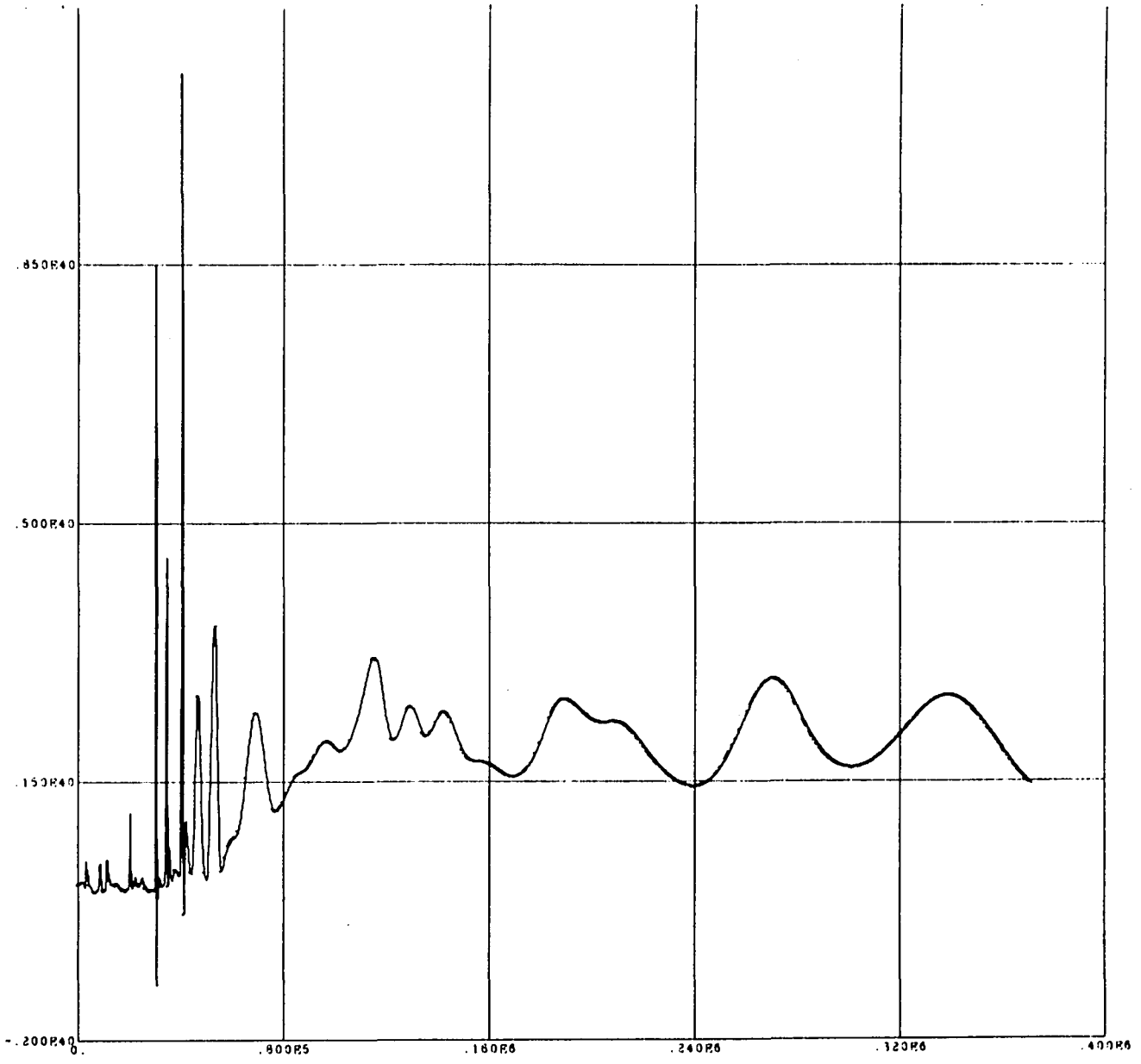
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 8

$(M-M_i = 4.9 \cdot 10^{-6} M)$

* computer generated plot



LUM(8) VS TIME (Q=1.0000)

223 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY CORR = 5.0E 00 - COMPLETE OPACITY 08/28/70

FIGURE 6.15

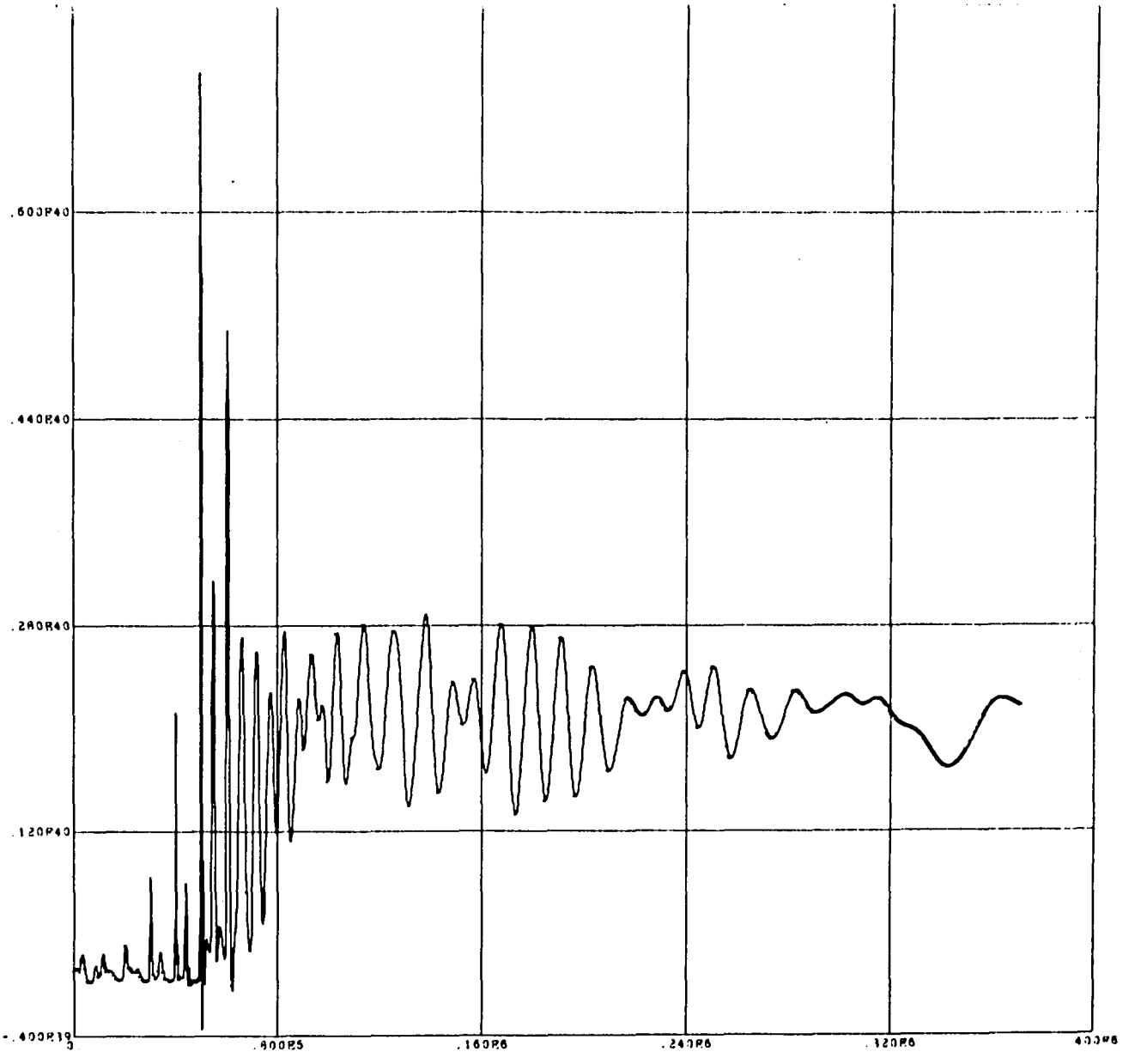
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 12

$(M-M_i = 3.0 \cdot 10^{-5} M)$

* computer generated plot



LUM(10) VS TIME (Q=1.0000)

221 ZONES - MASS = 2.984E 14 GM - 8 CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/28/70

FIGURE 6.16

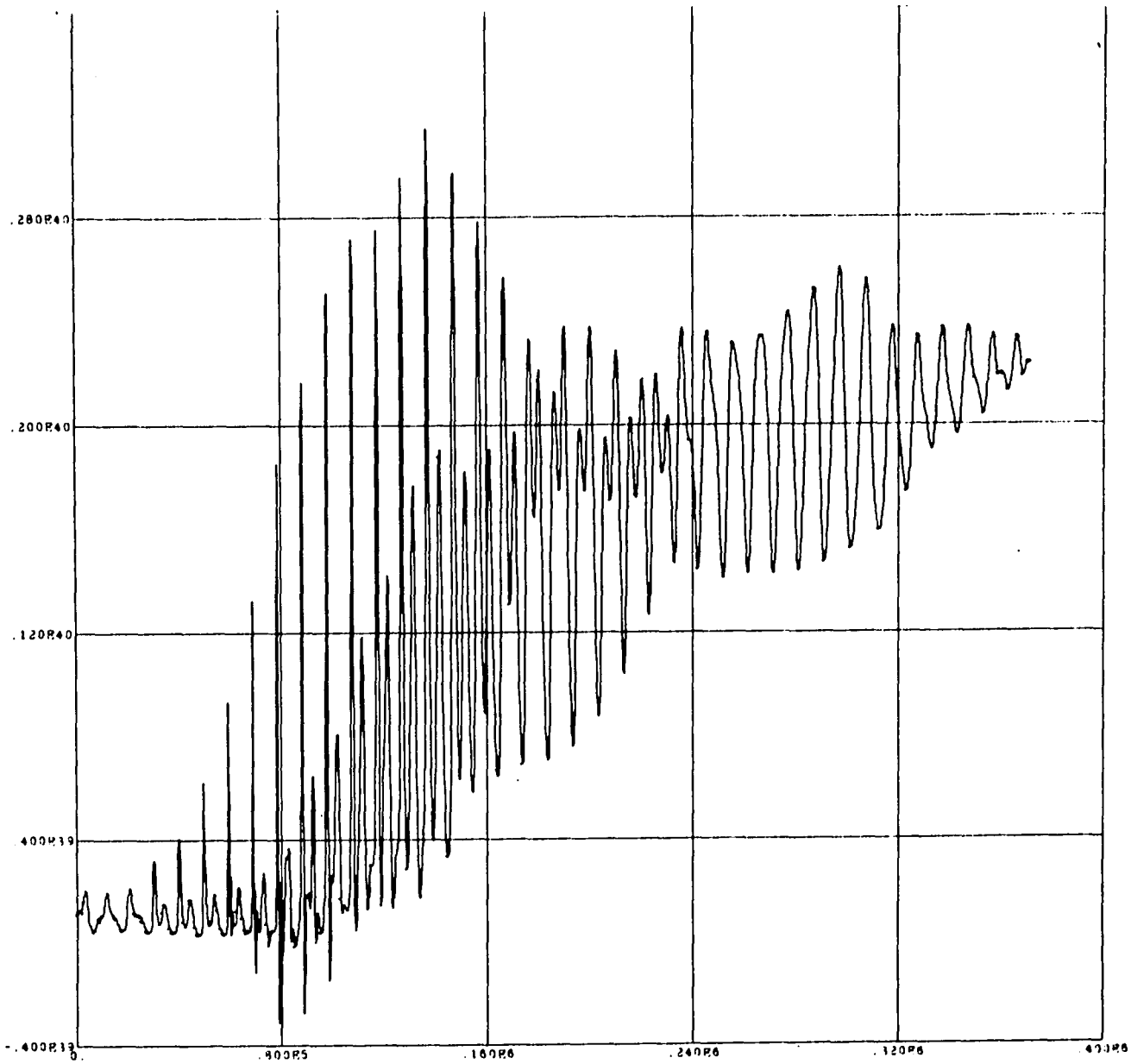
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 16

$(M-M_i = 1.8 \cdot 10^{-4} M)$

* computer generated plot



LUM(12) VS TIME (Q=0.9998)

221 ZONES - MASS = 2.984E 34 GM - 8 CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/29/70

FIGURE 6.17

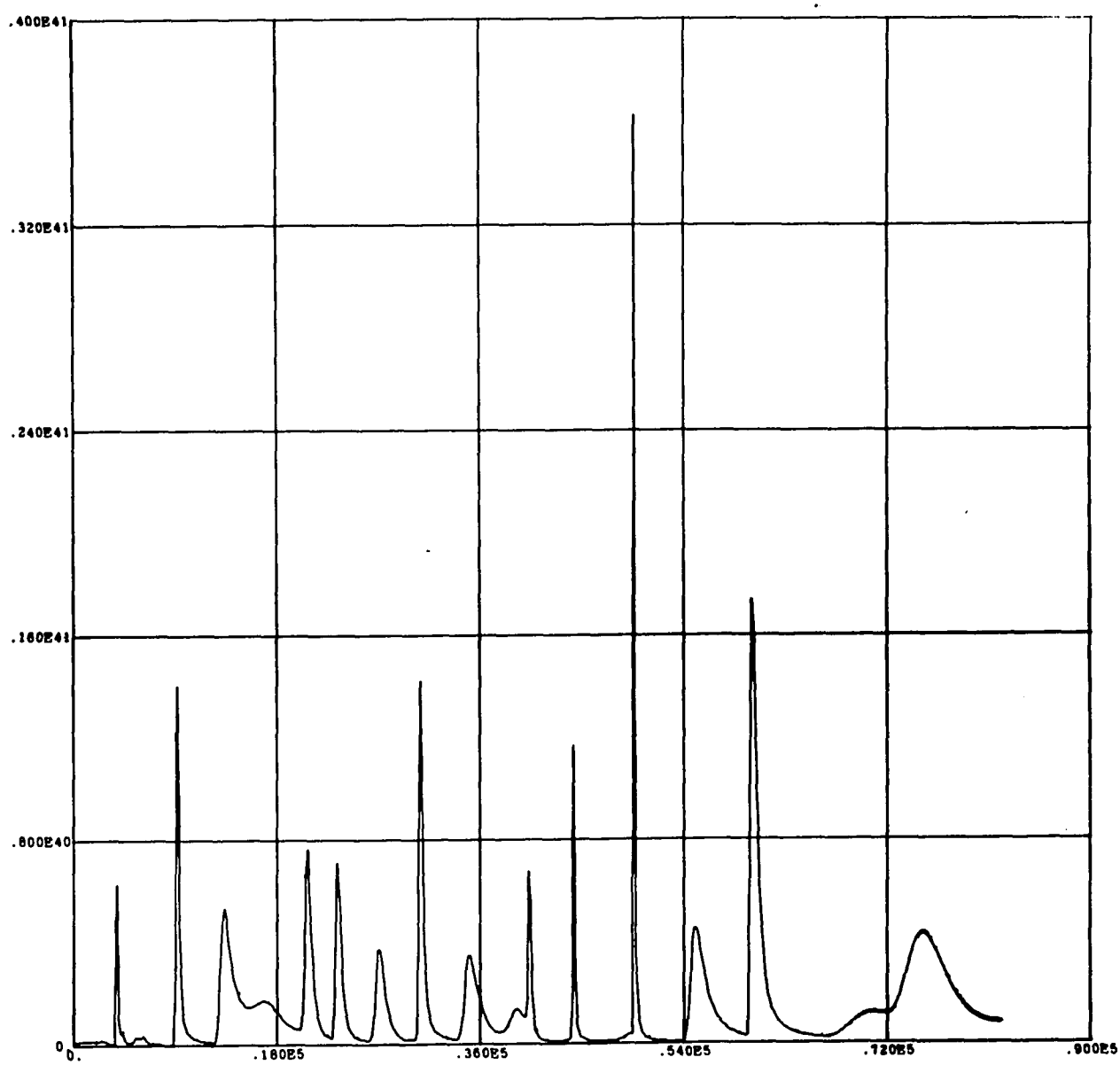
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 2

$(M-M_i = 2.2 \cdot 10^{-8} M)$

* computer generated plot



LUM(1) VS TIME (Q=1.0000)

228 ZONES - MASS = 2.984D 34 GM - 8 CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY

FIGURE 6.18

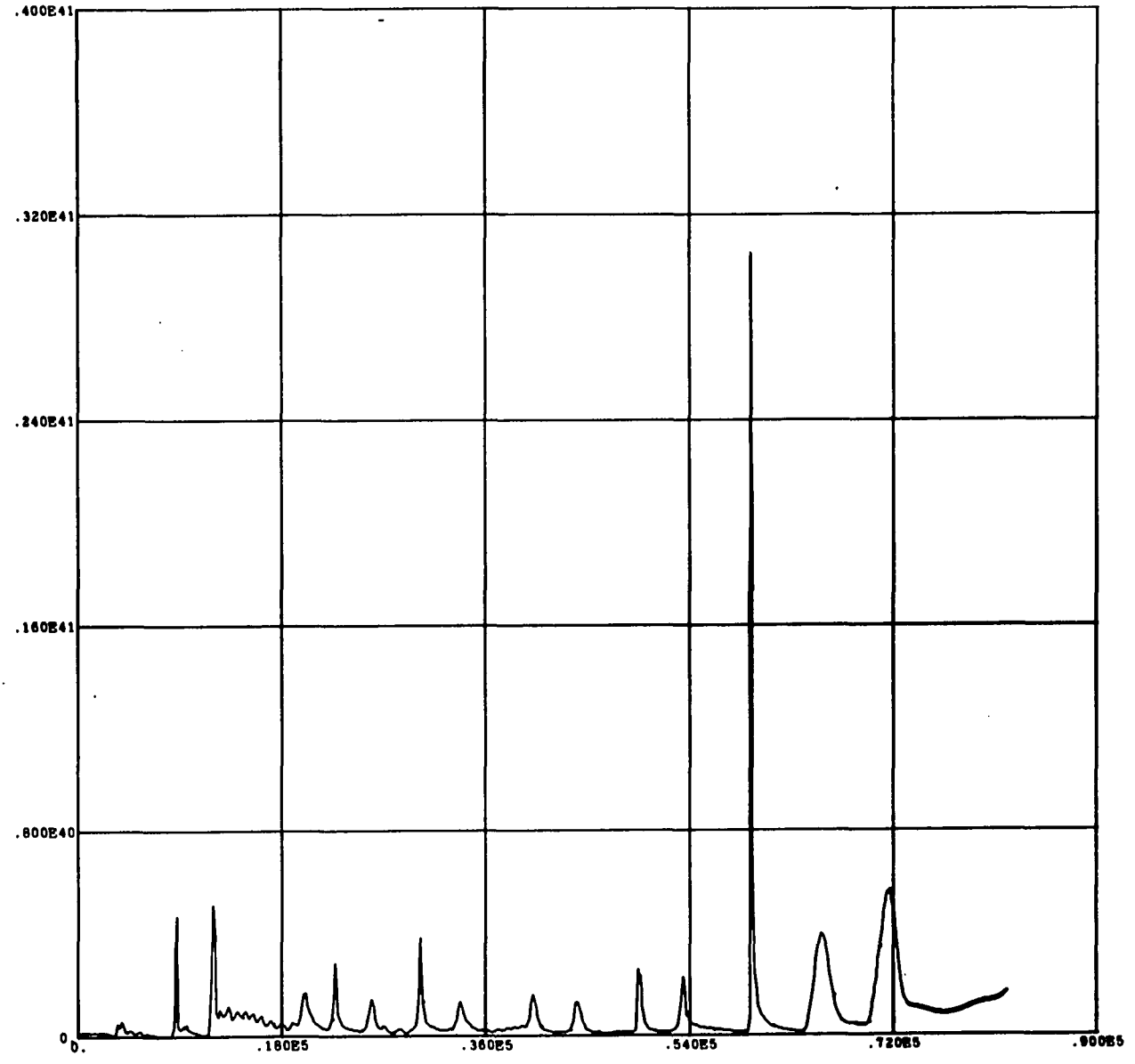
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 8

$(M-M_i = 5.3 \cdot 10^{-7} M)$

* computer generated plot



LUM(4) VS TIME (Q=1.0000)

228 ZONES - MASS = 2.984D 34 GM - 8 CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY

FIGURE 6.19

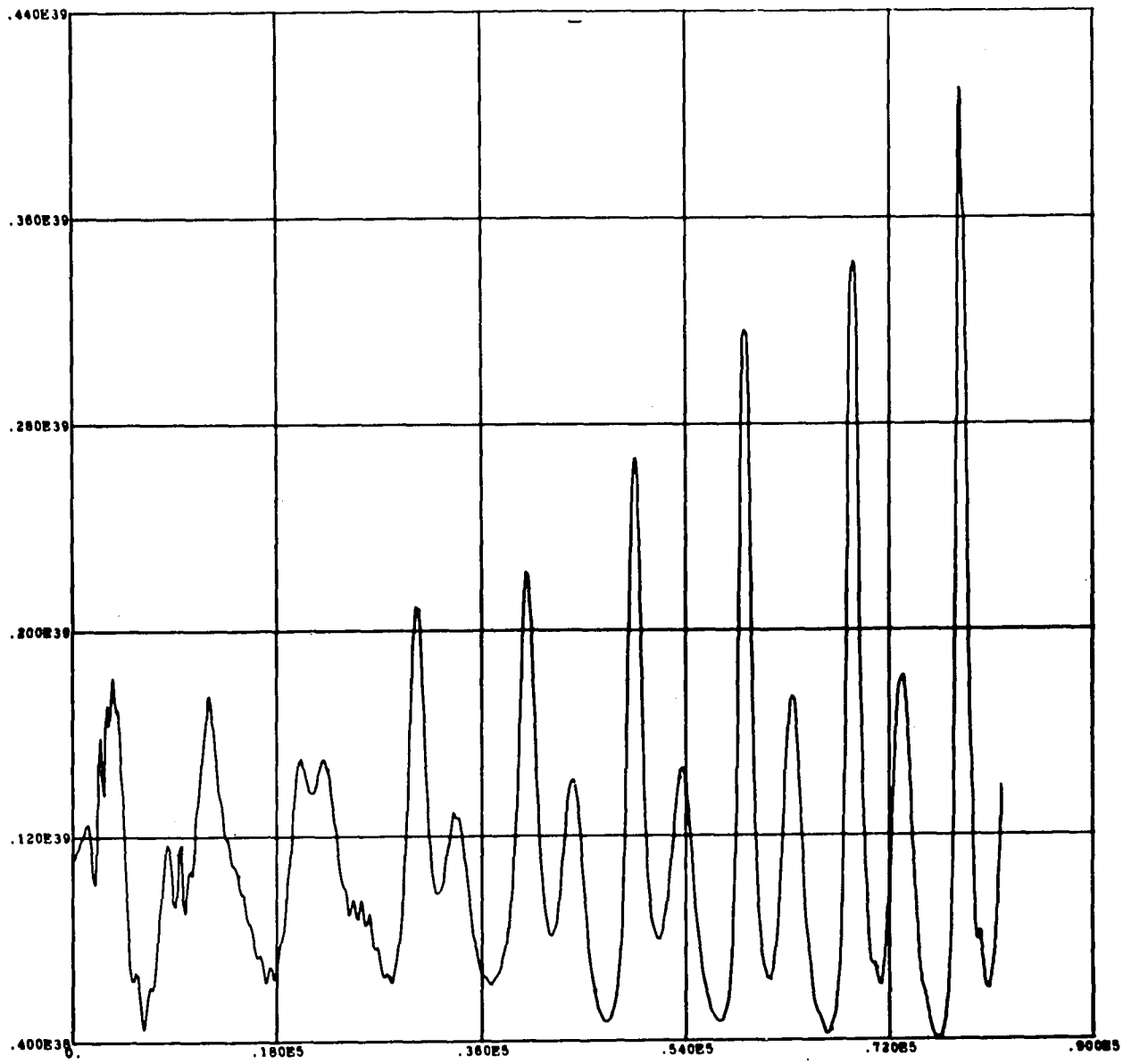
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 20

$(M-M_i = 1.2 \cdot 10^{-4} M)$

* computer generated plot



LUM(7) VS TIME (Q=0.9999)

220 ZONES - MASS = 2.984D 34 GM - 8 CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY

FIGURE 6.20

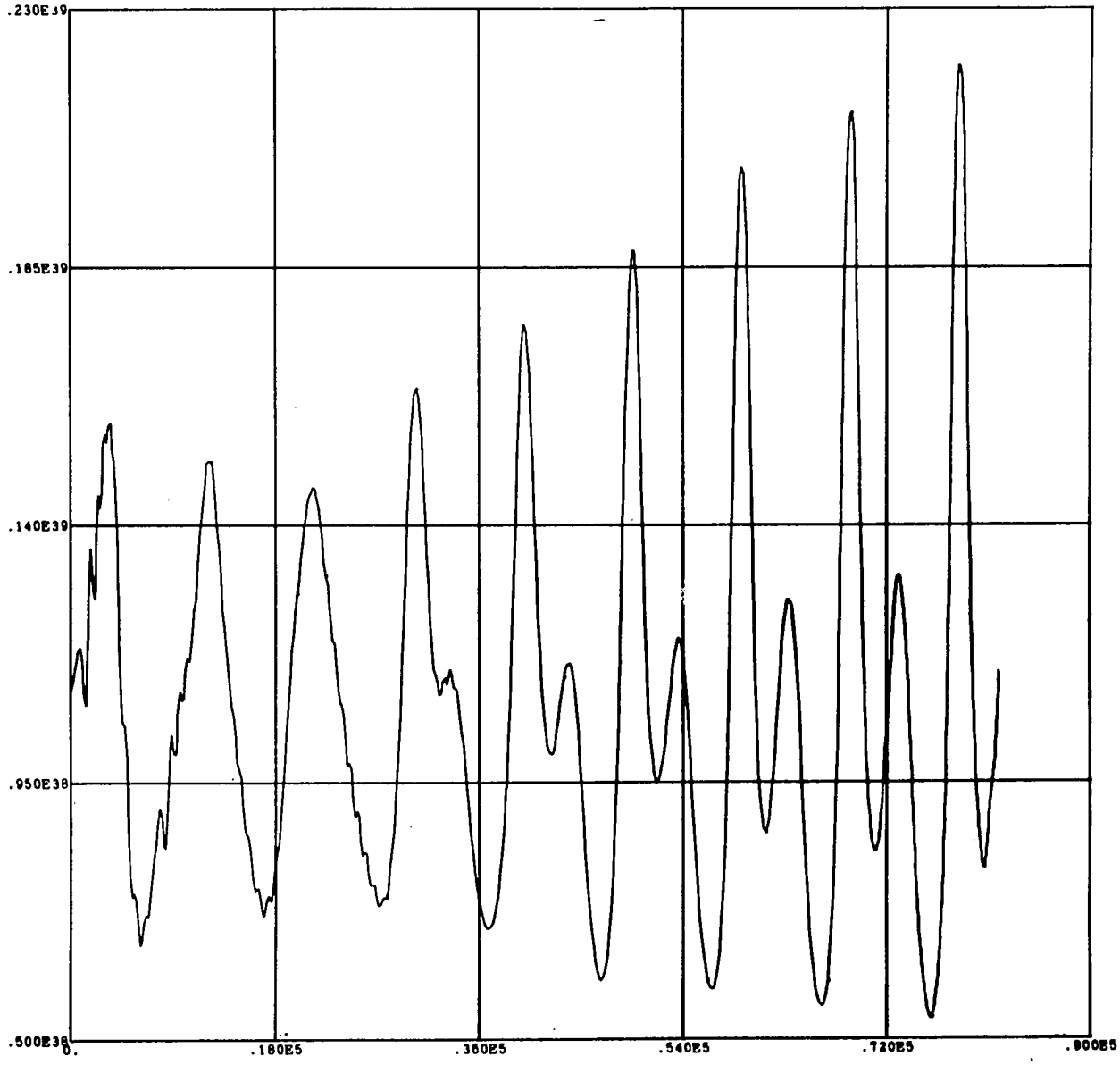
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 24

$(M-M_i = 6.9 \cdot 10^{-4} M)$

* computer generated plot



LUM(8) VS TIME (Q=0.9993)

228 ZONES - MASS = 2.984D 34 GM - 8 CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY

FIGURE 6.21

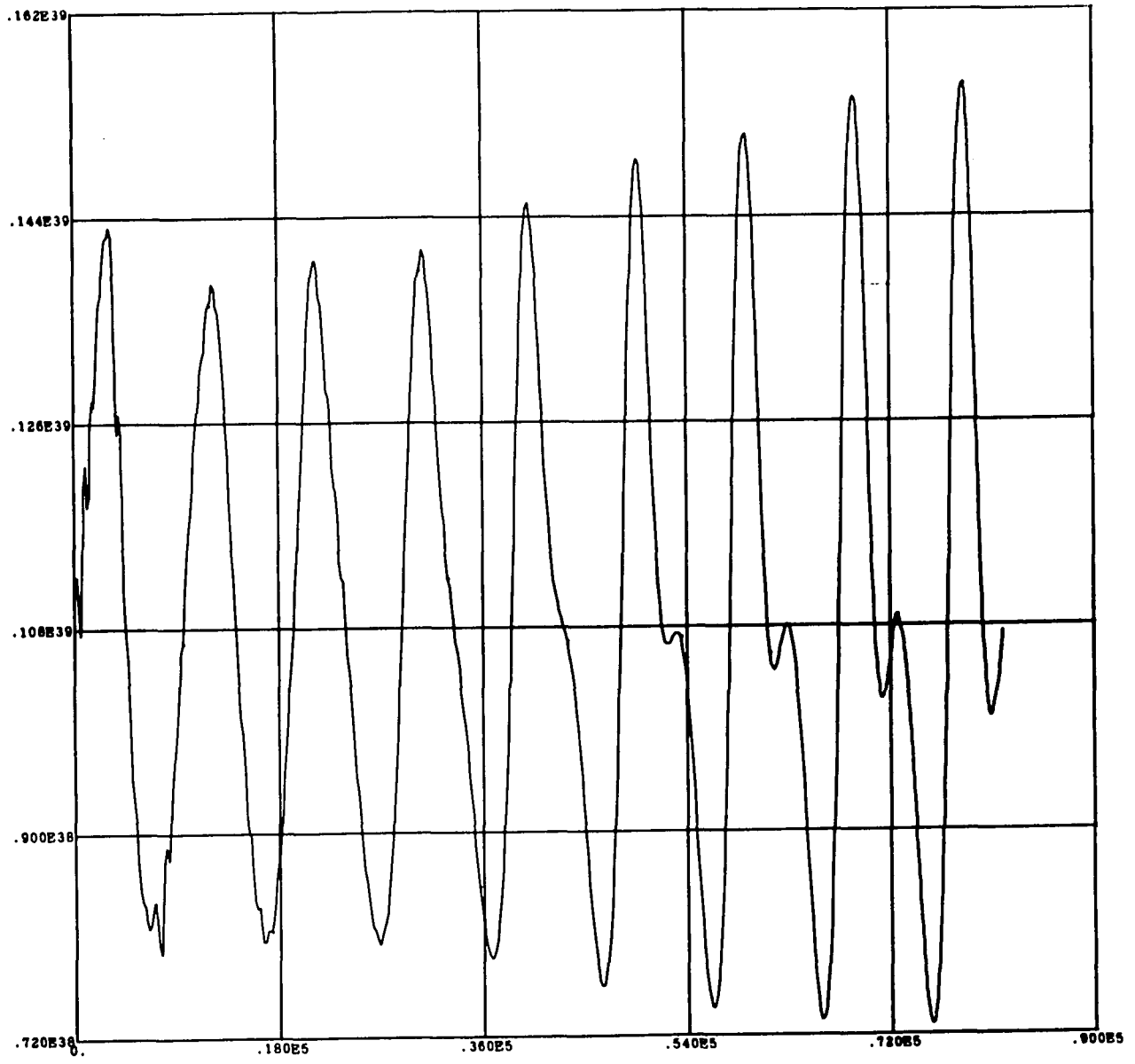
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 28

$(M-M_i = 4.1 \cdot 10^{-3} M)$

* computer generated plot



LUM(9) VS TIME (Q=0.9959)

228 ZONES - MASS = 2.984D 34 GM - # CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY

FIGURE 6.22

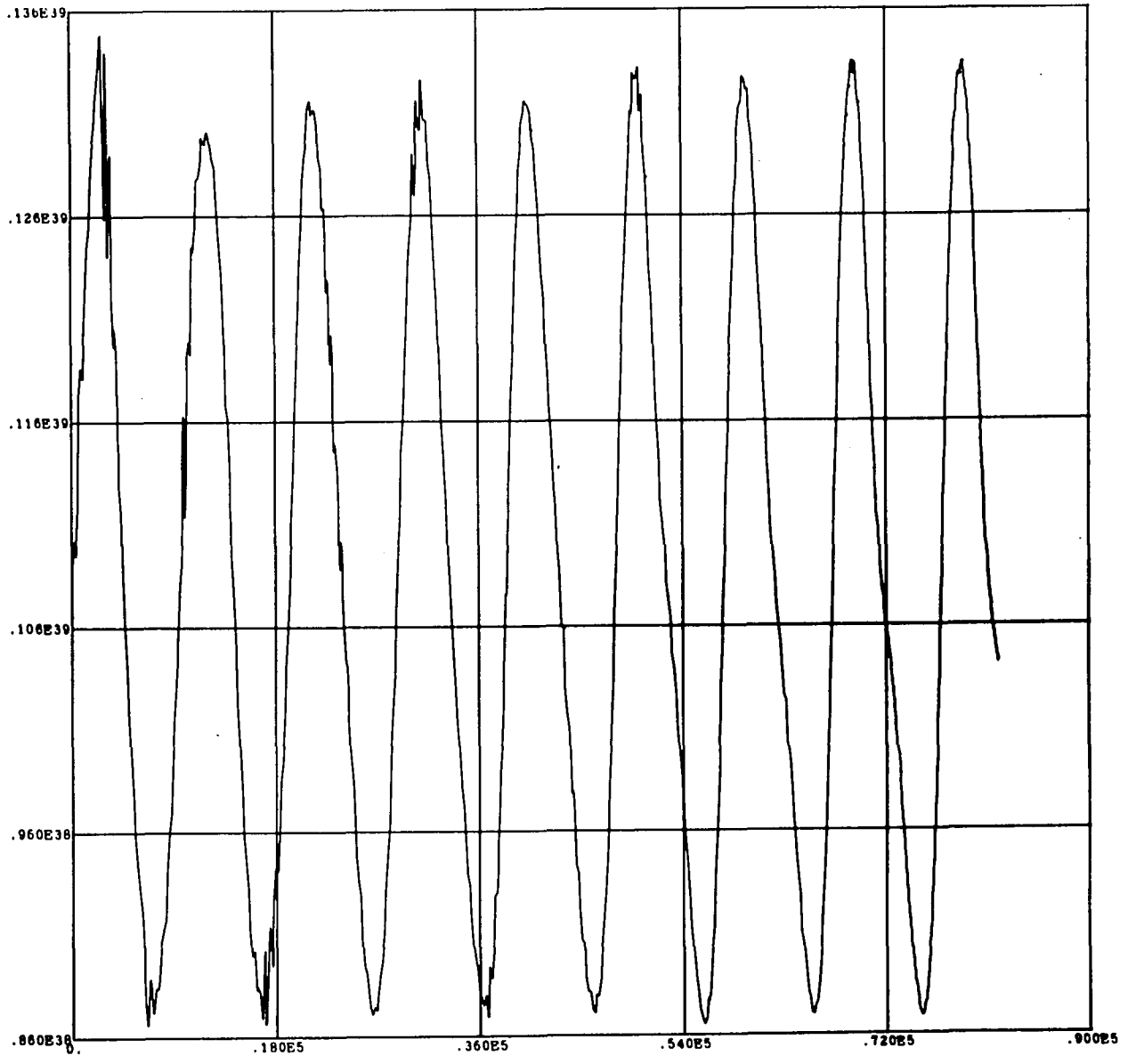
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 32

$(M-M_i = 2.0 \cdot 10^{-2} M)$

* computer generated plot



LUM(10) VS TIME (Q=0.9600)

228 ZONES - MASS = 2.984D 34 GM - A CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 - COMPLETE OPACITY

FIGURE 6.23

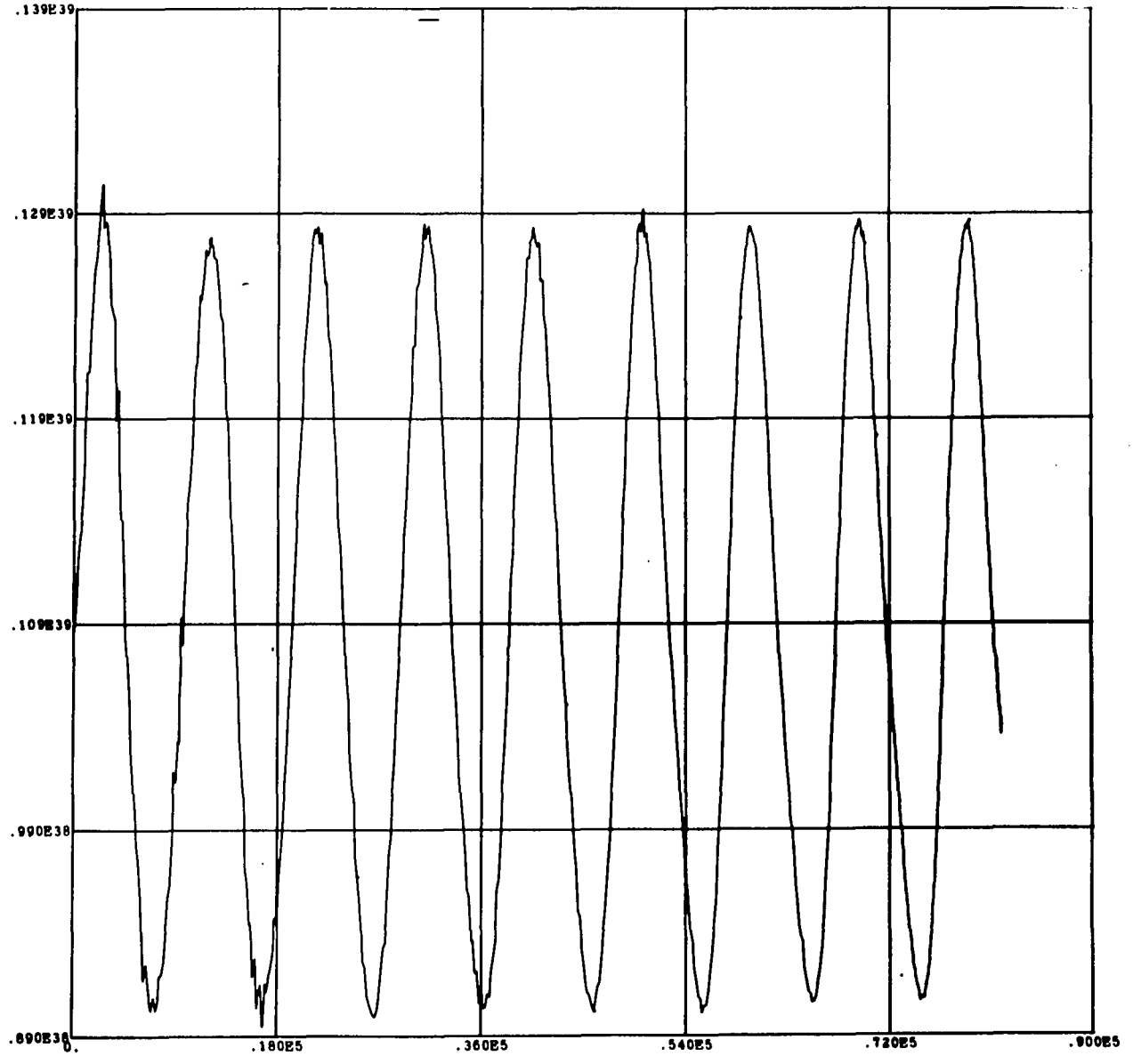
β CEPHEI MODEL # 16 *

LUMINOSITY AT SUBSURFACE

ZONE # 36

$(M-M_i = 4.0 \cdot 10^{-2} M)$

* computer generated plot



LUM(11) VS TIME (Q=0.9600)

228 ZONES - MASS = 2.984D 34 GM - 8 CEPHEID MODEL NUMBER 16 - VISCOSITY COEF = 5.0D 00 -. COMPLETE OPACITY

FIGURE 6.24

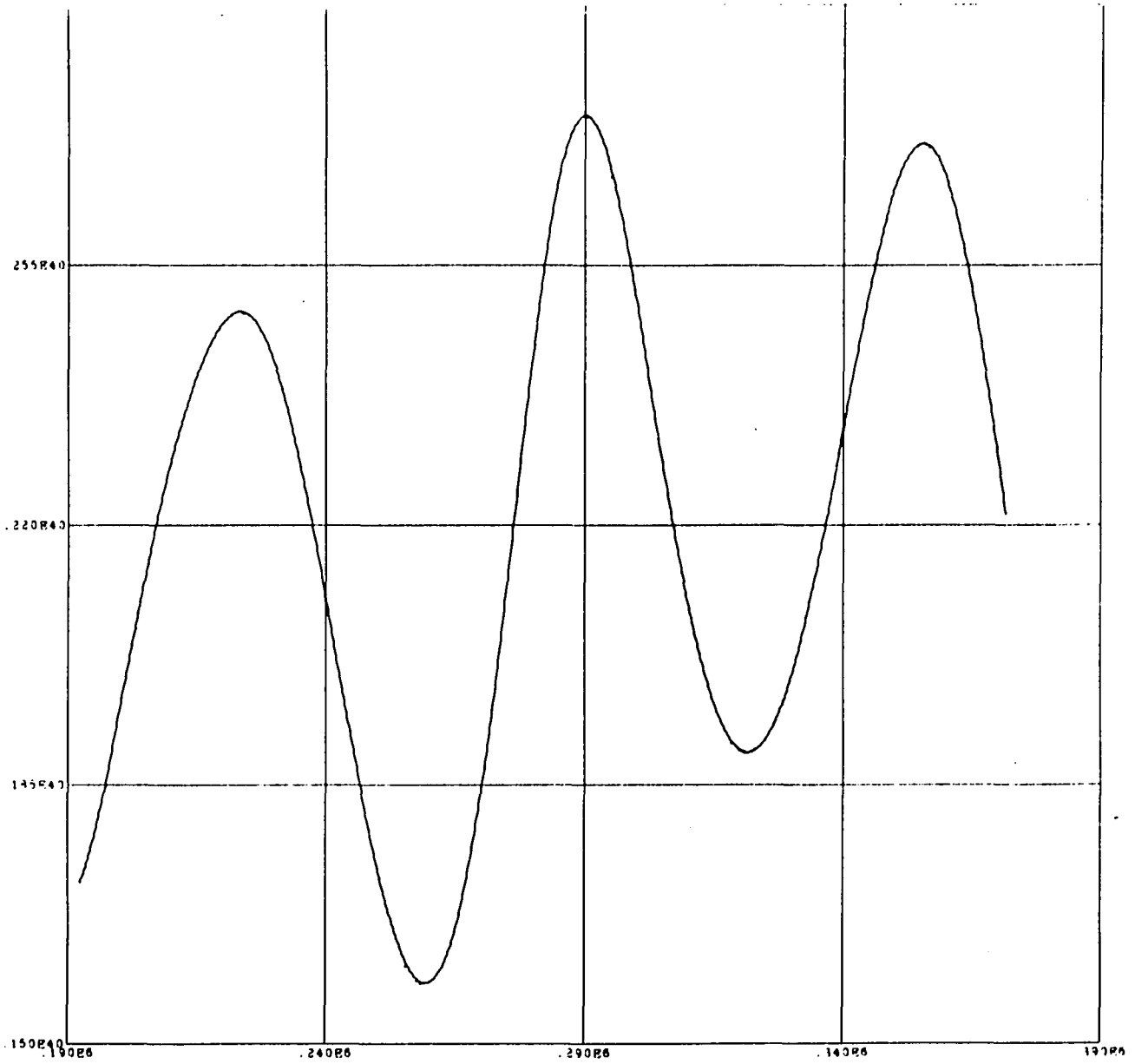
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 1, PLOT II

$(M-M_i = 8.0 \cdot 10^{-8} M)$

* computer generated plot



LUM(2) VS TIME (Q=1.0000)

221 ZONES - MASS = 2.964E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY CONF = 5.0E 00 - COMPLETE OPACITY 09/25/70

FIGURE 6.25

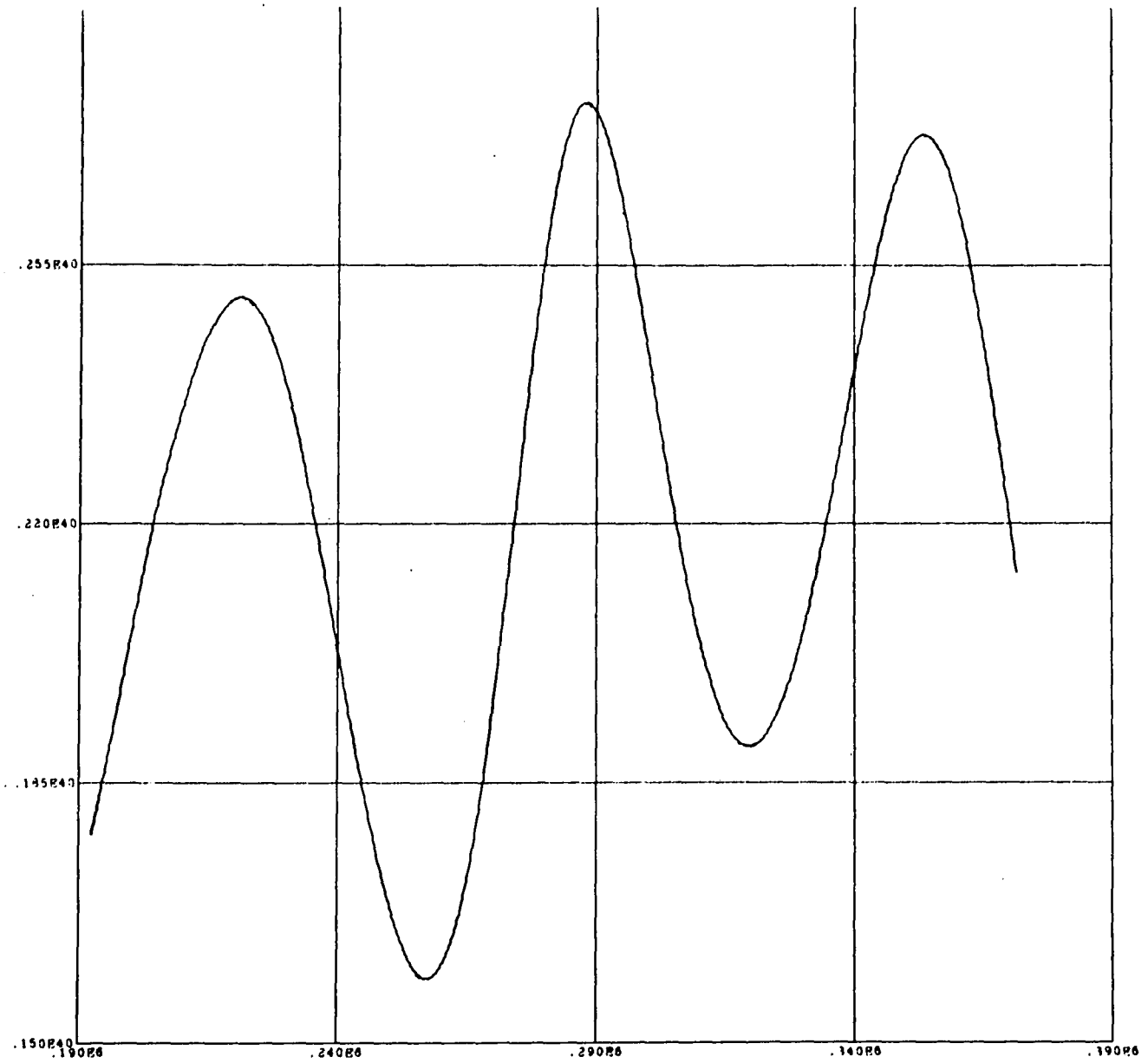
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 5, PLOT II

$(M-M_i = 1.2 \cdot 10^{-6} M)$

* computer generated plot



LUM(6) VS TIME (Q=1.0000)

223 ZONES - MASS = 2.984E 34 GM - 8 CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 06/26/70

FIGURE 6.26

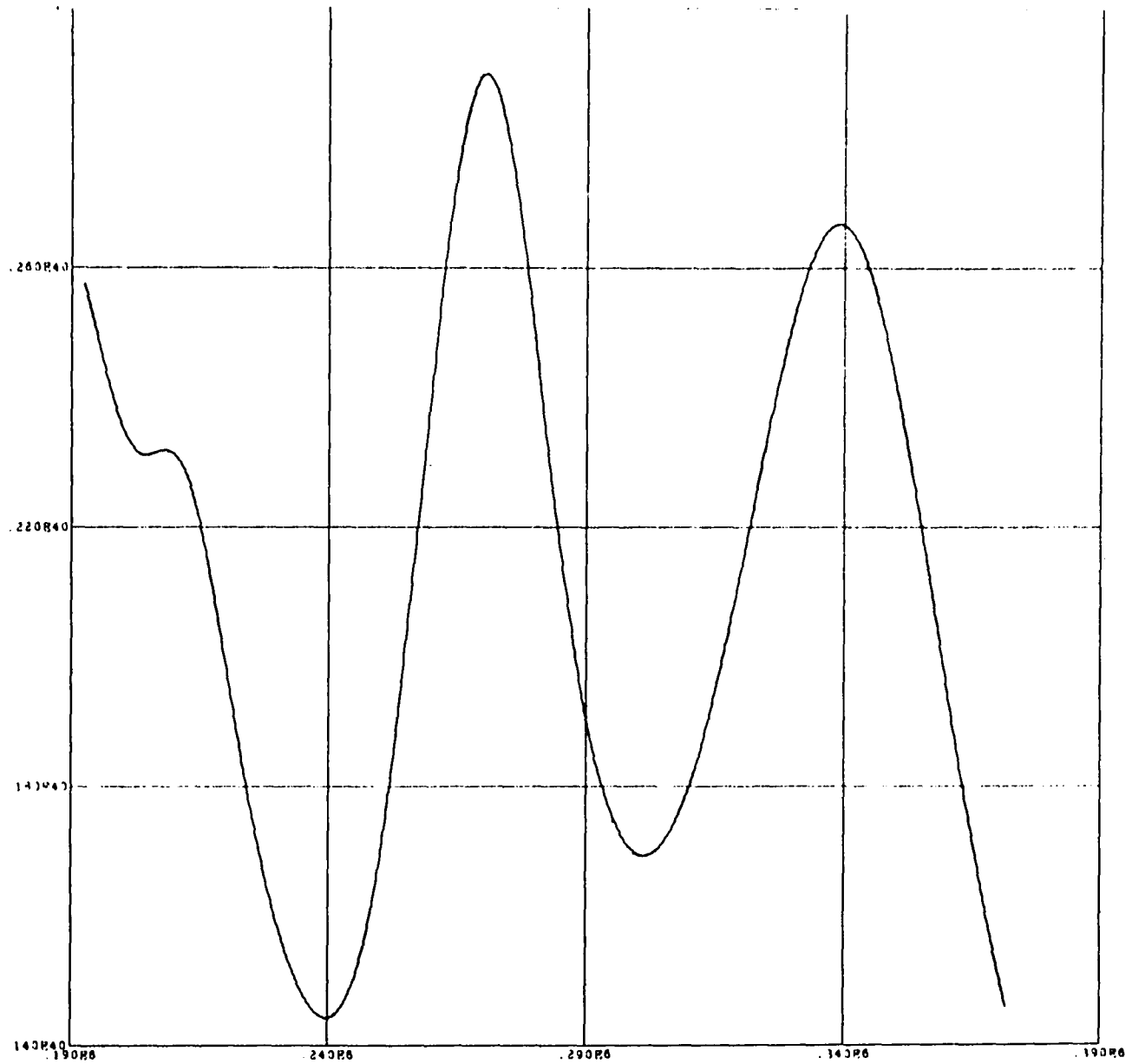
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 8, PLOT II

$(M-M_i = 4.9 \cdot 10^{-6} M)$

* computer generated plot



LUM(8) VS TIME (Q=1.0000)

221 ZONES - MASS = 2.964R 34 GM - 6 CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.08 00 - COMPLETE OPACITY 08/28/70

FIGURE 6.27

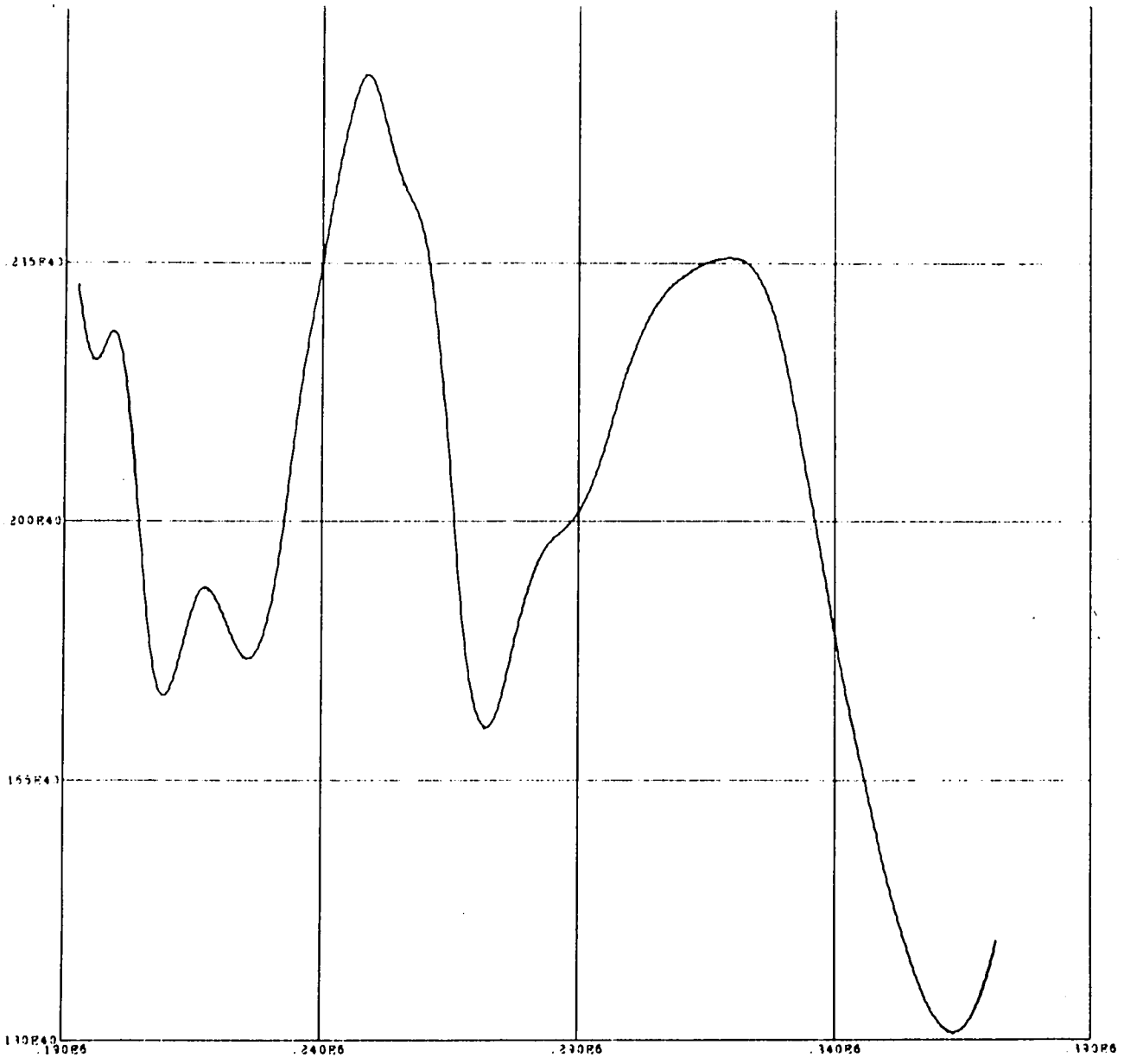
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 10, PLOT II

$(M-M_i = 1.2 \cdot 10^{-5} M)$

* computer generated plot



LUM(9) VS TIME (Q=1.0000)

223 ZONES - MASS = 2.9848 34 GM - 8 CEPHRID MODEL NUMBER 12 - VISCOSITY COEF = 5.08 00 - COMPLETE OPACITY 08/26/70

FIGURE 6.28

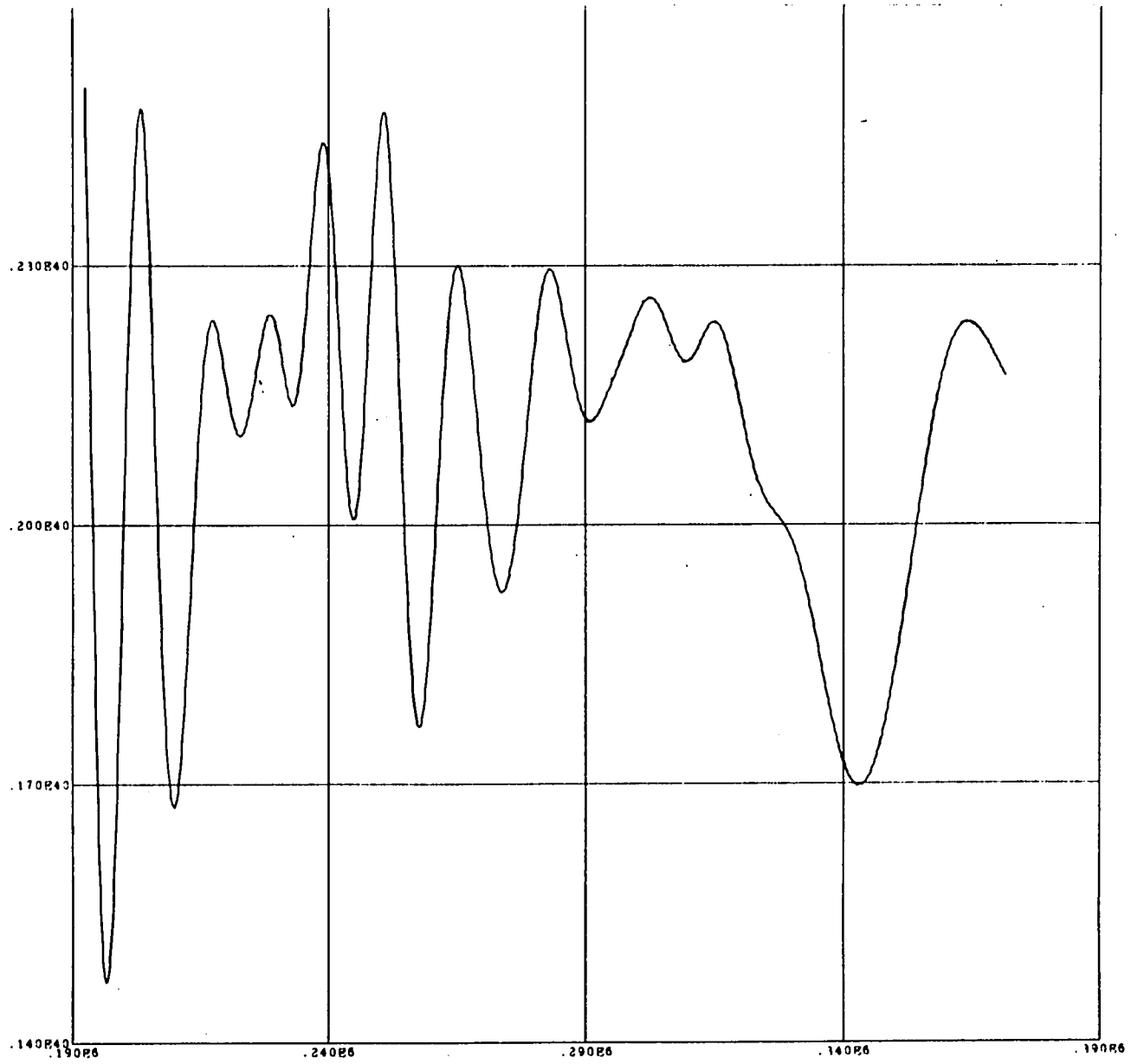
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 12, PLOT II

$(M-M_i = 3.0 \cdot 10^{-5} M)$

* computer generated plot



LUM(10) VS TIME (Q=1.0000)

221 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLRTR OPACITY 08/28/70

FIGURE 6.29

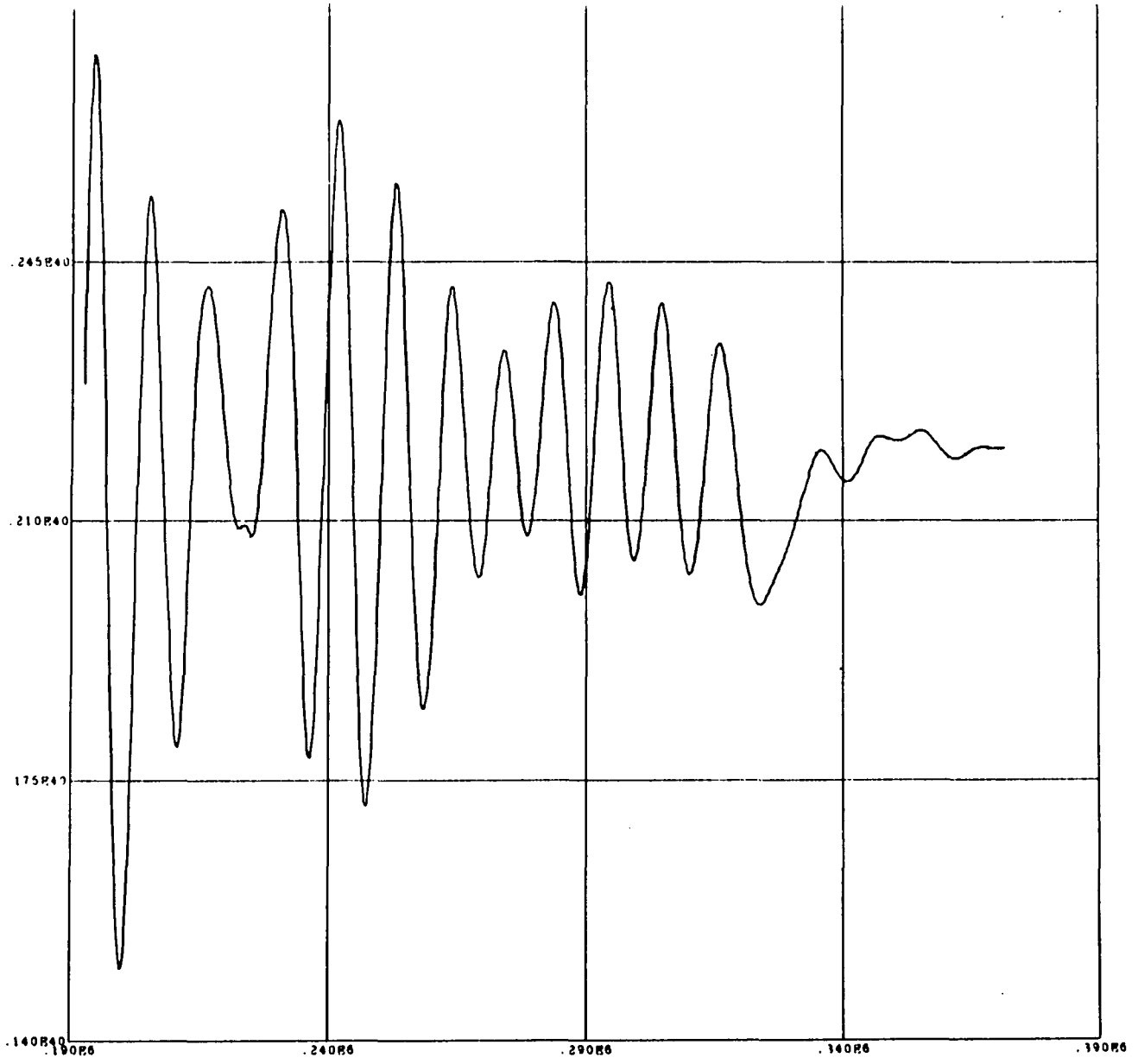
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 14, PLOT II

$(M-M_i = 7.4 \cdot 10^{-5} M)$

* computer generated plot



LUM(11) VS TIME (Q=0.9999)

223 ZONES - MASS = 2.984E 14 GM - 8 CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 09/28/70

FIGURE 6.30

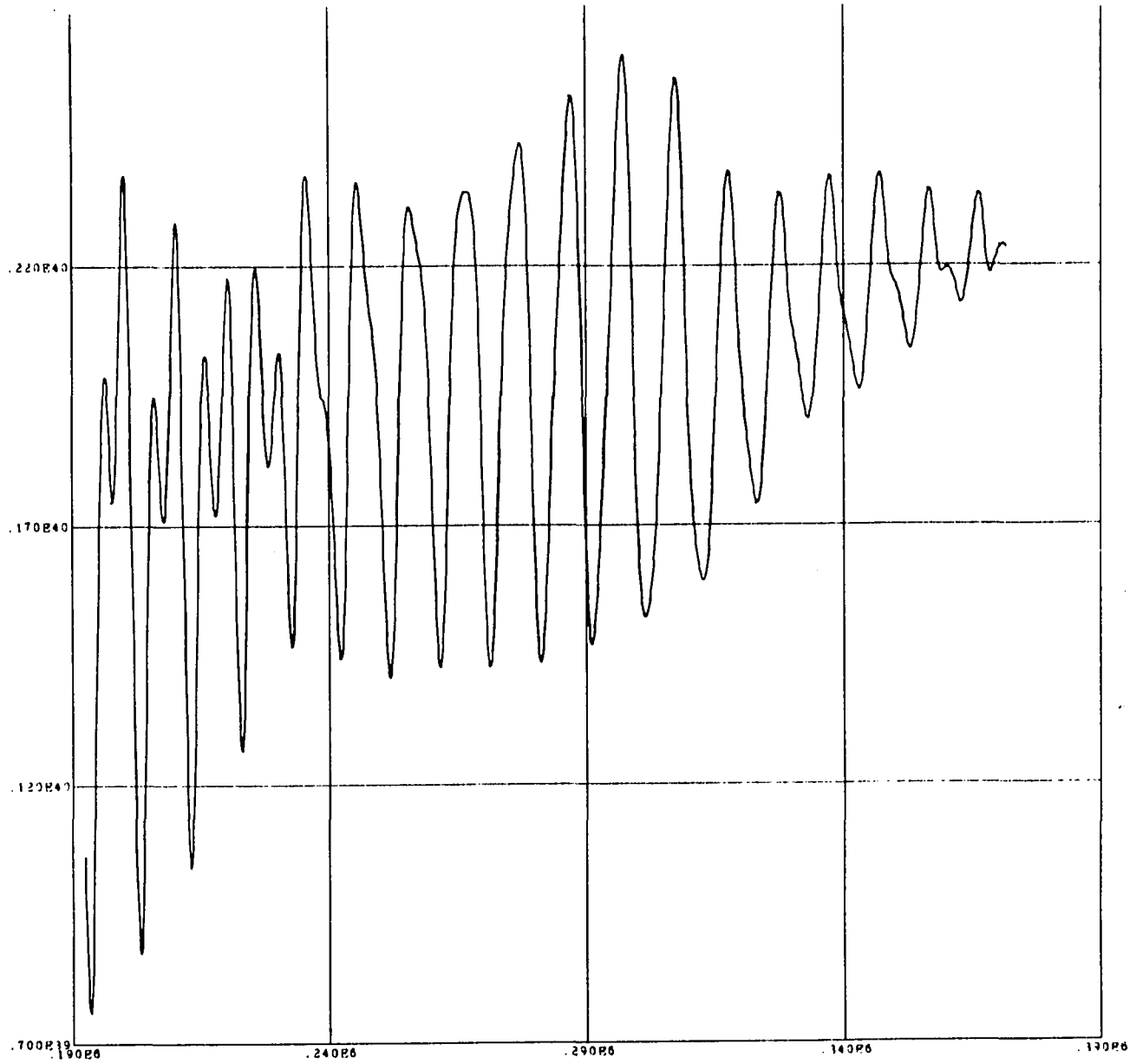
β CEPHEI MODEL # 12 *

LUMINOSITY AT SUBSURFACE

ZONE # 16, PLOT II

$(M-M_i = 1.4 \cdot 10^{-4} M)$

* computer generated plot



LUM(12) VS TIME (Q=0.9998)

223 ZONES - MASS = 2.984E 34 GM - A CEPHEID MODEL NUMBER 12 - VISCOSITY COEF = 5.0E 00 - COMPLETE OPACITY 08/28/70

CHAPTER 7

CONCLUSIONS

Perhaps the most fruitful consequences of our investigations come from the qualitative agreement between certain aspects of some of the inhomogenous models described in section (6.4) and observations of astronomical objects. Since these models are based on the 15 $M_\odot$ β Cephei model of Stothers and Simon (1969), it is not surprising that they agree with β Cephei observations in the same areas that Stothers and Simon's linearized treatment does. However, our treatment lends additional credence to the correlations between the model and β Cephei stars. Stothers (1965) as well as Stothers and Simon (1969) point out the evidence of observers such as Struve, implying ejection of shells of matter from β Cephei stars. As will be indicated shortly, this phenomenon is a feature of the more physically tenable treatments of our model. If we suppress mass loss by lowering the magnitude of the initial velocity distribution, raising the value of the artificial viscosity coefficient C_q, or omitting the thermal viscous term; we achieve a two component radial velocity curve very similar to those observed for β Cephei stars. We also achieve a two component light curve that is reminiscent of β Cephei light curves except for a phase reversal of the minor and major components. It must be noted that good observations of β Cephei stars are rather sparse, and hence any comparison between theory and observation is

highly qualitative. Nevertheless, there are some well established features of observation that correlate strikingly well with those of our suppressed mass-loss models. Both theory and observation indicate zero phase shift between each component of the light curve and the corresponding component of the radial velocity curve. It is of considerable interest that a strong correlation exists between several aspects of our β Cephei model and observations of Wolf-Rayet stars. If we consider mass loss as occurring when a mass shell becomes optically thin, we find that our model is losing at least $5 \cdot 10^{-5}$ $M_\odot$, but probably no more than 10^{-4} $M_\odot$, per year. This is within the order of magnitude accuracy of the observationally estimated mass loss from the rapidly expanding envelope surrounding Wolf-Rayet stars made recently by Underhill (1969). Furthermore, we project estimated cloud velocities of $0.8 \cdot 10^3$ km/sec as compared to Underhill's estimate of $\sim 10^3$ km/sec. It has been suggested (Westerlund and Henize, 1963; Johnson and Hogg, 1965; Smith, 1967; Simon and Stothers, 1970) that the nebula surrounding some Wolf-Rayet stars may be more massive than 7 $M_\odot$, structurally consisting of many thin shells. If we follow the development of our most finely zoned model, we find that initially a mass shell consisting of 3 or 4 zones is lost, followed some time later by a loss of the next 2 or 3 mass zones. Thus, rather than losing mass continuously, our models appear to be ejecting shells of from $5 \cdot 10^{-8}$ $M_\odot$

to $5 \cdot 10^{-7}$ $M_\odot$. It must be noted that imposition of the photospheric boundary conditions, as discussed in section (3.5), introduces inaccuracies near the surface of a model in which mass loss occurs, after such a model has lost a couple of shells. In the past Stothers (1965) has contended that Wolf-Rayet stars form an "apparent extension" of the β Cephei strip on the H-R diagram. This, when considered in light of the above analysis, leads to the conjecture that some Wolf-Rayet stars may either represent an early evolutionary state of β Cephei stars, or may evolve from similar but somewhat more massive objects.

It is appropriate to summarize the nature and importance of the computational techniques devised for this study. A computer code has been developed that synthesizes the full non-linear dynamic behavior of a spherically symmetric fluid distribution. It takes into account gravitational effects as well as energy transfer and generation. Although the code was designed with an inviscid fluid in mind it may be adapted to treat viscous effects. The modular construction of the code through its subroutine structure (see APPENDIX F) allows for relatively simple modification, and thus makes it applicable to a large class of astrophysical problems. The code allows us to follow, with high resolution, the time development of all pertinent physical parameters

throughout a stellar model. We have constructed subsidiary routines which, when used in conjunction with the main code, produce graphical displays of the light and velocity curves during any time interval specified, as well as radius amplitude curves for the surface and any specified subsurface zones. Still other routines produce plots of the spatial distribution of the physically interesting variables at any point during the time development. The code calculates many derived quantities such as the spatially integrated kinetic, gravitational and nuclear generated energies at each time step. These are outputed in a form suitable for processing by additional routines. One of the most important aspects of our computer code is its ability to follow the occurrence and development of mass loss from stellar models.

The most casual consideration leads to a multitude of interpretations of the data presented at the beginning of this section. The conjectures that arise are too numerous to enumerate. However, for any of these conjectures to bear fruit, future investigations must proceed along two lines. Firstly, we must obtain more and better observations of both the Wolf-Rayet objects and β Cephei stars. Secondly we must improve and expand our theoretical investigations in several areas. A better treatment of the stellar boundary is necessary to give a

more accurate picture of the mass loss mechanism. A more realistic composition gradient, as well as a more thorough treatment of energy transport, is required to enhance the model's credibility. Finally, models of other masses and composition should be constructed. There is no guarantee that such a program will provide an explanation for the behavior of some identifiable class of stellar objects, but, if not, it will most certainly aid in determining the most promising directions for further research.

APPENDICES

APPENDIX A

LINEARIZED PULSATION THEORY

(for small adiabatic perturbations)

Let: $r \longrightarrow r' = r + \delta r = (1 + \frac{\delta r}{r})r$

correspondingly $\rho' = (1 + \frac{\delta\rho}{\rho})\rho$

$$P' = (1 + \frac{\delta P}{P})P$$

$$T' = (1 + \frac{\delta T}{T})T$$

$$L' = (1 + \frac{\delta L}{L})L$$

Adopting the conventions of Schwarzchild, 1958, pp 103 and 109,

$$M_r = qM, \qquad r = xR, \qquad v = -\frac{r}{P}\frac{dP}{dr}, \qquad n + 1 = \frac{T}{P}\frac{dP}{dT}$$

The Equation of Continuity must hold for the perturbed star, therefore:

$$\frac{dr'}{dM_r} = \frac{1}{4\pi r'^2 \rho'}$$

but: $\frac{dr'}{dM_r} = \frac{d}{dM_r}[(1 + \frac{\delta r}{r})r] = (1 + \frac{\delta r}{r})\frac{dr}{dM_r} + r\frac{d}{dM_r}(\frac{\delta r}{r})$

from the continuity condition of the unperturbed model:

$$\frac{dr}{dM_r} = \frac{1}{4\pi r^2 \rho} \quad \text{or} \quad \frac{d}{dM_r} = \frac{1}{4\pi r^2 \rho} \frac{d}{dr}$$

thus: $$\frac{dr'}{dM_r} = \frac{1}{4\pi r^2 \rho}\left(1 + \frac{\delta r}{r}\right) + \frac{1}{4\pi r^2 \rho} r \frac{d}{dr}\left(\frac{\delta r}{r}\right) =$$

$$\frac{1}{4\pi r^2 \rho}\left[1 + \frac{\delta r}{r} + x\frac{d}{dx}\left(\frac{\delta r}{r}\right)\right]$$

A simple Taylor's Series expansion, ignoring terms greater than first order, leads to:

$$\frac{1}{4\pi r'^2 \rho'} = \frac{1}{4\pi r^2 \rho}\left(1 - 2\frac{\delta r}{r} - \frac{\delta \rho}{\rho}\right)$$

Therefore: $$1 + \frac{\delta r}{r} + x\frac{d}{dx}\left(\frac{\delta r}{r}\right) = 1 - 2\frac{\delta r}{r} - \frac{\delta \rho}{\rho}$$

or $$x\frac{d}{dx}\left(\frac{\delta r}{r}\right) = -\frac{\delta \rho}{\rho} - 3\frac{\delta r}{r}$$

This corresponds to Equation (1) of Schwarzchild and Härm (1959).

Conservation of momentum dictates

$$\frac{dP'}{dr'} = -\rho' \frac{M_r G}{r'^2} - \rho' \frac{d^2 r'}{dt^2}$$

$$\frac{d}{dr'}\left[\left(1+\frac{\delta P}{P}\right)P\right] = \rho\left(1+\frac{\delta\rho}{\rho}\right)\frac{M_r G}{r^2\left(1+\frac{\delta r}{r}\right)^2}$$

$$- \rho\left(1+\frac{\delta\rho}{\rho}\right)\frac{d^2}{dt^2}\left[r\left(1+\frac{\delta r}{r}\right)\right]$$

$$P\frac{d}{dr'}\left(\frac{\delta P}{P}\right) + \left(1+\frac{\delta P}{P}\right)\frac{dP}{dr'} \simeq -\frac{M_r G}{r^2}\left(1+\frac{\delta\rho}{\rho}-2\frac{\delta r}{r}\right)$$

$$- \rho r\frac{d^2}{dt^2}\left(\frac{\delta r}{r}\right)\left(1+\frac{\delta\rho}{\rho}\right)$$

[NOTE: $\frac{dr}{dt} = 0$ by definition]

$$dr' = \left(1+\frac{\delta r}{r}\right)dr + r\cdot d\left(\frac{\delta r}{r}\right)$$

$$= dr\left[1+\frac{\delta r}{r}+x\frac{d}{dx}\left(\frac{\delta r}{r}\right)\right]$$

using the previously derived value for $x\frac{d}{dx}\left(\frac{\delta r}{r}\right)$

we find

$$dr' = dr\left(1-\frac{\delta\rho}{\rho}-2\frac{\delta r}{r}\right)$$

therefore, to first order

$$P\frac{d}{dr}\left(\frac{\delta P}{P}\right) + \left(1+\frac{\delta P}{P}\right)\frac{dP}{dr} \simeq -\rho\frac{M_r G}{r^2}\left(1-4\frac{\delta r}{r}\right)$$

$$- \rho r\left(1-2\frac{\delta r}{r}\right)\frac{d^2}{dt^2}\left(\frac{\delta r}{r}\right)$$

Noting that $\frac{dP}{dr} = -\rho\frac{M_r G}{r^2}$ for the unperturbed model, and rearranging terms, we find:

$$r(1 - 2\frac{\delta r}{r})\frac{d^2}{dt^2}(\frac{\delta r}{r}) = -P\frac{d}{dr}(\frac{\delta P}{P}) + (1 + \frac{\delta P}{P})\rho\frac{M_r G}{r^2} - (1 - 4\frac{\delta r}{r})\rho\frac{M_r G}{r^2}$$

$$\rho r(1 - 2\frac{\delta r}{r})\frac{d^2}{dt^2}(\frac{\delta r}{r}) = -\frac{P}{r}x\frac{d}{dx}(\frac{\delta P}{P}) + \rho\frac{M_r G}{r^2}(\frac{\delta P}{P} + 4\frac{\delta r}{r})$$

finally, again to first order:

$$\frac{d^2}{dt^2}(\frac{\delta r}{r}) = \frac{M_r G}{r^3}(\frac{\delta P}{P} + 6\frac{\delta r}{r}) - \frac{P}{\rho r^2}(1 + 2\frac{\delta r}{r})x\frac{d}{dx}(\frac{\delta P}{P})$$

If we assume: $\delta r(r,t) \equiv \delta_r r(r)\cdot\delta_t r(t)$ we find that above equation separable in the variables x, t, since:

$$\frac{\delta P}{P} = \gamma\frac{\delta\rho}{\rho} \qquad \text{(adiabatic perturbation)}$$

and $\frac{\delta\rho}{\rho} = -3\frac{\delta r}{r} - x\frac{d}{dx}(\frac{\delta r}{r}) = \delta_t r\cdot f(r,\delta_r r)$

Therefore we may conclude that if we assume a time dependency of the form: $\delta_t r = \exp(i\omega_o t)$,

$$\frac{d^2}{dt^2}\left(\frac{\delta r}{r}\right) = -\omega_o^2 \frac{\delta r}{r}$$

where: $\omega_o = \frac{2\pi}{\tau_o}$ and τ_o is the period of motion.

Thus: $\omega_o^2 \frac{\delta r}{r} = \frac{P}{\rho r^2}\left(1 + 2\frac{\delta r}{r}\right) x \frac{d}{dx}\left(\frac{\delta P}{P}\right) - \frac{M_r G}{r^3}\left(\frac{\delta P}{P} + 6\frac{\delta r}{r}\right)$

and once again to first order,

$$d\frac{d}{dx}\left(\frac{\delta P}{P}\right) = \frac{M_r G \rho}{rP}\left(\frac{\delta P}{P} + 4\,\frac{\delta r}{r}\right) + \frac{\rho r^2}{P}\omega_o^2 \frac{\delta r}{r}$$

but $\frac{M_r G \rho}{rP} = -\frac{r}{P}\frac{dP}{dr} \equiv V$

$$x\frac{d}{dx}\left(\frac{\delta P}{P}\right) = V\left(\frac{\delta P}{P} + 4\frac{\delta r}{r}\right) + V\frac{r^3\omega_o^2}{M_r G}\frac{\delta r}{r}$$

$$x\frac{d}{dx}\left(\frac{\delta P}{P}\right) = V\frac{\delta P}{P} + V\left[4 + \frac{x^3}{q}\left(\frac{R^3\omega_o^2}{MG}\right)\right]\frac{\delta r}{r}$$

If we now substitute: $\omega^2 \equiv \omega_o^2 \frac{R^3}{GM}$

where ω represents a non-dimensional frequency, we get Equation (2) of Schwarzchild and Härm (1959), except for a sign difference between the 4 and $\omega^2\frac{x^3}{q}$ terms. This difference of sign was also noted by Simon (1970a) and attributed to a typographical error.

We may summarize as follows:

$$x\frac{d}{dx}\left(\frac{\delta r}{r}\right) = -\frac{\delta\rho}{\rho} - 3\frac{\delta r}{r} \quad ,$$

$$x\frac{d}{dx}\left(\frac{\delta P}{P}\right) = V\frac{\delta P}{P} + V\left(4 + \omega^2\frac{x^3}{q}\right)\frac{\delta r}{r} \quad ,$$

$$\frac{\delta P}{P} = \gamma \frac{\delta \rho}{\rho} \quad \text{with} \quad \gamma = \beta + \frac{2}{3} \frac{(4-3\beta)^2}{\beta+8(1-\beta)} \quad .$$

The non dimensional frequency ω is defined by

$$\omega^2 = \left(\frac{2\pi}{\text{Period}}\right)^2 \frac{R^3}{GM}$$

We may find the amplitude of the luminosity perturbation by assuming that the perturbed envelope is still in radiative equilibrium.

$$L_r' = - 4\pi r'^2 \frac{4ac}{3\kappa} \frac{T'^3}{\rho'} \frac{dT'}{dr'}$$

but: $$L_r = - 4\pi r^2 \frac{4ac}{3\kappa} \frac{T^3}{\rho} \frac{dT}{dr}$$

(Note that the perturbed and unperturbed opacities are identical because we have assumed that the opacity is due solely to electron scattering)

Dividing the foregoing equation into the one preceding, and expanding all powers to first order:

$$1 + \frac{\delta L_r}{L_r} = \left(1 + 2\frac{\delta r}{r}\right)\left(1 + 3\frac{\delta T}{T}\right)\left(1 - \frac{\delta \rho}{\rho}\right)\frac{dr}{dr'} \frac{dT'}{dT}$$

but we have shown:

$$\frac{dr}{dr'} = \left(1 + \frac{\delta \rho}{\rho} + 2\frac{\delta r}{r}\right)$$

Therefore:

$$1 + \frac{\delta L_r}{L_r} = (1 + 3\frac{\delta T}{T} + 4\frac{\delta r}{r})\ \frac{dT'}{dT}$$

however:

$$dT' = (1 + \frac{\delta T}{T})dT + T \cdot d(\frac{\delta T}{T})$$

$$= [(1 + \frac{\delta T}{T} + \frac{T}{r}\frac{dr}{dT} \times \frac{d}{dx}(\frac{\delta T}{T})]\ dT$$

and noting that:

$$\frac{T}{r}\frac{dr}{dT} = \frac{P}{r}\frac{dr}{dP}(\frac{T}{P}\frac{dP}{dT}) = -\frac{n+1}{v}$$

we find that:

$$\frac{dT'}{dT} = 1 + \frac{\delta T}{T} - \frac{n+1}{v} \times \frac{d}{dx}(\frac{\delta T}{T})$$

finally:

$$1 + \frac{\delta L_r}{L_r} = (1 + 3\frac{\delta T}{T} + 4\frac{\delta r}{r})(1 - \frac{n+1}{v} \times \frac{d}{dx}(\frac{\delta T}{T}) + \frac{\delta T}{T})$$

giving us Equation (5) of Schwarzchild and Härm (1959)

$$\frac{\delta L_r}{L_r} = -\frac{n+1}{v} \times \frac{d}{dx}(\frac{\delta T}{T}) + 4\frac{\delta T}{T} + 4\frac{\delta r}{r}$$

the constriction of the perturbation being adiabatic imposes the following condition:

$$\frac{\delta T}{T} = \theta\frac{\delta \rho}{\rho} \qquad \text{and} \qquad \theta = \frac{2}{3}\frac{4-3\beta}{\beta+8(1-\beta)}$$

This completes the dynamic analysis of the pulsational amplitudes. The linearization of the theory results in a sinusoidal time development with the amplitude time dependence eliminated. In order to discover, whether the pulsations are growing or decaying we must revert to energy considerations.

The pulsations will grow if the net energy generation per period exceeds that radiated away. A complete exposition on this aspect of the pulsational stability may be found in Schwarzchild and Harm (1959).

APPENDIX B
ENTROPY OF AN IDEAL GAS WITH RADIATION

$$TdS = dU + PdV$$

$$= dU - \frac{P}{\rho^2}\, d\rho$$

$$U = \frac{3}{2}\frac{k}{\mu H} T + \frac{a}{\rho} T^4 \text{ for ideal gas and radiation}$$

$$dU = \frac{3}{2}\frac{k}{\mu H} dT + \frac{4a}{\rho} T^3 d\,T - \frac{a}{\rho^2} T^4 d\rho$$

$$dS = \left\{\frac{3}{2}\frac{k}{\mu H}\frac{1}{T} + \frac{4a}{\rho} T^2\right\} dT - \left\{\frac{aT^3}{\rho^2} + \frac{P}{\rho^2 T}\right\} d\rho$$

since: $P = \frac{k}{\mu H}\rho T + \frac{a}{3} T^4$

$$dS = \left\{\frac{3}{2}\frac{k}{\mu H}\frac{1}{T} + \frac{4a}{\rho} T^2\right\} dT - \left\{\frac{k}{\mu H \rho} + \frac{4}{3}\frac{aT^3}{\rho^2}\right\} d\rho$$

$$S = \int \frac{\partial S}{\partial T}\, dT + f_1(\rho) + \text{const}_1$$

$$S = \int \frac{\partial S}{\partial \rho} d\rho + f_2(T) + \text{const}_2$$

$$S = \frac{3}{2} \frac{k}{\mu H} \ln T + \frac{4}{3} \frac{aT^3}{\rho} + f_1(\rho) + \text{const}_1$$

$$S = -\frac{k}{\mu H} \ln\rho + \frac{4}{3} \frac{aT^3}{\rho} + f_2(T) + \text{const}_2$$

$$\therefore \quad f_1(\rho) = -\frac{k}{\mu H} \ln\rho$$

$$f_2(T) = \frac{3}{2} \frac{k}{\mu H} \ln T$$

$$\text{const}_1 = \text{const}_2 \equiv \text{const}$$

$$S = \frac{3}{2} \frac{k}{\mu H} \ln T - \frac{k}{\mu H} \ln\rho + \frac{4}{3} \frac{aT^3}{\rho} + \text{const}$$

$$= \frac{k}{\mu H} \ln\left(\frac{T^{3/2}}{\rho}\right) + \frac{4}{3} \frac{aT^3}{\rho} + \text{const}$$

APPENDIX C

DETERMINATION OF ATMOSPHERIC MASS

We may define an optical depth, τ, in terms of the opacity as follows:

$$d\tau \equiv -\kappa\rho dr = -\frac{\kappa}{4\pi r^2}\, dm$$

leading to the relation

$$dm = -\frac{4\pi r^2}{\kappa} d\tau$$

From the theory of radiative transfer in stellar atmospheres, we find, to first approximation, that the optical depth of the photosphere is $\tau_R = 2/3$ (Schwarzchild, 1959, p. 89);

$$m_{atmo} = -\int_{2/3}^{0} \frac{4\pi r^2}{\kappa}\, d\tau = \int_{0}^{\frac{2}{3}} \frac{4\pi r^2}{\kappa}\, d\tau$$

For electron scattering $\kappa \equiv \kappa_e$ is a constant. For the more complete opacities used in computing some of the models considered in this paper, it was found that opacity varied from its photospheric value by less than 25 per cent throughout the atmosphere. Thus we may consider $\kappa = \bar{\kappa} \sim \kappa_R$,

leading to

$$m_{atmo} \simeq \frac{4\pi}{\kappa_R} \int_0^{\frac{2}{3}} r^2 d\tau$$

For all the models considered r is less than 3 parts in 10^4 greater than R, even at several density scale heights into the atmosphere. Therefore, we find

$$m_{atmo} \simeq \frac{8\pi R^2}{3\kappa_R}$$

APPENDIX D

NUMERICAL APPROXIMATIONS OF DERIVATIVES

CENTERED DERIVATIVES

$y(x_o) \equiv y_o$

$\left.\frac{dy}{dx}\right|_{x_o} \equiv y'_o$

$\left.\frac{d^2y}{dx^2}\right|_{x_o} \equiv y''_o$

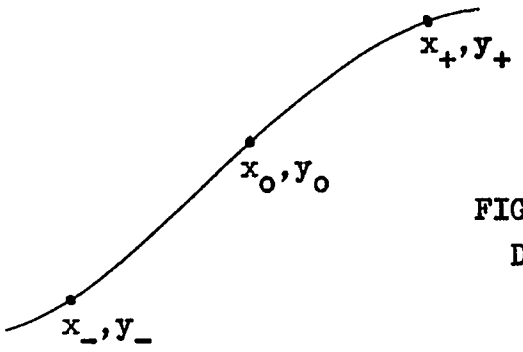


FIGURE D.1

Consider expanding the function $y(x)$ about x_o to 2nd order to obtain approximate values at x_- and x_+.

$$y_+ = y_o + y'_o(x_+ - x_o) + y''_o\frac{(x_+ - x_o)^2}{2} + \ldots$$

$$y_- = y_o + y'_o(x_- - x_o) + y''_o\frac{(x_- - x_o)^2}{2} + \ldots$$

Subtracting the last equation from the one preceding,

$$y_+ - y_- = y'_o(x_+ - x_-) + \frac{y''_o}{2}[(x_+ - x_o)^2 - (x_- - x_o)^2] + \ldots$$

but:

$$(x_+ - x_o)^2 = x_+^2 - 2x_+x_o + x_o^2$$

$$(x_- - x_o)^2 = x_-^2 - 2x_-x_o + x_o^2$$

Therefore:

$$(x_+ - x_o)^2 - (x_- - x_o)^2 = x_+^2 - x_-^2 - 2(x_+ - x_-)x_o$$

$$= (x_+ - x_-)[(x_+ + x_-) - 2x_o]$$

Defining, $\bar{x} = \frac{x_+ + x_-}{2}$

We find that:

$$y_+ - y_- = y'_o(x_+ - x_-) + y''_o(x_+ - x_-)(\bar{x} - x_o) + \mathcal{O}(3^+)$$

dividing thru by $(x_+ - x_-)$

$$\frac{y_+ - y_-}{x_+ - x_-} = y'_o + y''_o(\bar{x} - x_o)$$

For the specific case in which x_o is centered between x_- and x_+,

$$x_+ - x_o = x_o - x_-$$

$$x_+ + x_- = 2x_o$$

but: $x_+ + x_- = 2\bar{x}$

Therefore if x_o is centered $\bar{x} = x_o$, and our second order expansion leads to

$$\frac{y_+ - y_-}{x_+ - x_-} = y'_o$$

Thus for centered derivatives

$$\left.\frac{dy}{dx}\right|_{x_i} \simeq \frac{y_{i+\frac{1}{2}} - y_{i-\frac{1}{2}}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}$$

turns out to be an approximation scheme based on a second order expansion due to the vanishing of the second order term.

DERIVATIVE BASED ON THREE POINT PARABOLIC FIT

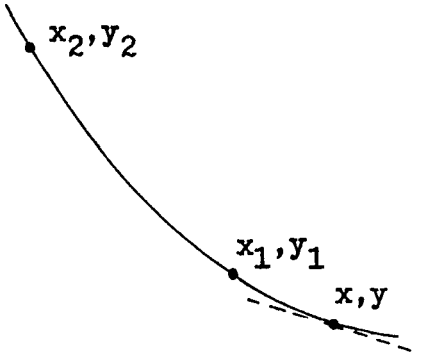


FIGURE D.2

define: $\Delta x_i = x - x_i$, $\Delta y_i = y - y_i$

then from: $$y_1 = ax_1^2 + bx_1 + c \qquad (1)$$

$$y_2 = ax_2^2 + bx_2 + c \qquad (2)$$

$$y = ax^2 + bx + c \qquad (3)$$

$$\frac{dy}{dx} = 2ax + b \qquad (4)$$

we get: $$\frac{\Delta y_1}{\Delta x_1} = a(x + x_1) + b. \qquad (5)$$

by subtracting (1) from (3) and dividing through by $(x - x_1)$ and similarly

$$\frac{\Delta y_2}{\Delta x_2} = a(x + x_2) + b \qquad (6)$$

Eliminating b between (5) and (4) we get:

$$\frac{dy}{dx} = a\Delta x_1 + \frac{\Delta y_1}{\Delta x_1} \qquad (7)$$

similarly from (6) and (4) we get:

$$\frac{dy}{dx} = a\Delta x_2 + \frac{\Delta y_2}{\Delta x_2} \qquad (8)$$

Dividing (7) by Δx_1 and (8) by Δx_2 and subtracting (8) from (7)

$$\left\{\frac{1}{\Delta x_1} - \frac{1}{\Delta x_2}\right\} \frac{dy}{dx} = \frac{\Delta y_1}{\Delta x_1} - \frac{\Delta y_2}{\Delta x_2}$$

$$\frac{dy}{dx} = \frac{1}{(\Delta x_2 - \Delta x_1)} \left\{\frac{\Delta x_2}{\Delta x_1}\Delta y_1 - \frac{\Delta x_1}{\Delta x_2}\Delta y_2\right\}$$

$$= \frac{\Delta x_1 + \Delta x_2}{\Delta x_1 \Delta x_2} y - \frac{\Delta x_2}{\Delta x_1} \frac{y_1}{x_1 - x_2} + \frac{\Delta x_1}{\Delta x_2} \frac{y_2}{x_1 - x_2}$$

APPENDIX E

LUMINOSITY FUNCTION STUDY FOR YOUNG CLUSTERS

Presented here is a partial summary of a study of the luminosity functions of a group of very young galactic clusters conducted in 1968. A description of the methods employed to construct these luminosity functions would be rather extensive and is omitted in the interest of brevity. We will, however, enumerate the five factors we required for the construction of a cluster's luminosity function:

(i) a precise determination of spectral type for each cluster member, based on the MK system of classification;

(ii) photometric data for each member based on multi-color photometry;

(iii) an accurate measurement of the cluster's distance modulus;

(iv) an estimate of the amount of interstellar reddening each cluster member undergoes;

(v) the appropriate bolometric corrections to translate absolute visual magnitude into absolute bolometric magnitude for each star in the clusters.

The last two factors may be further elaborated upon. In correcting for interstellar reddening it is not, as is customary, assumed that R, the ratio of total to selective absorption, is a constant approximately equal to 3.

Variation in R from cluster to cluster was determined from sources such as Reddish (1967) and Wampler (1962). The values obtained by these means were verified wherever possible by comparing photometric data for stars of similar spectral type within a given cluster. Values chosen for the bolometric corrections were based on those supplied by Harris (1963), Johnson (1963), Morton and Adams (1968), and Stothers (1968, private communication). While the bolometric corrections for the very most luminous stars were derived solely from models of stellar atmospheres, the others were, at least in part, based on observational evidence. Recent OAO and rocket observations indicate that our bolometric corrections for the highest luminosity stars may have been significantly underestimated.

In choosing clusters for this study we were highly selective in terms of considering only those for which the five factors enumerated above could be accurately determined. Further criteria consisted of evidence for extreme youth, observations of proper motion of cluster components, and other evidence that allowed us to determine cluster membership with a high degree of certainty. After careful consideration of the clusters selected as well as other clusters rejected because of less reliably determined data, it was concluded that the selection process would not bias our results, but rather produce truly representative composite luminosity functions reflective of the data's reliability.

Table (E.1) lists the more important clusters considered, along with some of the prime references used in determining factors (i) through (iii) for each. In addition to the tabulated references, the following served as highly reliable sources for determination of distance moduli for most of the clusters listed: Hoag and Applequist, (1965); Hoag, Johnson, Iriarte, Mitchell, Hallam, and Sharpless (1961); Johnson (1960); Johnson, Hoag, Iriarte, Mitchell, and Hallam (1961).

Figure (E.1) is a computer plot of the composite luminosity function for the clusters indicated. $\Phi(M_{bol})$ is a normalized quantity representing the number of stars in the magnitude range $M_{bol} \pm \frac{1}{2}$. In this plot Φ has been calculated at 0.1 magnitude intervals. Notice the sharp fall off in Φ at zero bolometric magnitude. Since this corresponds to the relatively low luminosity members of our clusters this phenomenon is simply reflective of an observational cutoff. Actually, for some of the more distant clusters the observational cutoff should correspond to a point one or two magnitudes brighter. Thus we may consider the luminosity curve as increasingly depressed from magnitude $M_{bol} = 1.5$ and fainter. It has been indicated that a truly composite initial luminosity function should show some sort of exponential behavior (Iben and Talbot, 1966; Idlis, 1957;

TABLE E.1 CLUSTERS USED IN 1968 STUDY

NGC 2264	(Morgan, Hiltner, Neff, Garrison, & Osterbreck, 1965; Vasilevskis, Sanders, and Van Altena, 1956b; Walker, 1956)
NGC 6530	(Hiltner, Morgan and Neff, 1965; Iben and Talbot, 1966; Walker, 1957)
NGC 6611	(Morgan, Code, and Whitford, 1955; Walker, 1961)
NGC 6067	(Thackery, Wesselink, and Harding, 1962)
NGC 1893	(Hiltner, 1956, data presented at IAU symposium)
NGC 2244	(Morgan, Hiltner, Neff, Garrison, & Osterbreck, 1965)
IC 5146	(Walker, 1959)
IC 2944	(Thackery and Wesselink, 1965)
IC 1805	(Vasilevskis, Sanders, and Van Altena, 1956a)
ORION Aggregate	(Sharpless, 1952; Strand, 1958; Johnson and Hiltner, 1956, Ap. J., 123, 267)
HYADES Cluster	(Crawford & Perry, 1966; Morgan & Hiltner, 1965)
h & χ PERSEI Clusters	(Schild, 1967)
PLEIDES Cluster	(Johnson and Mitchell, 1958)

Salpeter, 1955; Sandage, 1957a and 1957b; Van Den Bergh, 1957). For convenience we choose to fit the composite luminosity function of figure (E.1) with a gaussian distribution. We use only those objects brighter than 0 magnitude to determine the curve's parameters through a least squares fit. Notice that the portion of the curve near the gaussian's mode approximates the effect of the observational depression prior to the observational cutoff, while the tail of the gaussian fit reflects the exponential behavior of the luminosity function. The curve of figure (E.1) has a mean of 0 and a standard deviation of 4.2. After applying a minor correction to compensate in part for the observational cutoff effect, we find by the usual techniques an expectation value of somewhat less than 3% for the occurrence of objects brighter than $M_{bol} = -10$. Note that because of the manner in which it is defined $\Phi > 0$ up to magnitude - 10.5, even though the brightest star in the survey is of magnitude -10. There are 283 stars represented by figure (E.1), leading to an estimate of at least 7 stars brighter than $M_{bol} = -10$, if the sampling is representative. Furthermore, consideration of figure (E.2) indicates that our estimate is somewhat too low. By omission of the nearest cluster, figure (E.2) emphasizes the effect of observational cutoff for the more distant clusters on the low luminosity end of the luminosity function. In figure (E.2) we have forced a fit to a gaussian having

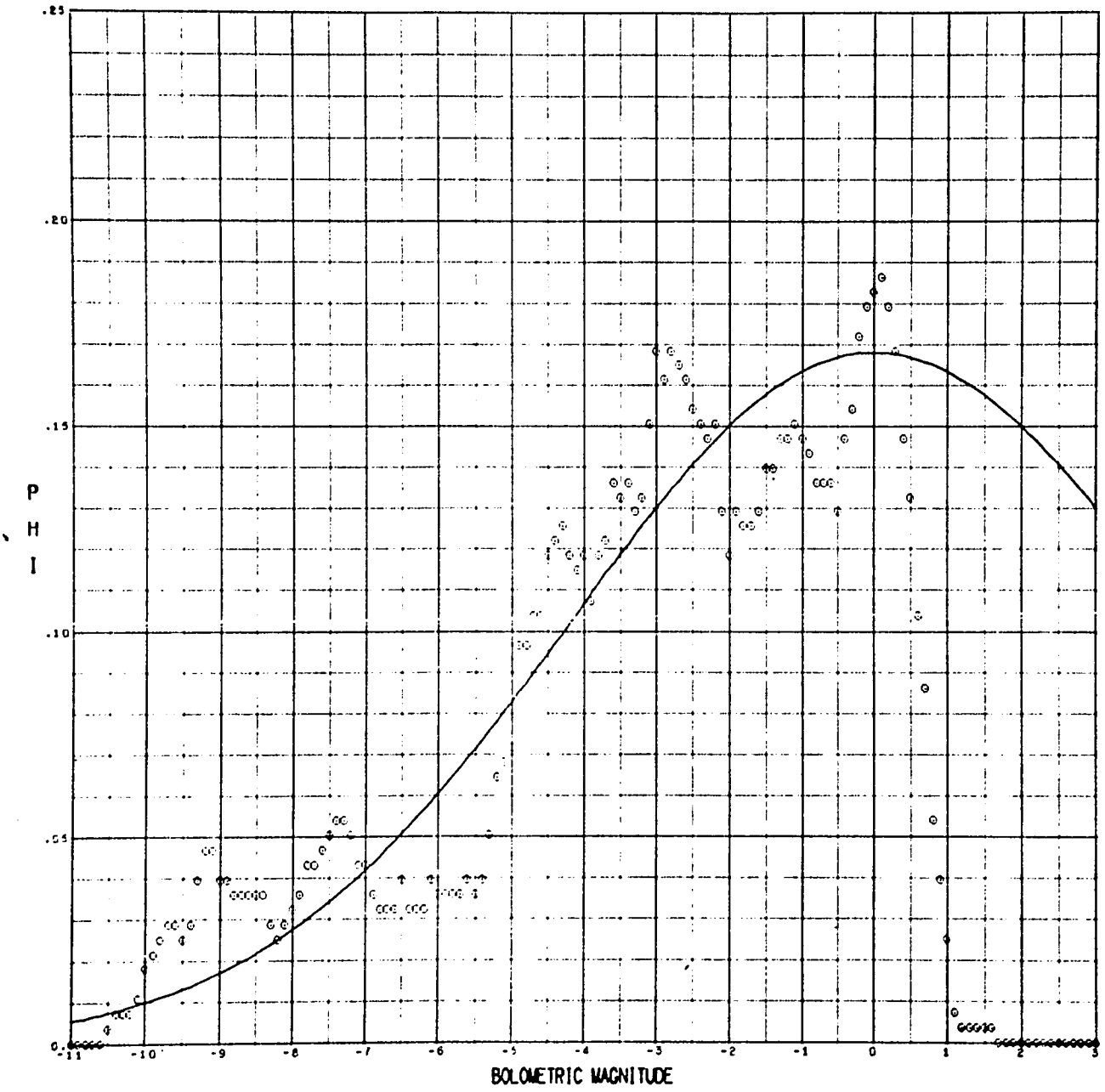
the same mean as in figure (E.1). In this case our standard deviation is greater than 5.4. When we extrapolate this information to the data of figure (E.1), we should conservatively expect to find at least 9 stars out of the 283 sampled to be brighter than $M_{bol} = -10$. In fact, we find none in the sampling just discussed or in any other sampling we considered.

Comparisons with other composites as well as the individual cluster luminosity functions lead ostensibly to the same conclusion; if the luminosity distributions in young clusters are indicative of some exponential initial luminosity, we find a lack of super luminous (very high mass) main sequence stars, which must be attributed to non-genetic effects. An independant evaluation of our data by Simon and Stothers (1970) resulted in the conclusion that when considering the normal O stars in our young clusters, "a cutoff occurs at high luminosity which is faster than the gradual tapering expected by fitting the bulk of the O stars to an exponential representation of the luminosity function; a simple extrapolation of the fitted luminosity function predicts 2 'superluminous' objects per 20 normal O-type stars". These predicted superluminous objects have not been identified as main sequence stars. The significance of the main sequence luminosity cutoff is discussed in Chapter 1.

FIGURE E.1

LUMINOSITY FUNCTION I *

* computer generated plot



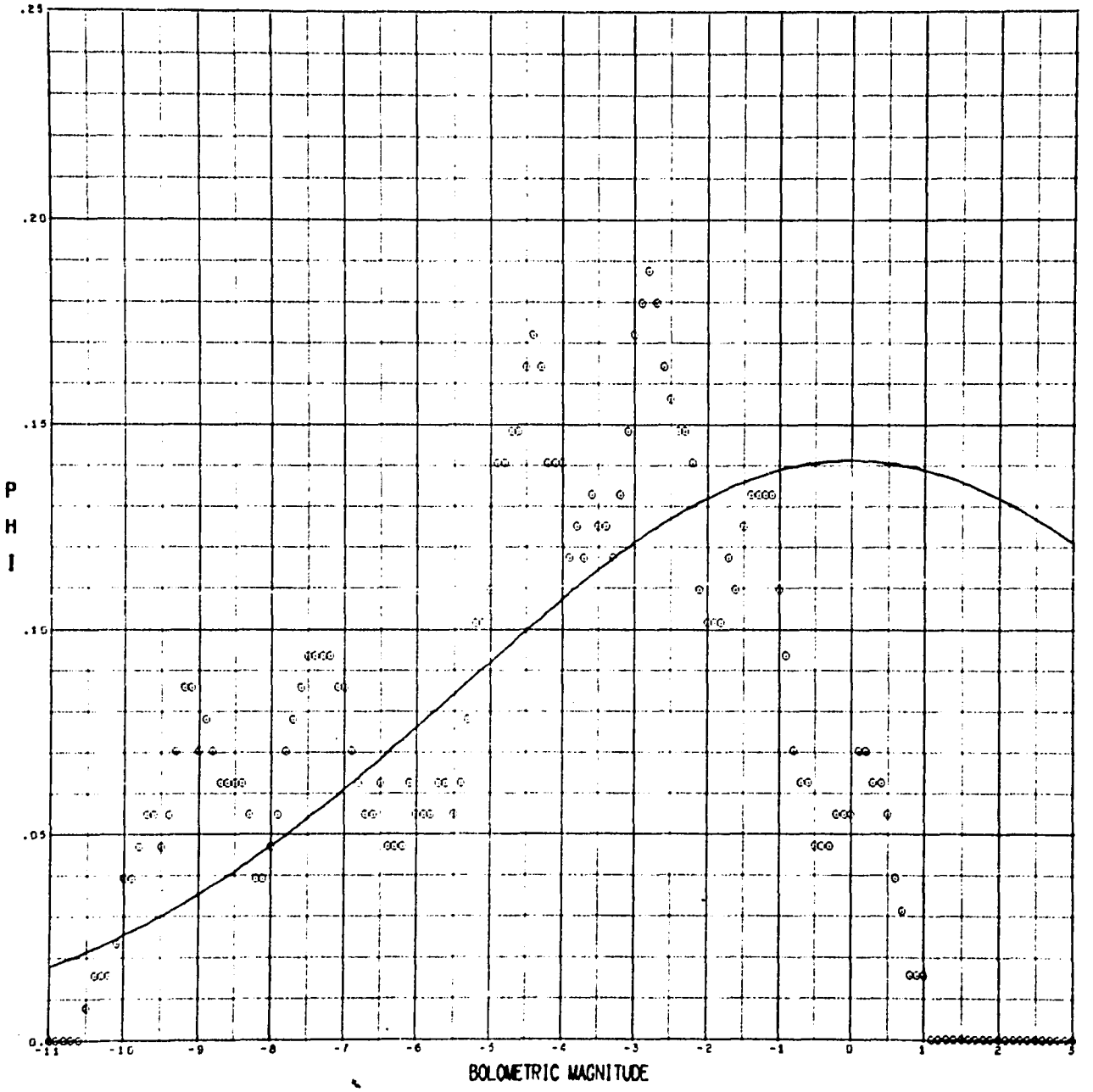
NORMALIZED COMPOSITE OF NGC 1893, 2244, 2264, 6530, 6531, 6611, IC 2944, ORION AGG

NOTE - PHI = PHI(MBOL±0.5)

FIGURE E.2

LUMINOSITY FUNCTION II *

* computer generated plot



NORMALIZED COMPOSITE OF NGC 1893, 2244, 2264, 6530, 6531, 6611, IC 2944

NOTE - PHI = PHI(MBOL±0.5)

APPENDIX F

SAMPLE LISTINGS OF COMPUTER CODES

Because of the large variety of models and techniques developed for this study, several different levels of computer codes were developed. In this appendix we consider the codes used in constructing β Cephei model # 16. We have omitted the subsidiary service routines such as the one used in preparing the plots for this thesis. We have, however, included routines that illustrate how we might have employed a different zoning scheme or boundary conditions for model 16. A list of routines included in this appendix is given below. Each routine is preceded by a brief explanation of its function. The terminology employed is the same as that used in the main text.

F.1 CEPHEI START

F.2 CEPHEI CALC

F.3 CEPHEI ZONE

F.4 CEPHEI ZONE II

F.5 CE. PU. START

F.6 CEPH. PULS.

F.7 CEPHEI LIBRARY (subroutine package)

F.8 BNDRY

F.9 EQUILIB ZONE III

F.1 CEPHEI START

This routine initializes the equilibrium model data set with the first trial values of T_c,P_c,L and R as well as setting the composition and mass parameters X_c,Y_c,X_f, Y_f,M_f, and M.

```
/ID   1444   T UNGAR,S. CPPHRID START      005          01   000
/DD  4   DSNAME=CALC,DISP=SCR
/EXEC    FORTX
      IMPLICIT REAL*8(A-H,K-M,O-Z,$)
      A = 7.5641D-15
      H = 1.67333D-24
      KBOLTZ = 1.38046D-16
      LSUN = 3.90D33
      RSUN = 6.9598D10
C
C
C     ****************THESE VALUES FROM STOTHERS*************************
C
      NMSUN = 15
      XC = 0.7D0
      YC = 0.27D0
      TC = 10.D0**7.5200D0
      RHOC = 10.D0**0.6900D0
      MFFRAC = 0.66
      XF = 0.35
      YF = 0.62
      L0 = 10.D0**4.4600D0*LSUN
      R0 = 10.D0**0.5800*RSUN
C
C     ************************************************************************
C
C
      MUINV = 0.5D0 + 1.5D0*XC + 0.25D0*YC
      C1 = KBOLTZ*MUINV/H
      C5 = A/3.D0
      PC = C1*RHOC*TC + C5*TC**4
      WRITE(4,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   98 FORMAT(I5,1P9D25.16)
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   99 FORMAT(///I9,' SOLAR MASSES'/'0      XC ='F7.4/'      YC =',F7.4/
     X          '0      PC =',1PD23.16/'      TC =',1PD23.16/
     X          '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X          '0      XF =',F7.4/'      YF =',F7.4//
     X          '      L0 =',1PD23.16/'      R0 =',1PD23.16////)
      STOP
      END
/*
/EXEC    LDR
/DD  4   DSNAME=CALC,DEVT=2314,RECFM=F,LRECL=250,DISP=KEEP
/EXEC    *
/*
```

F.2 CEPHEI CALC

These routines find improved values of T_c, P_c, L and R by a series of successive iterations employing the fitting point scheme of section (4.2).

```
/ID   1444   T UNGAR,S. CEPHEID CALC        015          02   000
/DD  4    DSNAME=CALC,DISP=PASS
/EXEC     FORTX,'OPT=2'
C
C     ********************************************************************
C
C     THIS PROGRAM IS DERIVED FROM EQUILIB CALC II, WITH PROVISION MADE
C
C     FOR A DISCONTINUITY IN CHEMICAL COMPOSITION AT MF.
C
C     ********************************************************************
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA, KBOLTZ, KAPPAT
      DIMENSION FPCON(5), FTCON(5), FLCON(5), FRCON(5)
      DIMENSION FPRAD(5), FTRAD(5), FLRAD(5), FRRAD(5)
      DIMENSION PFO(5), TFO(5), LFO(5), RFO(5)
      DIMENSION PFI(5), TFI(5), LFI(5), RFI(5)
      DIMENSION AMAT(5,5), BMAT(5,4)
      COMMON X, Y, Z, ALPHA, KAPPAT
      DATA AD, RAD / '  AD    ', '  RAD   ' /
  100 FORMAT(1H-,T33,'P U L S A T I N G   S T A R       ',15,'   S O L A
     X R   M A S S E S       (EQUILIBRIUM MODEL)'/)
  400 FORMAT(1H0,T5,'MASS',T17,'RADIUS',T33,'HEAT',T46,'PRESSURE',
     X       T62,'DENSITY',T75,'TEMPERATURE',T92,'ENTROPY',T106,
     X       'LUMINOSITY',T122,'ENERGY'/T3,'FRACTION',T18,'(CM)',T31,
     X       'TRANSFER',T45,'(GM/SQ.CM)',T62,'(GM/CC)',T76,'(DEG-ABS)',
     X       T89,'(ERGS/GM-DEG)',T103,'(ERGS/SQ.CM-SEC)',T121,
     X       '(ERGS/GM)'/)
  500 FORMAT(F9.3,1PD15.3,7X,A8,1P6D15.3)
C
C     READ MODEL'S MASS AND CHEMICAL COMPOSITION.
C                 ********** AND **********
C     READ STARTING VALUES FOR CENTRAL PRESSURE AND TEMPERATURE AS WELL
C     AS STARTING VALUES FOR SURFACE LUMINOSITY AND RADIUS.
C
      READ(4,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   98 FORMAT(I5,1P9D25.16)
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   99 FORMAT(///19,' SOLAR MASSES'/'0      XC ='F7.4/'       YC =',F7.4/
     X           '0      PC =',1PD23.16/'       TC =',1PD23.16/
     X           '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X           '0      XF =',F7.4//'       YF =',F7.4//
     X           '       L0 =',1PD23.16/'       R0 =',1PD23.16////)
C
C     SET MAXIMUM MASS ZONE SIZE.
C
      N = 1000
      PI = 3.14159265358979320D0
      FPI = 4.D0*PI
      FPITHD = FPI/3.D0
      A = 7.5641D-15
      A0 = 2.81785D-13
      C = 2.997929D10
      G = 6.673D-8
      H = 1.67333D-24
```

```
      KBOLTZ = 1.38046D-16
      RSUN = 6.9598D10
      MSUN = 1.989D33
      LSUN = 3.90D33
      SIGMA = A*C/4.D0
      C2 = 4.D0*A/3.D0
      C4 = 4.D0*A
      C5 = A/3.D0
      OPCON4 = 64.D0*A*C*PI**2/3.D0
      MSTAR = NMSUN*MSUN
      DMM = MSTAR/N
C
C     SET FITTING POINT MASS.
C
      MP = MPFRAC*MSTAR
      EXP1 = 1.D0/3.D0
      EXP2 = 2.D0/3.D0
      FRAC = 8.D0/3.D0
      GCONST = G/FPI
C
C     SET MAXIMUM PRESSURE ZONE SIZE.
C
      GCON2 = 1.D-1/GCONST
C
C     SET SIZE OF PERTURBATIONS.
C
      DELFAC = 1.D-6
C
C     CHOOSE NUMBER OF SETS OF SIX INTEGRATIONS FOR RUN.
C
      NITER = 8
      DO 1000 ITER=1,NITER
      DELPPC = DELFAC*PC
      DELTTC = DELFAC*TC
      DELLL0 = DELFAC*L0
      DELRR0 = DELFAC*R0
      ASSIGN 30 TO NI
    5 CONTINUE
C
C     SET COMPOSITION DEPENDANT CONSTANTS.
C
      X = XC
      Y = YC
      Z = 1.D0 - X - Y
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)
      KAPPAT = (FPI/3.D0)*A0**2*(1.D0 + X)/H
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      C1 = KBOLTZ*MUINV/H
      C3 = 3.D0*C1
      XCNO = (1.D0 - X - Y)/2.D0
      FCONST = 7.9D31*X*XCNO
C
C     SET BRANCHING FOR ONLY ONE CONVECTIVE REGION.
C
      ASSIGN 12 TO NLABEL
C
```

```
C     ********************************************************************
C     START INTEGRATION FROM CENTER OUT.              *****CORE*****
C     ********************************************************************
C
C     CALCULATE CENTRAL VALUES.
C
      V = AD
      M = 0.D0
      MFRAC = 0.D0
      R = 0.D0
      L = 0.D0
      T = TC
      P = PC
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,100) NMSUN
      WRITE(3,400)
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
C
C     CALCULATE VALUES ONE ZONE OUT BY APPROXIMATE EXPANSION.
C
      BETA = C1*RHO*T/P
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)
      M = DMM
      R = ((3.D0/FPI)*DMM/RHO)**EXP1
      T = T - (FPI/6.D0)*G*R**2*RHO**2*(1.D0 - 1.D0/GAMMA2)*T/P
      P = P - (FPI/6.D0)*G*R**2*RHO**2
      L = DMM*F
      MFRAC = M/MSTAR
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
C
C     BEGIN INTEGRATION.
C
    6 CONTINUE
      PM = P
      TM = T
      LM = L
      RM = R
      DMP = GCON2*R**4*P/M
      DM = DMIN1(DMM,DMP)
    8 DO 10 K=2,5
      A = ((K - 1)/2)/2.D0
      MT = M + DM*A
      PTCON = P + FPCON(K-1)*A
      PTRAD = P + FPRAD(K-1)*A
      TTCON = T + FTCON(K-1)*A
      TT4RAD = T**4 + FTRAD(K-1)*A
      TTRAD = TT4RAD**0.25D0
      LTCON = L + FLCON(K-1)*A
      LTRAD = L + FLRAD(K-1)*A
      RT3CON = R**3 + FRCON(K-1)*A
      RTCON = RT3CON**EXP1
```

```
      RT3RAD = R**3 + FRRAD(K-1)*A
      RTRAD = RT3RAD**EXP1
      RHOCON = (PTCON - C5*TTCON**4)/(C1*TTCON)
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)
      BETA= C1*RHOCON*TTCON/PTCON
      GAMMA2 = 1.D0 + (BETA - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)
      FPCON(K) = -GCONST*MF*DM/RTCON**4
      FPRAD(K) = -GCONST*MF*DM/RTRAD**4
      FTCON(K) = (1.D0 - 1.D0/GAMMA2)*TTCON*FPCON(K)/PTCON
      OPCON3 = OPCON4/KAPPA(RHORAD,TTRAD)
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)
      FLCON(K)=FCONST*RHOCON*DEXP(-1.5231D4/TTCON**EXP1)*DM/TTCON**EXP2
      FLRAD(K)=FCONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2
      FRCON(K) = DM/(FPITHD*RHOCON)
      FRRAD(K) = DM/(FPITHD*RHORAD)
   10 CONTINUE
      M = M + DM
      MFRAC = M/MSTAR
      PCON = P + (FPCON(2)+2.D0*FPCON(3)+2.D0*FPCON(4)+FPCON(5))/6.D0
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0
      TCON = T + (FTCON(2)+2.D0*FTCON(3)+2.D0*FTCON(4)+FTCON(5))/6.D0
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0
      TRAD = T4RAD**0.25D0
      LCON = L + (FLCON(2)+2.D0*FLCON(3)+2.D0*FLCON(4)+FLCON(5))/6.D0
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0
      R3CON = R**3+(FRCON(2)+2.D0*FRCON(3)+2.D0*FRCON(4)+FRCON(5))/6.D0
      RCON = R3CON**EXP1
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0
      RRAD = R3RAD**EXP1
      IF(TCON .GT. TRAD) GO TO NLABEL, (11,12)
C
C     SET BRANCHING FOR ONLY RADIATIVE REGION CALCULATIONS.
C
      ASSIGN 11 TO NLABEL
   11 CONTINUE
      P = PRAD
      T = TRAD
      L = LRAD
      R = RRAD
      V = RAD
      GO TO 14
   12 CONTINUE
      P = PCON
      T = TCON
      L = LCON
      R = RCON
      V = AD
   14 CONTINUE
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      IF(M - MF) 6,20,16
   16 M = M - DM
      P = PM
      T = TM
      L = LM
```

```
      R = RM
      DM = MF - M
      GO TO 8
   20 CONTINUE
      GO TO N1, (30,40,50,70,80,90)
   30 CONTINUE
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
      PFO(1) = P
      TFO(1) = T
      LFO(1) = L
      RFO(1) = R
      PC = PC + DELPPC
      ASSIGN 40 TO N1
      GO TO 5
   40 CONTINUE
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
      PFO(2) = P
      TFO(2) = T
      LFO(2) = L
      RFO(2) = R
      PC = PC - DELPPC
      TC = TC + DELTTC
      ASSIGN 50 TO N1
      GO TO 5
   50 CONTINUE
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
      PFO(3) = P
      TFO(3) = T
      LFO(3) = L
      RFO(3) = R
      PFO(4) = PFO(1)
      TFO(4) = TFO(1)
      LFO(4) = LFO(1)
      RFO(4) = RFO(1)
      PFO(5) = PFO(1)
      TFO(5) = TFO(1)
      LFO(5) = LFO(1)
      RFO(5) = RFO(1)
      TC = TC - DELTTC
      ASSIGN 70 TO N1
  105 CONTINUE
C
C     RESET COMPOSITION DEPENDANT CONSTANTS.
C
      X = XF
      Y = YF
      Z = 1.D0 - X - Y
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)
      KAPPAT = (PPI/3.D0)*A0**2*(1.D0 + X)/H
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      C1 = KBOLTZ*MUINV/H
      C3 = 3.D0*C1
      XCNO = (1.D0 - X - Y)/2.D0
      FCONST = 7.9D31*X*XCNO
C
C     ******************************************************************
```

```
C     START INTEGRATION FROM SURFACE IN.                *****ENVELOPE*****
C     ************************************************************************
C
C     CALCULATE SURFACE VALUES.
C
      V = RAD
      T0 = (L0/(FPI*SIGMA*R0**2))**0.25D0
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPAT*R0**2) + C5*T0**4/2.D0
      RHOT = (P0 - C5*T0**4)/(C1*T0)
C
C     ITERATE TO FIND SURFACE PRESSURE AND DENSITY.
C
 1045 CONTINUE
      RHO = RHOT
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPA(RHO,T0)*R0**2) + C5*T0**4/2.D0
      RHOT = (P0 - C5*T0**4)/(C1*T0)
      IF(DABS((RHO-RHOT)/RHOT) .GT. 1.D-12) GO TO 1045
      M = MSTAR
      MFRAC = 1.D0
      L = L0
      R = R0
      P = P0
      T = T0
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,100) NMSUN
      WRITE(3,400)
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
C
C     BEGIN INTEGRATION.
C
  106 CONTINUE
      PM = P
      TM = T
      LM = L
      RM = R
      DMP = GCON2*R**4*P/M
      DM = -DMIN1(DMM,DMP)
  108 DO 110 K=2,5
      A = ((K - 1)/2)/2.D0
      MT = M + DM*A
      PTRAD = P + FPRAD(K-1)*A
      TT4RAD = T**4 + FTRAD(K-1)*A
      TTRAD = TT4RAD**0.25D0
      LTRAD = L + FLRAD(K-1)*A
      RT3RAD = R**3 + FRRAD(K-1)*A
      RTRAD = RT3RAD**EXP1
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)
      FPRAD(K) = -GCONST*MT*DM/RTRAD**4
      OPCON3 = OPCON4/KAPPA(RHORAD,TTRAD)
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)
      FLRAD(K)=ECONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2
      FRRAD(K) = DM/(FPITHD*RHORAD)
  110 CONTINUE
      M = M + DM
```

```
      MFRAC = M/MSTAR
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0
      TRAD = T4RAD**0.25D0
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0
      RRAD = R3RAD**EXP1
      P = PRAD
      T = TRAD
      L = LRAD
      R = RRAD
      V = RAD
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      IF(M - MF) 116,60,106
  116 M = M - DM
      P = PM
      T = TM
      L = LM
      R = RM
      DM = MF - M
      GO TO 108
   60 CONTINUE
      GO TO N1, (30,40,50,70,80,90)
   70 CONTINUE
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
      PPI(1) = P
      TPI(1) = T
      LPI(1) = L
      RPI(1) = R
      PPI(2) = PPI(1)
      TPI(2) = TPI(1)
      LPI(2) = LPI(1)
      RPI(2) = RPI(1)
      PPI(3) = PPI(1)
      TPI(3) = TPI(1)
      LPI(3) = LPI(1)
      RPI(3) = RPI(1)
      L0 = L0 + DELLL0
      ASSIGN 80 TO N1
      GO TO 105
   80 CONTINUE
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
      PPI(4) = P
      TPI(4) = T
      LPI(4) = L
      RPI(4) = R
      L0 = L0 - DELLL0
      R0 = R0 + DELRR0
      ASSIGN 90 TO N1
      GO TO 105
   90 CONTINUE
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
      PPI(5) = P
      TPI(5) = T
```

```
      LPI(5) = L
      RPI(5) = R
      R0 = R0 - DELRR0
      DO 95 I=1,5
      AMAT(I,1) = PPO(I) - PPI(I)
      AMAT(I,2) = TPO(I) - TPI(I)
      AMAT(I,3) = LPO(I) - LPI(I)
      AMAT(I,4) = RPO(I) - RPI(I)
      AMAT(I,5) = -1.D0
      DO 95 J=1,4
      BMAT(I,J) = 0.D0
   95 CONTINUE
      WRITE(3,97) (AMAT(1,I), I=1,4)
   97 FORMAT(///'       DELP =',1PD24.16/'       DELT =',1PD24.16/
     X           '       DELL =',1PD24.16/'       DELR =',1PD24.16////)
      BMAT(2,1) = DELPPC
      BMAT(3,2) = DELTTC
      BMAT(4,3) = DELLL0
      BMAT(5,4) = DELRR0
      CALL LEQ(AMAT,BMAT,5,4,5,5,ERR)
      DELPC = BMAT(5,1)
      DELTC = BMAT(5,2)
      DELL0 = BMAT(5,3)
      DELR0 = BMAT(5,4)
      PC = PC - DELPC
      TC = TC - DELTC
      L0 = L0 - DELL0
      R0 = R0 - DELR0
 1000 CONTINUE
      REWIND 4
      WRITE(4,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
      STOP
      END
      FUNCTION KAPPA(RHO,T)
      IMPLICIT REAL*8(A-H,K,O-Z,$)
      COMMON X, Y, Z, ALPHA, KAPPAT
      PF = ALPHA*RHO*T
      T4 = T / 1.D4
      KAPPAS = 4.85D-13*PF/((1.D0+2.2D-5*T4)*RHO*T4)
      KAPPAX = PF*X*(T4**0.5D0/(2.D6/T4**4+2.1D0*T4**6)+1.D0
     1          /(4.5D0*T4**6+1.D0/(T4*(4.D-3/T4**4+2.D-4/RHO**0.25D0))))
      KAPPAY = PF*Y*(1.D0/(1.4D3*T4+T4**6)+1.5D0/(1.D6+0.1D0*T4**6))
      KAPPAZ = PF*Z*T4**0.5D0/(2.D1*T4+3.5D1*T4**4+3.8D2*Z
     1          *(RHO*(2.D0-Z)**2*(1+X))**0.33D0*T4**4.71D0)
      KAPPA = KAPPAS + KAPPAX + KAPPAY + KAPPAZ
      RETURN
      END
/*
/EXEC    LDR
/EXEC    *
/EOJ
/EOJ
/EOJ
/*
```

F.3 CEPHEI ZONE

These routines produce a zoned equilibrium model in accordance with either the subdivided or uniform mass zoning schemes from the values determined by one or more passes of CEPHEI CALC (F.2). The output serves as an input data set for CE. PU. START (F.5).

```
/ID   1444   T UNGAR,S. CEPHEID ZONE        005          02   000
/DD  4    DSNAME=CALC,DISP=PASS
/DD  8    DSNAME=TEMP,DISP=SCR
/EXEC     FORTX,'OPT=2'
C
C     **********************************************************************
C
C     THIS PROGRAM IS DERIVED FROM EQUILIB ZONE V, WITH PROVISION MADE
C
C     FOR A DISCONTINUITY IN CHEMICAL COMPOSITION AT MFF.
C
C     **********************************************************************
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)                                    ZNE00020
      REAL*8 KAPPA, KBOLTZ, KAPPAT                                      ZNE00030
      DIMENSION FPCON(5), FTCON(5), FLCON(5), FRCON(5)                  ZNE00040
      DIMENSION FPRAD(5), FTRAD(5), FLRAD(5), FRRAD(5)                  ZNE00050
      DATA AD, RAD / '  AD    ', '  RAD   ' /                          ZNE00060
  100 FORMAT(1H1,T33,'P U L S A T I N G   S T A R       ',15,'   S O L AZNE00070
     X R   M A S S E S       (EQUILIBRIUM MODEL)'/)                     ZNE00080
  200 FORMAT('/*')                                                      ZNE00090
  300 FORMAT(8A8)                                                       ZNE00100
  400 FORMAT(1H0,T5,'MASS',T17,'RADIUS',T33,'HEAT',T46,'PRESSURE',      ZNE00110
     X       T62,'DENSITY',T75,'TEMPERATURE',T92,'ENTROPY',T106,        ZNE00120
     X       'LUMINOSITY',T122,'ENERGY'/T3,'FRACTION',T18,'(CM)',T31,   ZNE00130
     X       'TRANSFER',T45,'(GM/SQ.CM)',T62,'(GM/CC)',T76,'(DEG-ABS)', ZNE00140
     X       T89,'(ERGS/GM-DEG)',T103,'(ERGS/SQ.CM-SEC)',T121,          ZNE00150
     X       '(ERGS/GM)'/)                                              ZNE00160
  500 FORMAT(F9.4,1PD15.3,7X,A8,1P6D15.3)                               ZNE00170
C
C     CHOOSE NUMBER OF OUTPUT ZONES FOR EQUIVALENT EQUALLE-ZONED MODEL. ZNE00180
C
      NZONE = 300                                                       ZNE00190
      NZONE = 500
      NZONE = 600
      NZONE = 400
C
C     READ MODEL'S MASS AND CHEMICAL COMPOSITION.                       ZNE00200
C                 ********** AND **********                             ZNE00210
C     READ STARTING VALUES FOR CENTRAL PRESSURE AND TEMPERATURE AS WELL ZNE00220
C     AS STARTING VALUES FOR SURFACE LUMINOSITY AND RADIUS.             ZNE00230
C
      READ(4,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   98 FORMAT(I5,1P9D25.16)
      WRITE(9,198) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
  198 FORMAT(A4,9A8)
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   99 FORMAT(///I9,' SOLAR MASSES'/'0     XC ='F7.4/'      YC =',F7.4/
     X          '0     PC =',1PD23.16/'      TC =',1PD23.16/          ZNE00300
     X          '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X          '0     XF =',F7.4/'      YF =',F7.4//
     X          '      L0 =',1PD23.16/'      R0 =',1PD23.16////)      ZNE00310
C
C     SET MAXIMUM MASS ZONE SIZE.                                       ZNE00320
C
      N = 1000                                                          ZNE00330
```

```
      PI = 3.14159265358979320                                          ZNE00340
      PPI = 4.D0*PI                                                     ZNE00350
      PPITHD = PPI/3.D0                                                 ZNE00360
      A = 7.5641D-15                                                    ZNE00370
      A0 = 2.81785D-13                                                  ZNE00380
      C = 2.997929D10                                                   ZNE00390
      G = 6.673D-8                                                      ZNE00400
      H = 1.67333D-24                                                   ZNE00410
      KBOLTZ = 1.38046D-16                                              ZNE00420
      RSUN = 6.9598D10                                                  ZNE00430
      MSUN = 1.989D33                                                   ZNE00440
      LSUN = 3.90D33                                                    ZNE00450
      SIGMA = A*C/4.D0                                                  ZNE00480
      C2 = 4.D0*A/3.D0                                                  ZNE00500
      C4 = 4.D0*A                                                       ZNE00520
      C5 = A/3.D0                                                       ZNE00530
      OPCON4 = 64.D0*A*C*PI**2/3.D0                                     ZNE00540
      MSTAR = NMSUN*MSUN                                                ZNE00550
      DMM = MSTAR/N                                                     ZNE00580
C
C     SET FITTING POINT MASS.
C
      MFF = MFFRAC*MSTAR
C
C     SELECT VALUES FOR FIXED CONSTANTS.
C
      EXP1 = 1.D0/3.D0                                                  ZNE00610
      EXP2 = 2.D0/3.D0                                                  ZNE00620
      FRAC = 8.D0/3.D0                                                  ZNE00630
      GCONST = G/PPI                                                    ZNE00640
C
C     SET MAXIMUM PRESSURE ZONE SIZE.                                   ZNE00650
C
      GCON2 = 1.D-1/GCONST                                              ZNE00660
      GCON2 = 1.D-2/GCONST
C
C     CHOOSE CONSTANT ZONE SIZE FOR CORE REGION.                        ZNE00730
C
      DMZONE = MSTAR/NZONE
C
C     REVALUATE MFF TO ASSURE ALIGNMENT WITH MASS ZONES.
C
      MFF = IDINT(NZONE*MFF/MSTAR + 9.D-1)*DMZONE
C
C     SET COMPOSITION DEPENDANT CONSTANTS.
C
      X = XC
      Y = YC
      KAPPAT = (PPI/3.D0)*A0**2*(1.D0 + X)/H                            ZNE00460
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y                                ZNE00470
      C1 = KBOLTZ*MUINV/H                                               ZNE00490
      C3 = 3.D0*C1                                                      ZNE00510
      XCNO = (1.D0 - X - Y)/2.D0                                        ZNE00590
      ECONST = 7.9D31*X*XCNO                                            ZNE00600
C
C     SET BRANCHING FOR ONLY ONE CONVECTIVE REGION.
```

```
C
      ASSIGN 12 TO NLABEL
C
C     ************************************************************************ZNE00750
C     START INTEGRATION FROM CENTER OUT.              *****CORE*****       ZNE00760
C     ************************************************************************ZNE00770
C
C     CALCULATE CENTRAL VALUES.                                              ZNE00780
C
      V = AD                                                                 ZNE00790
      M = 0.D0                                                               ZNE00800
      MFRAC = 0.D0                                                           ZNE00810
      R = 0.D0                                                               ZNE00820
      L = 0.D0                                                               ZNE00830
      T = TC                                                                 ZNE00840
      P = PC                                                                 ZNE00850
      RHO = (P - C5*T**4)/(C1*T)                                             ZNE00860
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                                ZNE00870
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                         ZNE00880
      WRITE(3,100) NMSUN                                                     ZNE00890
      WRITE(9,300) M, R, RHO, P, T, S, L, E                                  ZNE00900
      WRITE(3,400)                                                           ZNE00910
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E                                   ZNE00920
C
C     CALCULATE VALUES ONE ZONE OUT BY APPROXIMATE EXPANSION.                ZNE00930
C
      BETA = C1*RHO*T/P                                                      ZNE00940
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)        ZNE00950
      M = DMM                                                                ZNE00960
      R = ((3.D0/FPI)*DMM/RHO)**EXP1                                         ZNE00970
      T = T - (FPI/6.D0)*G*R**2*RHO**2*(1.D0 - 1.D0/GAMMA2)*T/P              ZNE00980
      P = P - (FPI/6.D0)*G*R**2*RHO**2                                       ZNE00990
      L = DMM*E                                                              ZNE01000
      MFRAC = M/MSTAR                                                        ZNE01010
      RHO = (P - C5*T**4)/(C1*T)                                             ZNE01020
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                                ZNE01030
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                         ZNE01040
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E                                   ZNE01050
C
C     BEGIN INTEGRATION.                                                     ZNE01060
C
      MF = DMZONE                                                            ZNE01070
    6 CONTINUE                                                               ZNE01080
      MM = M                                                                 ZNE01090
      PM = P                                                                 ZNE01100
      TM = T                                                                 ZNE01110
      LM = L                                                                 ZNE01120
      RM = R                                                                 ZNE01130
      DMP = GCON2*R**4*P/M                                                   ZNE01140
      DM = DMIN1(DMM,DMP)                                                    ZNE01150
    8 DO 10 K=2,5                                                            ZNE01160
      A = ((K - 1)/2)/2.D0                                                   ZNE01170
      MT = M + DM*A                                                          ZNE01180
      PTCON = P + FPCON(K-1)*A                                               ZNE01190
      PTRAD = P + FPRAD(K-1)*A                                               ZNE01200
      TTCON = T + FTCON(K-1)*A                                               ZNE01210
```

```
      TT4RAD = T**4 + FTRAD(K-1)*A                                          ZNE01220
      TTRAD = TT4RAD**0.25D0                                                ZNE01230
      LTCON = L + FLCON(K-1)*A                                              ZNE01240
      LTRAD = L + FLRAD(K-1)*A                                              ZNE01250
      RT3CON = R**3 + FRCON(K-1)*A                                          ZNE01260
      RTCON = RT3CON**EXP1                                                  ZNE01270
      RT3RAD = R**3 + FRRAD(K-1)*A                                          ZNE01280
      RTRAD = RT3RAD**EXP1                                                  ZNE01290
      RHOCON = (PTCON - C5*TTCON**4)/(C1*TTCON)                             ZNE01300
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)                             ZNE01310
      BETA= C1*RHOCON*TTCON/PTCON                                           ZNE01320
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)       ZNE01330
      FPCON(K) = -GCONST*MT*DM/RTCON**4                                     ZNE01340
      FPRAD(K) = -GCONST*MT*DM/RTRAD**4                                     ZNE01350
      FTCON(K) = (1.D0 - 1.D0/GAMMA2)*TTCON*FPCON(K)/PTCON                  ZNE01360
      OPCON3 = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)                               ZNE01370
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)                           ZNE01380
      FLCON(K)=FCONST*RHOCON*DEXP(-1.5231D4/TTCON**EXP1)*DM/TTCON**EXP2     ZNE01390
      FLRAD(K)=FCONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2     ZNE01400
      FRCON(K) = DM/(FP1THD*RHOCON)                                         ZNE01410
      FRRAD(K) = DM/(FP1THD*RHORAD)                                         ZNE01420
   10 CONTINUE                                                              ZNE01430
      M = M + DM                                                            ZNE01440
      MFRAC = M/MSTAR                                                       ZNE01450
      PCON = P + (FPCON(2)+2.D0*FPCON(3)+2.D0*FPCON(4)+FPCON(5))/6.D0       ZNE01460
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0       ZNE01470
      TCON = T + (FTCON(2)+2.D0*FTCON(3)+2.D0*FTCON(4)+FTCON(5))/6.D0       ZNE01480
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0     ZNE01490
      TRAD = T4RAD**0.25D0                                                  ZNE01500
      LCON = L + (FLCON(2)+2.D0*FLCON(3)+2.D0*FLCON(4)+FLCON(5))/6.D0       ZNE01510
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0       ZNE01520
      R3CON = R**3+(FRCON(2)+2.D0*FRCON(3)+2.D0*FRCON(4)+FRCON(5))/6.D0     ZNE01530
      RCON = R3CON**EXP1                                                    ZNE01540
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0     ZNE01550
      RRAD = R3RAD**EXP1                                                    ZNE01560
      IF(TCON .GT. TRAD) GO TO NLABEL, (11,12)
C
C     SET BRANCHING FOR ONLY RADIATIVE REGION CALCULATIONS.
C
      ASSIGN 11 TO NLABEL
   11 CONTINUE
      P = PRAD                                                              ZNE01580
      T = TRAD                                                              ZNE01590
      L = LRAD                                                              ZNE01600
      R = RRAD                                                              ZNE01610
      V = RAD                                                               ZNE01620
      GO TO 14                                                              ZNE01630
   12 CONTINUE                                                              ZNE01640
      P = PCON                                                              ZNE01650
      T = TCON                                                              ZNE01660
      L = LCON                                                              ZNE01670
      R = RCON                                                              ZNE01680
      V = AD                                                                ZNE01690
   14 CONTINUE                                                              ZNE01700
      RHO = (P - C5*T**4)/(C1*T)                                            ZNE01710
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                               ZNE01720
```

```
      P = PCONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                    ZNE01730
      IF(M - MF) 6,20,16                                                ZNE01740
   16 CONTINUE                                                          ZNE01750
      M = MM                                                            ZNE01760
      P = PM                                                            ZNE01770
      T = TM                                                            ZNE01780
      L = LM                                                            ZNE01790
      R = RM                                                            ZNE01800
      DM = MF - M                                                       ZNE01810
      GO TO 8                                                           ZNE01820
   20 CONTINUE                                                          ZNE01830
      WRITE(9,300) M, R, RHO, P, T, S, L, E                             ZNE01840
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E                              ZNE01850
      MF = MF + DMZONE                                                  ZNE01860
      IF(MF .LE. MFF) GO TO 6                                           ZNE01870
      WRITE(9,200)                                                      ZNE01880
      PFO = P                                                           ZNE01890
      TFO = T                                                           ZNE01900
      LFO = L                                                           ZNE01910
      RFO = R                                                           ZNE01920
C
C     RESET COMPOSITION DEPENDANT CONSTANTS.
C
      X = XF
      Y = YF
      KAPPAT = (FPI/3.D0)*A0**2*(1.D0 + X)/H
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      C1 = KBOLTZ*MUINV/H
      C3 = 3.D0*C1
      XCNO = (1.D0 - X - Y)/2.D0
      ECONST = 7.9D31*X*XCNO
C
C     ************************************************************************ZNE01930
C     START INTEGRATION FROM SURFACE IN.              *****ENVELOPE*****  ZNE01940
C     ************************************************************************ZNE01950
C
C     CALCULATE SURFACE VALUES.                                         ZNE01960
C
      V = RAD                                                           ZNE01970
      T0 = (L0/(FPI*SIGMA*R0**2))**0.25D0                               ZNE01980
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPAT*R0**2) + C5*T0**4/2.D0             ZNE01990
      RHOT = (P0 - C5*T0**4)/(C1*T0)                                    ZNE02000
C
C     ITERATE TO FIND SURFACE PRESSURE AND DENSITY.                     ZNE02010
C
 1045 CONTINUE                                                          ZNE02020
      RHO = RHOT                                                        ZNE02030
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPA(X,Y,RHO,T0)*R0**2) + C5*T0**4/2.D0  ZNE02040
      RHOT = (P0 - C5*T0**4)/(C1*T0)                                    ZNE02050
      IF(DABS((RHO-RHOT)/RHOT) .GT. 1.D-12) GO TO 1045                  ZNE02060
      M = MSTAR                                                         ZNE02070
      MFRAC = 1.D0                                                      ZNE02080
      L = L0                                                            ZNE02090
      R = R0                                                            ZNE02100
      P = P0                                                            ZNE02110
      T = T0                                                            ZNE02120
```

```
      RHO = (P - C5*T**4)/(C1*T)                                        ZNE02130
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                           ZNE02140
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                    ZNE02150
      WRITE(3,100) NMSUN                                                ZNE02160
      WRITE(3,400)                                                      ZNE02170
      WRITE(9,300) M, R, RHO, P, T, S, L, E                             ZNE02180
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E                              ZNE02190
C
C     CHOOSE NUMBER OF ZONES FOR ENVELOPE REGION.
C
C     NOTE - FOR UNIFORM ZONE SIZE SET NEXP = 0, NFITZN .GT. 2.
C
      NFITZN = 10
      NEXP = 5
      NFITZN = 6
      NEXP = 7
      NEXP = 0
      NEXP = 8
C
C     SET OUTPUT MASS ZONE SIZES IN ENVELOPE REGION.                    ZNE0220
C
      CALL FITSET(DMZONE,NFITZN,NEXP)
      MF = MSTAR - DMZONE
C
C     BEGIN INTEGRATION.                                                ZNE02220
C
  106 CONTINUE                                                          ZNE02230
      MM = M                                                            ZNE02240
      PM = P                                                            ZNE02250
      TM = T                                                            ZNE02260
      LM = L                                                            ZNE02270
      RM = R                                                            ZNE02280
      DMP = GCON2*R**4*P/M                                              ZNE02290
      DM = -DMIN1(DMM,DMP)                                              ZNE02300
  108 DO 110 K=2,5                                                      ZNE02310
      A = ((K - 1)/2)/2.D0                                              ZNE02320
      MT = M + DM*A                                                     ZNE02330
      PTRAD = P + FPRAD(K-1)*A                                          ZNE02350
      TT4RAD = T**4 + FTRAD(K-1)*A                                      ZNE02370
      TTRAD = TT4RAD**0.25D0                                            ZNE02380
      LTRAD = L + FLRAD(K-1)*A                                          ZNE02400
      RT3RAD = R**3 + FRRAD(K-1)*A                                      ZNE02430
      RTRAD = RT3RAD**EXP1                                              ZNE02440
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)                         ZNE02460
      FPRAD(K) = -GCONST*MT*DM/RTRAD**4                                 ZNE02500
      OPCON3 = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)                           ZNE02520
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)                       ZNE02530
      FLRAD(K)=ECONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2 ZNE02550
      FRRAD(K) = DM/(FPITHD*RHORAD)                                     ZNE02570
  110 CONTINUE                                                          ZNE02580
      M = M + DM                                                        ZNE02590
      MFRAC = M/MSTAR                                                   ZNE02600
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0   ZNE02620
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0 ZNE02640
      TRAD = T4RAD**0.25D0                                              ZNE02650
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0   ZNE02670
```

```
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0 ZNE02700
      RRAD = R3RAD**EXP1                                                ZNE02710
      P = PRAD                                                          ZNE02730
      T = TRAD                                                          ZNE02740
      L = LRAD                                                          ZNE02750
      R = RRAD                                                          ZNE02760
      V = RAD                                                           ZNE02770
      RHO = (P - C5*T**4)/(C1*T)                                        ZNE02860
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                           ZNE02870
      F = FCONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                    ZNE02880
      IF(M - MF) 116,60,106                                             ZNE02890
  116 CONTINUE                                                          ZNE02900
      M = MM                                                            ZNE02910
      P = PM                                                            ZNE02920
      T = TM                                                            ZNE02930
      L = LM                                                            ZNE02940
      R = RM                                                            ZNE02950
      DM = MF - M                                                       ZNE02960
      GO TO 108                                                         ZNE02970
   60 CONTINUE                                                          ZNE02980
      WRITE(9,300) M, R, RHO, P, T, S, L, F                             ZNE02990
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F                              ZNE03000
C
C     FIND OUTPUT MASS ZONE SIZES IN ENVELOPE REGION.
C
      CALL MASFIT(DMZONE)
      MF = MF - DMZONE
      IF((MF - MFF)/MF .GT. 1.D-12) GO TO 106
      IF(MF .EQ. MFF - DMZONE) GO TO 80
      IF((MFF - MF)/MF .GT. 1.D-12) WRITE(3,95) MFF, MF
   95 FORMAT(///' Z O N E   O V E R L A P   T O O   L A R G E     MFF =',
     X       1PD23.16,'     MF ='1PD23.16//)
      MF = MFF
      GO TO 106
   80 CONTINUE
      WRITE(9,200)                                                      ZNE03030
      PFI = P                                                           ZNE03040
      TFI = T                                                           ZNE03050
      LFI = L                                                           ZNE03060
      RFI = R                                                           ZNE03070
      DELP = PFO - PFI                                                  ZNE04130
      DELT = TFO - TFI                                                  ZNE04140
      DELL = LFO - LFI                                                  ZNE04150
      DELR = RFO - RFI                                                  ZNE04160
      WRITE(3,97) DELP, DELT, DELL, DELR                                ZNE04170
   97 FORMAT(1H1/'-     GOODNESS OF FIT AT FITTING POINT'///           ZNE04180
     X           '      DELP =',1PD24.16/'      DELT =',1PD24.16/      ZNE04190
     X           '      DELL =',1PD24.16/'      DELR =',1PD24.16////)  ZNE04200
      REWIND 9                                                          ZNE04210
      STOP                                                              ZNE04220
      END                                                               ZNE04230
      FUNCTION KAPPA(X,Y,RHO,T)                                         ZNE05040
      IMPLICIT REAL*8(A-H,K,O-Z,$)                                      ZNE05050
      DATA H /1.67333D-24/, KBOLTZ /1.38046D-16/
      Z = 1.D0 - X - Y                                                  ZNE05100
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)                                ZNE05110
```

```
      PR = ALPHA*RHO*T                                                  ZNF05140
      T4 = T / 1.D4                                                     ZNF05150
      KAPPAS = 4.85D-13*PR/((1.D0+2.2D-5*T4)*RHO*T4)                    ZNF05160
      KAPPAX = PR*X*(T4**0.5D0/(2.D6/T4**4+2.1D0*T4**6)+1.D0            ZNF05170
     1          /(4.5D0*T4**6+1.D0/(T4*(4.D-3/T4**4+2.D-4/RHO**0.25D0))))ZNF05180
      KAPPAY = PR*Y*(1.D0/(1.4D3*T4+T4**6)+1.5D0/(1.D6+0.1D0*T4**6))    ZNF05190
      KAPPAZ = PR*Z*T4**0.5D0/(2.D1*T4+3.5D1*T4**4+3.8D2*Z              ZNF05200
     1          *(RHO*(2.D0-Z)**2*(1+X))**0.33D0*T4**4.71D0)            ZNF05210
      KAPPA = KAPPAS + KAPPAX + KAPPAY + KAPPAZ                         ZNF05220
      RETURN                                                            ZNF05230
      END                                                               ZNF05240
      SUBROUTINE FITSET(DMZONE,NSUBZN,NEXP)
      IMPLICIT REAL*8(A-H,O-Z,$)
      IF (NSUBZN/2 .EQ. (NSUBZN+1)/2)  GO TO 10
      WRITE(3,20) NSUBZN
   20 FORMAT(/'-E R R O R    E X I T     ODD NUMBER OF SUBZONES INDICATED
     X      NSUBZN =',I3,'  IS ILLEGAL'///)
      CALL EXIT
   10 CONTINUE
      NCOUNT = -2
      SUBZNE = NSUBZN/2
      NSUBZN = NSUBZN - 2
      DMZONE = DMZONE/SUBZNE**NEXP
      NPIN = NEXP*NSUBZN + 1
      ENTRY MASFIT(DMZONE)
      IF(NCOUNT .GT. NPIN)  RETURN
      NCOUNT = NCOUNT + 1
      IF(NCOUNT .LE. 1)  RETURN
      IF(MOD(NCOUNT,NSUBZN) .EQ. 1)  DMZONE = SUBZNE*DMZONE
      RETURN
      END
/*
/EXEC     LDR
/DD  9    DSNAME=TEMP,DEVT=(2314,FILPRO),RECFM=CI,DISP=KEEP
/EXEC     *
/DD  99   DSNAME=TEST,DISP=SCR
/EXEC     FORTX          DATA SET REARRANGEMENT   BASED ON VERSION 3  9/69
C
C     THIS PROGRAM REARRANGES THE OUTPUT FROM EQUILIBRIUM ZONE 'IV' TO   ZNF0526
C     BE COMPATABLE WITH THE INPUT TO THE CURRENT VERSION OF THE STAR1   ZNF05270
C     SUBROUTINE IN THE PULSE III CODE.                                  ZNF05280
C
      REAL*8 XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
      REAL*8 A(1000), B(1000), C(1000), D(1000)                         ZNF05310
      REAL*8 E(1000), F(1000), G(1000), H(1000)                         ZNF05320
   10 FORMAT(8A8)                                                       ZNF05330
   20 FORMAT('/*')                                                      ZNF05340
   30 FORMAT(////' TOTAL NUMBER OF POINTS CONSIDERED =',I4//            ZNF05350
     X        ' FITTING POINT IS NUMBER',I4//
     X             ' NUMBER OF POINTS IN ENVELOPE =',I4////)
      DO 100 J=1,2                                                      ZNF05380
   90 CONTINUE                                                          ZNF05390
      I = I + 1                                                         ZNF05400
      READ(8,10,END=100) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)        ZNF05410
      GO TO 90                                                          ZNF05420
  100 CONTINUE                                                          ZNF05430
```

```
      N = I - 2                                                          ZNF05440
      I = 0                                                              ZNF05450
      REWIND 8                                                           ZNF05460
    1 CONTINUE                                                           ZNF05470
      I = I + 1                                                          ZNF05480
      READ(8,10,END=2) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)           ZNF05490
      GO TO 1                                                            ZNF05500
    2 CONTINUE                                                           ZNF05510
      NPIT = I - 2                                                       ZNF05520
      I = N                                                              ZNF05530
    3 CONTINUE                                                           ZNF05540
      I = I - 1                                                          ZNF05550
      READ(8,10,END=4) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)           ZNF05560
      GO TO 3                                                            ZNF05570
    4 CONTINUE                                                           ZNF05580
      NM1 = N - 1
      WRITE(9,10) (A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I),I=2,NM1)
      WRITE(9,20)                                                        ZNF05670
      REWIND 8                                                           ZNF05680
      READ(8,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   98 FORMAT(A4,9A8)
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   99 FORMAT(///19,' SOLAR MASSES'//'0      XC ='F7.4/'      YC =',F7.4/
     X           '0      PC =',1PD23.16/'      TC =',1PD23.16/          ZNF05730
     X           '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X           '0      XF =',F7.4/'      YF =',F7.4//
     X           '      L0 =',1PD23.16/'      R0 =',1PD23.16////)       ZNF05740
      NTOT = NM1 - 1
      NENV = NTOT - NPIT
      WRITE(3,30) NTOT, NPIT, NENV
      STOP                                                               ZNF05790
      END                                                                ZNF05800
/*
/EXEC    LDR
/DD  8   DSNAME=TEMP,DISP=PASS
/DD  9   DSNAME=TEST,DEVT=(2314,FILPRO),RECFM=CI,DISP=KEEP
/EXEC    *
/DD  8   DSNAME=TEMP,DISP=SCR
/EOJ
/EOJ
/EOJ
/*
```

F.4 CEPHEI ZONE II

This set of routines performs the same functions as CEPHEI ZONE (F.3), but produces a Christy zoned model.

```
/ID   1444   T UNGAR,S. CEPHEID ZONE II     005          02   000
/DD  4    DSNAME=CALC,DISP=PASS
/DD  8    DSNAME=TEMP,DISP=SCR
/EXEC     FORTX,'OPT=2'
C
C     ********************************************************************
C
C     REVISION OF CEPHEID ZONE WITH CHRISTY TYPE MASS ZONING SCHEME.
C
C     ********************************************************************
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)                                    ZNE00020
      REAL*8 KAPPA, KBOLTZ, KAPPAT                                      ZNE00030
      DIMENSION PPCON(5), PTCON(5), PLCON(5), PRCON(5)                  ZNE00040
      DIMENSION PPRAD(5), PTRAD(5), PLRAD(5), PRRAD(5)                  ZNE00050
      DATA AD, RAD / '  AD    ', '  RAD   ' /                          ZNE00060
  100 FORMAT(1H1,T33,'P U L S A T I N G   S T A R       ',I5,'   S O L AZNE00070
     X R   M A S S E S       (EQUILIBRIUM MODEL)'/)                     ZNE00080
  200 FORMAT('/*')                                                      ZNE00090
  300 FORMAT(8A8)                                                       ZNE00100
  400 FORMAT(1H0,T5,'MASS',T17,'RADIUS',T33,'HEAT',T46,'PRESSURE',      ZNE00110
     X       T62,'DENSITY',T75,'TEMPERATURE',T92,'ENTROPY',T106,        ZNE00120
     X       'LUMINOSITY',T122,'ENERGY'/T3,'FRACTION',T18,'(CM)',T31,   ZNE00130
     X       'TRANSFER',T45,'(GM/SQ.CM)',T62,'(GM/CC)',T76,'(DEG-ABS)', ZNE00140
     X       T89,'(ERGS/GM-DEG)',T103,'(ERGS/SQ.CM-SEC)',T121,          ZNE00150
     X       '(ERGS/GM)'/)                                              ZNE00160
  500 FORMAT(F9.4,1PD15.3,7X,A8,1P6D15.3)                               ZNE00170
C
C     CHOOSE NUMBER OF OUTPUT ZONES FOR EQUIVALENT EQUALLE-ZONED MODEL. ZNE00180
C
      NZONE = 300                                                       ZNE00190
      NZONE = 500
      NZONE = 600
      NZONE = 400
C
C     READ MODEL'S MASS AND CHEMICAL COMPOSITION.                       ZNE00200
C                  ********** AND **********                           ZNE00210
C     READ STARTING VALUES FOR CENTRAL PRESSURE AND TEMPERATURE AS WELL ZNE00220
C     AS STARTING VALUES FOR SURFACE LUMINOSITY AND RADIUS.             ZNE00230
C
      READ(4,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   98 FORMAT(I5,1P9D25.16)
      WRITE(9,198) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
  198 FORMAT(A4,9A8)
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   99 FORMAT(///I9,' SOLAR MASSES'/'0     XC ='F7.4/'      YC =',F7.4/
     X          '0     PC =',1PD23.16/'      TC =',1PD23.16/          ZNE00300
     X          '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X          '0     XF =',F7.4/'      YF =',F7.4//
     X          '      L0 =',1PD23.16/'      R0 =',1PD23.16////)      ZNE00310
C
C     SET MAXIMUM MASS ZONE SIZE.                                       ZNE00320
C
      N = 1000                                                          ZNE00330
      PI = 3.14159265358979320                                          ZNE00340
      PPI = 4.D0*PI                                                     ZNE00350
```

```
      PPITHD = PPI/3.D0                                                 ZNE00360
      A = 7.5641D-15                                                    ZNE00370
      A0 = 2.81785D-13                                                  ZNE00380
      C = 2.997929D10                                                   ZNE00390
      G = 6.673D-8                                                      ZNE00400
      H = 1.67333D-24                                                   ZNE00410
      KBOLTZ = 1.38046D-16                                              ZNE00420
      RSUN = 6.9598D10                                                  ZNE00430
      MSUN = 1.989D33                                                   ZNE00440
      LSUN = 3.90D33                                                    ZNE00450
      SIGMA = A*C/4.D0                                                  ZNE00480
      C2 = 4.D0*A/3.D0                                                  ZNE00500
      C4 = 4.D0*A                                                       ZNE00520
      C5 = A/3.D0                                                       ZNE00530
      OPCON4 = 64.D0*A*C*PI**2/3.D0                                     ZNE00540
      MSTAR = NMSUN*MSUN                                                ZNE00550
      DMM = MSTAR/N                                                     ZNE00580
C
C     SET FITTING POINT MASS.
C
      MFP = MFPFRAC*MSTAR
C
C     SELECT VALUES FOR FIXED CONSTANTS.
C
      EXP1 = 1.D0/3.D0                                                  ZNE00610
      EXP2 = 2.D0/3.D0                                                  ZNE00620
      FRAC = 8.D0/3.D0                                                  ZNE00630
      GCONST = G/PPI                                                    ZNE00640
C
C     SET MAXIMUM PRESSURE ZONE SIZE.                                   ZNE00650
C
      GCON2 = 1.D-1/GCONST                                              ZNE00660
      GCON2 = 1.D-2/GCONST
C
C     CHOOSE CONSTANT ZONE SIZE FOR CORE REGION.                        ZNE00730
C
      DMZONE = MSTAR/NZONE
C
C     REVALUATE MFP TO ASSURE ALIGNMENT WITH MASS ZONES.
C
      MFP = IDINT(NZONE*MFP/MSTAR + 9.D-1)*DMZONE
C
C     SET COMPOSITION DEPENDANT CONSTANTS.
C
      X = XC
      Y = YC
      KAPPAT = (PPI/3.D0)*A0**2*(1.D0 + X)/H                            ZNE00460
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y                                ZNE00470
      C1 = KBOLTZ*MUINV/H                                               ZNE00490
      C3 = 3.D0*C1                                                      ZNE00510
      XCNO = (1.D0 - X - Y)/2.D0                                        ZNE00590
      ECONST = 7.9D31*X*XCNO                                            ZNE00600
C
C     SET BRANCHING FOR ONLY ONE CONVECTIVE REGION.
C
      ASSIGN 12 TO NLABEL
```

```
C
C     ********************************************************************ZNE00750
C     START INTEGRATION FROM CENTER OUT.              *****CORE*****      ZNE00760
C     ********************************************************************ZNE00770
C
C     CALCULATE CENTRAL VALUES.                                           ZNE00780
C
      V = AD                                                              ZNE00790
      M = 0.D0                                                            ZNE00800
      MFRAC = 0.D0                                                        ZNE00810
      R = 0.D0                                                            ZNE00820
      L = 0.D0                                                            ZNE00830
      T = TC                                                              ZNE00840
      P = PC                                                              ZNE00850
      RHO = (P - C5*T**4)/(C1*T)                                          ZNE00860
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                             ZNE00870
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                      ZNE00880
      WRITE(3,100) NMSUN                                                  ZNE00890
      WRITE(9,300) M, R, RHO, P, T, S, L, E                               ZNE00900
      WRITE(3,400)                                                        ZNE00910
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E                                ZNE00920
C
C     CALCULATE VALUES ONE ZONE OUT BY APPROXIMATE EXPANSION.             ZNE00930
C
      BETA = C1*RHO*T/P                                                   ZNE00940
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)     ZNE00950
      M = DMM                                                             ZNE00960
      R = ((3.D0/FPI)*DMM/RHO)**EXP1                                      ZNE00970
      T = T - (FPI/6.D0)*G*R**2*RHO**2*(1.D0 - 1.D0/GAMMA2)*T/P           ZNE00980
      P = P - (FPI/6.D0)*G*R**2*RHO**2                                    ZNE00990
      L = DMM*E                                                           ZNE01000
      MFRAC = M/MSTAR                                                     ZNE01010
      RHO = (P - C5*T**4)/(C1*T)                                          ZNE01020
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                             ZNE01030
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                      ZNE01040
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E                                ZNE01050
C
C     BEGIN INTEGRATION.                                                  ZNE01060
C
      MF = DMZONE                                                         ZNE01070
    6 CONTINUE                                                            ZNE01080
      MM = M                                                              ZNE01090
      PM = P                                                              ZNE01100
      TM = T                                                              ZNE01110
      LM = L                                                              ZNE01120
      RM = R                                                              ZNE01130
      DMP = GCON2*R**4*P/M                                                ZNE01140
      DM = DMIN1(DMM,DMP)                                                 ZNE01150
    8 DO 10 K=2,5                                                         ZNE01160
      A = ((K - 1)/2)/2.D0                                                ZNE01170
      MT = M + DM*A                                                       ZNE01180
      PTCON = P + PPCON(K-1)*A                                            ZNE01190
      PTRAD = P + PPRAD(K-1)*A                                            ZNE01200
      TTCON = T + PTCON(K-1)*A                                            ZNE01210
      TT4RAD = T**4 + PTRAD(K-1)*A                                        ZNE01220
      TTRAD = TT4RAD**0.25D0                                              ZNE01230
```

```
      LTCON = L + PLCON(K-1)*A                                                  ZNE01240
      LTRAD = L + PLRAD(K-1)*A                                                  ZNE01250
      RT3CON = R**3 + PRCON(K-1)*A                                              ZNE01260
      RTCON = RT3CON**EXP1                                                      ZNE01270
      RT3RAD = R**3 + PRRAD(K-1)*A                                              ZNE01280
      RTRAD = RT3RAD**EXP1                                                      ZNE01290
      RHOCON = (PTCON - C5*TTCON**4)/(C1*TTCON)                                 ZNE01300
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)                                 ZNE01310
      BETA= C1*RHOCON*TTCON/PTCON                                               ZNE01320
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)           ZNE01330
      PPCON(K) = -GCONST*MT*DM/RTCON**4                                         ZNE01340
      PPRAD(K) = -GCONST*MT*DM/RTRAD**4                                         ZNE01350
      PTCON(K) = (1.D0 - 1.D0/GAMMA2)*TTCON*PPCON(K)/PTCON                      ZNE01360
      OPCON3 = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)                                   ZNE01370
      PTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)                               ZNE01380
      PLCON(K)=ECONST*RHOCON*DEXP(-1.5231D4/TTCON**EXP1)*DM/TTCON**EXP2         ZNE01390
      PLRAD(K)=ECONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2         ZNE01400
      PRCON(K) = DM/(PPITHD*RHOCON)                                             ZNE01410
      PRRAD(K) = DM/(PPITHD*RHORAD)                                             ZNE01420
   10 CONTINUE                                                                  ZNE01430
      M = M + DM                                                                ZNE01440
      MFRAC = M/MSTAR                                                           ZNE01450
      PCON = P + (PPCON(2)+2.D0*PPCON(3)+2.D0*PPCON(4)+PPCON(5))/6.D0           ZNE01460
      PRAD = P + (PPRAD(2)+2.D0*PPRAD(3)+2.D0*PPRAD(4)+PPRAD(5))/6.D0           ZNE01470
      TCON = T + (PTCON(2)+2.D0*PTCON(3)+2.D0*PTCON(4)+PTCON(5))/6.D0           ZNE01480
      T4RAD = T**4+(PTRAD(2)+2.D0*PTRAD(3)+2.D0*PTRAD(4)+PTRAD(5))/6.D0         ZNE01490
      TRAD = T4RAD**0.25D0                                                      ZNE01500
      LCON = L + (PLCON(2)+2.D0*PLCON(3)+2.D0*PLCON(4)+PLCON(5))/6.D0           ZNE01510
      LRAD = L + (PLRAD(2)+2.D0*PLRAD(3)+2.D0*PLRAD(4)+PLRAD(5))/6.D0           ZNE01520
      R3CON = R**3+(PRCON(2)+2.D0*PRCON(3)+2.D0*PRCON(4)+PRCON(5))/6.D0         ZNE01530
      RCON = R3CON**EXP1                                                        ZNE01540
      R3RAD = R**3+(PRRAD(2)+2.D0*PRRAD(3)+2.D0*PRRAD(4)+PRRAD(5))/6.D0         ZNE01550
      RRAD = R3RAD**EXP1                                                        ZNE01560
      IP(TCON .GT. TRAD) GO TO NLABEL, (11,12)
C
C     SET BRANCHING FOR ONLY RADIATIVE REGION CALCULATIONS.
C
      ASSIGN 11 TO NLABEL
   11 CONTINUE
      P = PRAD                                                                  ZNE01580
      T = TRAD                                                                  ZNE01590
      L = LRAD                                                                  ZNE01600
      R = RRAD                                                                  ZNE01610
      V = RAD                                                                   ZNE01620
      GO TO 14                                                                  ZNE01630
   12 CONTINUE                                                                  ZNE01640
      P = PCON                                                                  ZNE01650
      T = TCON                                                                  ZNE01660
      L = LCON                                                                  ZNE01670
      R = RCON                                                                  ZNE01680
      V = AD                                                                    ZNE01690
   14 CONTINUE                                                                  ZNE01700
      RHO = (P - C5*T**4)/(C1*T)                                                ZNE01710
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                                   ZNE01720
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                            ZNE01730
      IP(M - MP) 6,20,16                                                        ZNE01740
```

```
   16 CONTINUE                                                          ZNE01750
      M = MM                                                            ZNE01760
      P = PM                                                            ZNE01770
      T = TM                                                            ZNE01780
      L = LM                                                            ZNE01790
      R = RM                                                            ZNE01800
      DM = MP - M                                                       ZNE01810
      GO TO 6                                                           ZNE01820
   20 CONTINUE                                                          ZNE01830
      WRITE(9,300) M, R, RHO, P, T, S, L, F                             ZNE01840
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F                              ZNE01850
      MP = MP + DMZONE                                                  ZNE01860
      IF(MP .LE. MPP) GO TO 6                                           ZNE01870
      WRITE(9,200)                                                      ZNE01880
      PPO = P                                                           ZNE01890
      TPO = T                                                           ZNE01900
      LPO = L                                                           ZNE01910
      RPO = R                                                           ZNE01920
C
C     RESET COMPOSITION DEPENDANT CONSTANTS.
C
      X = XP
      Y = YP
      KAPPAT = (PPI/3.D0)*A0**2*(1.D0 + X)/H
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      C1 = KBOLTZ*MUINV/H
      C3 = 3.D0*C1
      XCNO = (1.D0 - X - Y)/2.D0
      ECONST = 7.9D31*X*XCNO
C
C     ************************************************************************ZNE01930
C     START INTEGRATION FROM SURFACE IN.              *****ENVELOPE*****  ZNE01940
C     ************************************************************************ZNE01950
C
C     CALCULATE SURFACE VALUES.                                         ZNE01960
C
      V = RAD                                                           ZNE01970
      T0 = (L0/(PPI*SIGMA*R0**2))**0.25D0                               ZNE01980
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPAT*R0**2) + C5*T0**4/2.D0             ZNE01990
      RHOT = (P0 - C5*T0**4)/(C1*T0)                                    ZNE02000
C
C     ITERATE TO FIND SURFACE PRESSURE AND DENSITY.                     ZNE02010
C
 1045 CONTINUE                                                          ZNE02020
      RHO = RHOT                                                        ZNE02030
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPA(X,Y,RHO,T0)*R0**2) + C5*T0**4/2.D0  ZNE02040
      RHOT = (P0 - C5*T0**4)/(C1*T0)                                    ZNE02050
      IF(DABS((RHO-RHOT)/RHOT) .GT. 1.D-12) GO TO 1045                  ZNE02060
      M = MSTAR                                                         ZNE02070
      MFRAC = 1.D0                                                      ZNE02080
      L = L0                                                            ZNE02090
      R = R0                                                            ZNE02100
      P = P0                                                            ZNE02110
      T = T0                                                            ZNE02120
      RHO = (P - C5*T**4)/(C1*T)                                        ZNE02130
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                           ZNE02140
```

```
      R = RCONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                    ZNE02150
      WRITE(3,100) NMSUN                                                ZNE02160
      WRITE(3,400)                                                      ZNE02170
      WRITE(9,300) M, R, RHO, P, T, S, L, P                             ZNE02180
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,P                              ZNE02190
C
C     CHOOSE NUMBER OF ZONES FOR ENVELOPE REGION.
C
C     NOTE - FOR UNIFORM ZONE SIZE SET NENV = (1 - MFE/MSTAR)*N.
C
      NENV = 175
      NENV = 300
      NENV = 200
      NENV = 150
      NENV = 250
      NENV = 30
      NENV = 50
      NENV = 60
C
C     SET OUTPUT MASS ZONING PARAMETER.
C
      MENV = 0.99D0*MSTAR
      CALL MASFIT(MSTAR,MENV,DMZONE,NENV,ZNCON)
C
C     FIND OUTPUT MASS ZONE SIZES IN ENVELOPE REGION.                   ZNE02200
C
      DMZONE = ZNCON**NENV*DMZONE
      MF = MSTAR - DMZONE
C
C     BEGIN INTEGRATION.                                                ZNE02220
C
  106 CONTINUE                                                          ZNE02230
      MM = M                                                            ZNE02240
      PM = P                                                            ZNE02250
      TM = T                                                            ZNE02260
      LM = L                                                            ZNE02270
      RM = R                                                            ZNE02280
      DMP = GCON2*R**4*P/M                                              ZNE02290
      DM = -DMIN1(DMM,DMP)                                              ZNE02300
  108 DO 110 K=2,5                                                      ZNE02310
      A = ((K - 1)/2)/2.D0                                              ZNE02320
      MT = M + DM*A                                                     ZNE02330
      PTRAD = P + FPRAD(K-1)*A                                          ZNE02350
      TT4RAD = T**4 + FTRAD(K-1)*A                                      ZNE02370
      TTRAD = TT4RAD**0.25D0                                            ZNE02380
      LTRAD = L + FLRAD(K-1)*A                                          ZNE02400
      RT3RAD = R**3 + FRRAD(K-1)*A                                      ZNE02430
      RTRAD = RT3RAD**EXP1                                              ZNE02440
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)                         ZNE02460
      FPRAD(K) = -GCONST*MT*DM/RTRAD**4                                 ZNE02500
      OPCON3 = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)                           ZNE02520
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)                       ZNE02530
      FLRAD(K)=ECONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2 ZNE02550
      FRRAD(K) = DM/(FPITHD*RHORAD)                                     ZNE02570
  110 CONTINUE                                                          ZNE02580
      M = M + DM                                                        ZNE02590
```

```
      MFRAC = M/MSTAR                                                    ZNE02600
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0    ZNE02620
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0  ZNE02640
      TRAD = T4RAD**0.25D0                                               ZNE02650
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0    ZNE02670
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0  ZNE02700
      RRAD = R3RAD**EXP1                                                 ZNE02710
      P = PRAD                                                           ZNE02730
      T = TRAD                                                           ZNE02740
      L = LRAD                                                           ZNE02750
      R = RRAD                                                           ZNE02760
      V = RAD                                                            ZNE02770
      RHO = (P - C5*T**4)/(C1*T)                                         ZNE02860
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO                            ZNE02870
      R = RCONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2                     ZNE02880
      IF(M - MF) 116,60,106                                              ZNE02890
  116 CONTINUE                                                           ZNE02900
      M = MM                                                             ZNE02910
      P = PM                                                             ZNE02920
      T = TM                                                             ZNE02930
      L = LM                                                             ZNE02940
      R = RM                                                             ZNE02950
      DM = MF - M                                                        ZNE02960
      GO TO 108                                                          ZNE02970
   60 CONTINUE                                                           ZNE02980
      WRITE(9,300) M, R, RHO, P, T, S, L, F                              ZNE02990
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F                               ZNE03000
C
C     FIND OUTPUT MASS ZONE SIZES IN ENVELOPE REGION.
C
      IF((MENV-MF)/MENV .LE. 1.D-12)   DMZONE = DMZONE/ZNECON
      MF = MF - DMZONE
      IF((MF - MFF)/MF .GT. 1.D-12) GO TO 106
      IF(MF .EQ. MFF - DMZONE) GO TO 80
      IF((MFF - MF)/MF .GT. 1.D-12) WRITE(3,95) MFF, MF
   95 FORMAT(///' Z O N E   O V E R L A P   T O O   L A R G E     MFF =',
     X       1PD23.16,'     MF ='1PD23.16//)
      MF = MFF
      GO TO 106
   80 CONTINUE
      WRITE(9,200)                                                       ZNE03030
      PPI = P                                                            ZNE03040
      TPI = T                                                            ZNE03050
      LPI = L                                                            ZNE03060
      RPI = R                                                            ZNE03070
      DELP = PPO - PPI                                                   ZNE04130
      DELT = TPO - TPI                                                   ZNE04140
      DELL = LPO - LPI                                                   ZNE04150
      DELR = RPO - RPI                                                   ZNE04160
      WRITE(3,97) DELP, DELT, DELL, DELR                                 ZNE04170
   97 FORMAT(1H1/'-     GOODNESS OF FIT AT FITTING POINT'///             ZNE04180
     X           '      DELP =',1PD24.16/'      DELT =',1PD24.16/       ZNE04190
     X           '      DELL =',1PD24.16/'      DELR =',1PD24.16////)   ZNE04200
      REWIND 9                                                           ZNE04210
      STOP                                                               ZNE04220
      END                                                                ZNE04230
```

```
      FUNCTION KAPPA(X,Y,RHO,T)                                          ZNE05040
      IMPLICIT REAL*8(A-H,K,O-Z,$)                                       ZNE05050
      DATA H /1.67333D-24/, KBOLTZ /1.38046D-16/
      Z = 1.D0 - X - Y                                                   ZNE05100
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)                                 ZNE05110
      PR = ALPHA*RHO*T                                                   ZNE05140
      T4 = T / 1.D4                                                      ZNE05150
      KAPPAS = 4.85D-13*PR/((1.D0+2.2D-5*T4)*RHO*T4)                     ZNE05160
      KAPPAX = PR*X*(T4**0.5D0/(2.D6/T4**4+2.1D0*T4**6)+1.D0             ZNE05170
     1          /(4.5D0*T4**6+1.D0/(T4*(4.D-3/T4**4+2.D-4/RHO**0.25D0))))ZNE05180
      KAPPAY = PR*Y*(1.D0/(1.4D3*T4+T4**6)+1.5D0/(1.D6+0.1D0*T4**6))     ZNE05190
      KAPPAZ = PR*Z*T4**0.5D0/(2.D1*T4+3.5D1*T4**4+3.8D2*Z               ZNE05200
     1          *(RHO*(2.D0-Z)**2*(1+X))**0.33D0*T4**4.71D0)             ZNE05210
      KAPPA = KAPPAS + KAPPAX + KAPPAY + KAPPAZ                          ZNE05220
      RETURN                                                             ZNE05230
      END                                                                ZNE05240
      SUBROUTINE MASFIT(MSTAR,MFF,DMZONE,NE,B)
C
C     THIS SUBROUTINE GIVES CHRISTY TYPE MASS ZONES OF UNIFORMLY
C     DECREASING MASSES.
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      F(B) = (1.D0 - B**NE)*B/(1.D0 - B) - (MSTAR - MFF)/DMZONE
      DF(B) = (1.D0 - (NE+1)*B**NE + B*(1.D0-B**NE)/(1.D0-B))/(1.D0-B)
      BT = 0.95D0
   50 CONTINUE
      B = BT
      BT = B - F(B)/DF(B)
      IF(DABS((B-BT)/BT) .GT. 1.D-12) GO TO 50
      RETURN
      END
/*
/EXEC     LDR
/DD  9    DSNAME=TEMP,DEVT=(2314,PILPRO),RECFM=CI,DISP=KEEP
/EXEC     *
/DD  99   DSNAME=TEST,DISP=SCR
/EXEC     PORTX          DATA SET REARRANGEMENT    BASED ON VERSION 3  9/69
C
C     THIS PROGRAM REARRANGES THE OUTPUT FROM EQUILIBRIUM ZONE 'IV' TO   ZNE0526
C     BE COMPATABLE WITH THE INPUT TO THE CURRENT VERSION OF THE STAR1   ZNE05270
C     SUBROUTINE IN THE PULSE III CODE.                                  ZNE05280
C
      REAL*8 XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
      REAL*8 A(1000), B(1000), C(1000), D(1000)                          ZNE05310
      REAL*8 E(1000), F(1000), G(1000), H(1000)                          ZNE05320
   10 FORMAT(8A8)                                                        ZNE05330
   20 FORMAT('/*')                                                       ZNE05340
   30 FORMAT(/////' TOTAL NUMBER OF POINTS CONSIDERED =',I4//            ZNE05350
     X        ' FITTING POINT IS NUMBER',I4//
     X             ' NUMBER OF POINTS IN ENVELOPE =',I4////)
      DO 100 J=1,2                                                       ZNE05380
   90 CONTINUE                                                           ZNE05390
      I = I + 1                                                          ZNE05400
      READ(8,10,END=100) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)         ZNE05410
      GO TO 90                                                           ZNE05420
  100 CONTINUE                                                           ZNE05430
```

```
      N = I - 2                                                          ZNP05440
      I = 0                                                              ZNP05450
      REWIND 8                                                           ZNP05460
    1 CONTINUE                                                           ZNP05470
      I = I + 1                                                          ZNP05480
      READ(8,10,END=2) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)           ZNP05490
      GO TO 1                                                            ZNP05500
    2 CONTINUE                                                           ZNP05510
      NPIT = I - 2                                                       ZNP05520
      I = N                                                              ZNP05530
    3 CONTINUE                                                           ZNP05540
      I = I - 1                                                          ZNP05550
      READ(8,10,END=4) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)           ZNP05560
      GO TO 3                                                            ZNP05570
    4 CONTINUE                                                           ZNP05580
      NM1 = N - 1
      WRITE(9,10) (A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I),I=2,NM1)
      WRITE(9,20)                                                        ZNP05670
      REWIND 8                                                           ZNP05680
      READ(8,98) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   98 FORMAT(A4,9A8)
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
   99 FORMAT(///19,' SOLAR MASSES'/'0      XC ='F7.4/'       YC =',F7.4/
     X           '0      PC =',1PD23.16/'       TC =',1PD23.16/          ZNP05730
     X           '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X           '0      XF =',F7.4/'       YF =',F7.4//
     X           '       L0 =',1PD23.16/'       R0 =',1PD23.16////)      ZNP05740
      NTOT = NM1 - 1
      NENV = NTOT - NPIT
      WRITE(3,30) NTOT, NPIT, NENV
      STOP                                                               ZNP05790
      END                                                                ZNP05800
/*
/EXEC    LDR
/DD  8   DSNAME=TEMP,DISP=PASS
/DD  9   DSNAME=TEST,DEVT=(2314,FILPRO),RECFM=CI,DISP=KEEP
/EXEC    *
/DD  8   DSNAME=TEMP,DISP=SCR
/EOJ
/EOJ
/EOJ
/*
```

F.5 CE. PU. START

This is a version of the main hydrodynamic code utilizing subroutines which read the initial input model, produce velocity perturbations, and otherwise initialize code parameters and data sets. The subroutines employed are from CEPHEI LIBRARY (F.7).

```
/ID   1444   T UNGAR,S. CE. PU. START      020          02   005
/DD  100 DSNAME=BOMB,DEVT=2314   JOB BOMBS IF I'M MULTI-PROGRAMMING....RESUBMIT.
/DD  50  DSNAME=MODL,DISP=SCR
/DD  91  DSNAME=TLUM,DISP=SCR
/DD  55  DSNAME=TNGY,DISP=SCR
/DD  66  DSNAME=ZNGY,DISP=SCR
/DD  77  DSNAME=TPLT,DISP=SCR
/DD  8   DSNAME=SAVE,DISP=SCR
/DD  12  DSNAME=LUM,DISP=SCR
/DD  11  DSNAME=ENGY,DISP=SCR
/DD  21  DSNAME=PLOT,DISP=SCR
/EXEC    FORTX,'OPT=2'              15 SOLAR MASS MODEL   HALF ZONING
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      DIMENSION RHOZM1(605), Q(605)
      DIMENSION F(605,5), G(605,5), H(605,5)
      INTEGER*4 ILUM(12) /2,4,6,8,12,16,20,24,28,32,36,40/
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,U,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,HATOM,VCOEF,NMODEL
      N= 281
      NCON = 201
      NSURF = N - 2
      N = 281
      NCON = 177
      N = 351
      NCON =223
      N = 231
      N = 291
      N = 301
      NCON = 246
      NCON = 164
      N = 221
      NCON = 164
      NSUBZN = 5
      NRZNF1 = N - 21
      N = 215
      NCON = 171
      NSUBZN = 3
      NRZNF1 = N - 15
      N = 217
      NRZNF1 = N - 17
      N = 224
      N = 229
      NSUBZN = 1
      NRZNF1 = N
      NSURF = N
      VCOEF = 1.D3
      VCOEF = 1.D-1
      VCOEF = 1.D-2
      VCOEF = 1.D1
      VCOEF = 7.D1
      VCOEF = 1.D0
      VCOEF = 2.D1
```

```
      VCOEF = 2.D2
      VCOEF = 1.D2
      VCOEF = 5.D0
      NM1 = N - 1
      NM10 = N - 10
      NP1 = N + 1
      NMODEL = 14
      NMODEL = 15
      NMODEL = 16
      XC = 0.70D0
      YC = 0.27D0
      XF = 0.35D0
      YF = 0.62D0
      NF = 133
      NCON = 106
      NCNO = NF - 1
      NCONM1 = NCON - 1
      CALL CONSET(XC,YC,XF,YF,NF,NCONM1)
      CALL STAR1(NRZNE1)
      CALL PUSH(-6.D6)
      DO 101 J=1,12
  101 ILUM(J) = N - ILUM(J)
      CALL LUMSET(ILUM)
      CALL BNDSET(XF,YF)
      DO 6 KK=1,301
      DO 4 K=2,5
      A = ((K-1)/2)/2.D0
      Z2 = Z1 + DZ*A
      DO 1 I=2,N
      V(I,2) = V(I,1) + F(I,K-1)*A
      R(I,2) = R(I,1) + G(I,K-1)*A
    1 CONTINUE
      DO 2 I=1,NM1
      S(I,2) = S(I,1) + H(I,K-1)*A
      RHOZM1(I) = RHO(I)
      RHO(I) = DM(2 *I)/(FPITHD*(R(I+1,2)**3 - R(I,2)**3))
      Q(I) = 0.D0
      IF(V(I+1,2) .LT. V(I,2))
     X    Q(I) = VCOEF*(V(I+1,2)-V(I,2))**2/(1/RHO(I)+1/RHOZM1(I))
    2 CONTINUE
      CALL THERMO
      CALL ECNO(NCNO)
      CALL OPACTY(NCON)
      CALL LUMRAD(NCON,NSUBZN,NRZNE1)
      CALL LUMCON(NCON-2)
      CALL BNDRY
      IF(K .NE.2)  GO TO 25
      CALL STEPSI(DZ)
      CALL OUTPUT(NCON,NSURF,Z1,DZ,150,1,25,NM10)
      CALL PULSEN(Z1,NSURF,'D')
      CALL LUMSAV(Z1,25)
   25 CONTINUE
      NREZNE = NRZNE1
      DO 3 I=2,N
      J = I
      IF(I .NE. NREZNE) GO TO 27
```

```
      IF(NREZNE .EQ. NM1) GO TO 27
      NREZNE = NREZNE + NSUBZN - 1
      J = I + NSUBZN/2
   27 CONTINUE
      F(I,K) = -(U*M(2*I-1)/R(I,2)**2 + FPI*R(I,2)**2*(P(J)-P(I-1)
     X           +Q(J)-Q(I-1))/DM(2*I-1))*DZ
      G(I,K) = V(I,2) * DZ
    3 CONTINUE
      DO 4 I=1,NM1
      H(I,K) = (E(I) - (L(I+1)-L(I))/DM(2*I))*DZ/T(I)
      H(I,K) = H(I,K) - FPI*Q(I)*(R(I+1,2)**2*V(I+1,2)
     X                 - R(I,2)**2*V(I,2))/(DM(2*I))*DZ/T(I)
    4 CONTINUE
      Z1 = Z1 + DZ
      DO 5 I=2,N
      V(I,1) = V(I,1) + (F(I,2)+2.D0*F(I,3)+2.D0*F(I,4)+F(I,5))/6.D0
      R(I,1) = R(I,1) + (G(I,2)+2.D0*G(I,3)+2.D0*G(I,4)+G(I,5))/6.D0
    5 CONTINUE
      DO 6 I=1,NM1
      S(I,1) = S(I,1) + (H(I,2)+2.D0*H(I,3)+2.D0*H(I,4)+H(I,5))/6.D0
    6 CONTINUE
      CALL STORE(Z1)
      STOP
      END
/*
/JOBLIB  DSNAME=LIBP,DISP=PASS            CEPHEID LIBRARY
/EXEC    LDR
/DD   8  DSNAME=SAVE,DEVT=2314,RECFM=F,LRECL=56,DISP=KEEP
/DD   9  DSNAME=TEST
/DD  11  DSNAME=ENGY,DEVT=(2314,FILPRO),SPACE=1000,RECFM=F,LRECL=56,DISP=KEEP
/DD  12  DSNAME=LUM,DEVT=(2314,FILPRO),RECFM=F,LRECL=52,DISP=KEEP
/DD  22  DSNAME=PLOT,DEVT=(2314,FILPRO),RECFM=F,LRECL=24,DISP=KEEP
/DD  50  DSNAME=MODL,DEVT=(2314,FILPRO),RECFM=F,LRECL=128,DISP=KEEP
/DD  31  DEVT=2250
/EXEC    *
/*
```

F.6 CEPH. PULS.

This is the main hydrodynamic code employed in its customary fashion. It uses the current status of the model for initial values, updating the current status data set, while extending the data sets which store time varying information. The subroutines employed are from CEPHEI LIBRARY (F.7).

```
/ID   1444   T UNGAR,S. CEPH. PULS.
/DD  99  DSNAME=PLS4,DISP=SCR
/EXEC    *
/DD  7   DSNAME=PLS4,DEVT=(2314,FILPRO),RECFM=CI,DISP=KEEP
/EXEC    PRPU
/DD  5   DISP=


 EXEC    COPIER,'1,F,M'
/DD  8   DSNAME=PLOT
/DD  9   DSNAME=TPLT,DEVT=(2314,FILPRO),RECFM=F,LRECL=24,DISP=PASS
/EXEC    COPIER,'1,F,M'
/DD  8   DSNAME=ENGY
/DD  9   DSNAME=TNGY,DEVT=(2314,FILPRO),SPACE=1000,RECFM=F,LRECL=56,DISP=PASS
/EXEC    COPIER,'1,F,M'
/EXEC    FORTX,'OPT=2'              15 SOLAR MASS MODEL   HALF ZONING
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      DIMENSION RHOZM1(605), Q(605)
      DIMENSION F(605,5), G(605,5), H(605,5)
      INTEGER*4 ILUM(12) /2,4,6,8,12,16,20,24,28,32,36,40/
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,U,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X               ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,HATOM,VCOEF,NMODEL
      NSUBZN = 1
      N = 229
      NRZNE1 = N
      NSURF = N
      NM1 = N - 1
      NM10 = N - 10
      NP1 = N + 1
      NMSUN = 15
      XC = 0.7D0
      YC = 0.27D0
      XF = 0.35
      YF = 0.62
      MFFRAC = 0.66
      NF = 133
      NCON = 106
      NCONM1 = NCON - 1
      NCNO = NF - 1
      WRITE(3,99) NMSUN, XC, YC, PC, TC, MFFRAC, XF, YF, L0, R0
  99 FORMAT(///19,' SOLAR MASSES'/'0      XC ='F7.4/'       YC =',F7.4/
     X           '0      PC =',1PD23.16/'       TC =',1PD23.16/
     X           '0 MFFRAC =',0PF5.2,' OF THE STAR''S MASS'/
     X           '0      XF =',F7.4/'       YF =',F7.4//
     X           '       L0 =',1PD23.16/'       R0 =',1PD23.16////)
      CALL CONSET(XC,YC,XF,YF,NF,NCONM1)
      CALL STAR2(Z1,NRZNE1)
      DO 101 J=1,12
  101 ILUM(J) = N - ILUM(J)
      CALL LUMSET(ILUM)
      CALL BNDSET(XF,YF)
```

```
      DO 6 KK=1,401
      DO 4 K=2,5
      A = ((K-1)/2)/2.D0
      Z2 = Z1 + DZ*A
      DO 1 I=2,N
      V(I,2) = V(I,1) + F(I,K-1)*A
      R(I,2) = R(I,1) + G(I,K-1)*A
    1 CONTINUE
      DO 2 I=1,NM1
      S(I,2) = S(I,1) + H(I,K-1)*A
      RHOZM1(I) = RHO(I)
      RHO(I) = DM(2*I)/(FPITHD*(R(I+1,2)**3 - R(I,2)**3))
      Q(I) = 0.D0
      IF(V(I+1,2) .LT. V(I,2))
     X    Q(I) = VCOEF*(V(I+1,2)-V(I,2))**2/(1/RHO(I)+1/RHOZM1(I))
    2 CONTINUE
      CALL THERMO
      CALL RCNO(NCNO)
      CALL OPACTY(NCON)
      CALL LUMRAD(NCON,NSUBZN,NRZNE1)
      CALL LUMCON(NCON-2)
      CALL BNDRY
      IF(K .NE.2)  GO TO 25
      CALL PULSEN(Z1,NSURF,'D')
      CALL LUMSAV(Z1,25)
      CALL STEPSI(DZ)
      CALL OUTPUT(NCON,NSURF,Z1,DZ,200,1,25,NM10)
   25 CONTINUE
      NREZNE = NRZNE1
      DO 3 I=2,N
      J = I
      IF(I .NE. NREZNE) GO TO 27
      IF(NREZNE .EQ. NM1) GO TO 27
      NREZNE = NREZNE + NSUBZN - 1
      J = I + NSUBZN/2
   27 CONTINUE
      F(I,K) = -(U*M(2*I-1)/R(I,2)**2 + FPI*R(I,2)**2*(P(J)-P(I-1)
     X           +Q(J)-Q(I-1))/DM(2*I-1))*DZ
      G(I,K) = V(I,2) * DZ
    3 CONTINUE
      DO 4 I=1,NM1
      H(I,K) = (P(I) - (L(I+1)-L(I))/DM(2*I))*DZ/T(I)
      H(I,K) = H(I,K) - FPI*Q(I)*(R(I+1,2)**2*V(I+1,2)
     X                - R(I,2)**2*V(I,2))/(DM(2*I))*DZ/T(I)
    4 CONTINUE
      Z1 = Z1 + DZ
      DO 5 I=2,N
      V(I,1) = V(I,1) + (F(I,2)+2.D0*F(I,3)+2.D0*F(I,4)+F(I,5))/6.D0
      R(I,1) = R(I,1) + (G(I,2)+2.D0*G(I,3)+2.D0*G(I,4)+G(I,5))/6.D0
    5 CONTINUE
      DO 6 I=1,NM1
      S(I,1) = S(I,1) + (H(I,2)+2.D0*H(I,3)+2.D0*H(I,4)+H(I,5))/6.D0
    6 CONTINUE
      CALL STORE(Z1)
      ENDFILE 69
      STOP
```

```
      END
/*
/JOBLIB  DSNAME=LIBP,DISP=PASS              CEPHEID LIBRARY
/EXEC    LDR
/DD  8   DSNAME=SAVE,DISP=PASS
/DD  11  DSNAME=TNGY,DISP=KEEP
/DD  12  DSNAME=TLUM,DISP=KEEP
/DD  22  DSNAME=TPLT,DISP=KEEP
/DD  69  DSNAME=6969,DEVT=2314,DISP=KEEP
/EXEC    *
/JOBLIB  DUMMY(1,POP)
/DD  69  DSNAME=6969
/DD  88  DSNAME=LUM,DISP=SCR
/DD  44  DSNAME=PLOT,DISP=SCR
/DD  77  DSNAME=ENGY,DISP=SCR
/EXEC    EXIT
/DD  8   DSNAME=TLUM
/DD  9   DSNAME=LUM,DEVT=(2314,FILPRO),SPACE=1000,RECFM=F,LRECL=52,DISP=KEEP
/EXEC    COPIER,'1,F,C'
/DD  88  DSNAME=TLUM,DISP=SCR
/DD  8   DSNAME=TPLT
/DD  9   DSNAME=PLOT,DEVT=(2314,FILPRO),RECFM=F,LRECL=24,DISP=KEEP
/EXEC    COPIER,'1,F,C'
/DD  99  DSNAME=TPLT,DISP=SCR
/DD  8   DSNAME=TNGY
/DD  9   DSNAME=ENGY,DEVT=(2314,FILPRO),SPACE=1000,RECFM=F,LRECL=56,DISP=KEEP
/EXEC    COPIER,'1,F,C'
/DD  66  DSNAME=TNGY,DISP=SCR
/DD  69  DSNAME=6969,DISP=SCR
/DD  2   SYSIN

/EOJ
/*
```

F.7 CEPHEI LIBRARY

This package contains the library of subroutines used by CEPH. PULS. (F.6) and other main code related routines. The function of each subroutine is briefly stated below.

(a) CONSET - initializes physical and mathematical constants in the main code and other subroutines

(b) STAR1 - introduces values from the initial equilibrium model into the code

(c) PUSH
(d) AMPUSH
(e) VOPUSH
} - produce initial velocity perturbations

(f) STAR2 - reinitializes the code with values computed at the end of the previous run

(g) QCALC - indicates where and when the artificial viscosity is turned on; and supplies information about pertinent physical parameters at these points.

(h) QCALC2 - similar to QCALC, but compares viscous pressure gradient to total pressure gradient

(i) THERMO (entry THMSET) - computes pressure and temperature distributions from current entropy and density distributions

(j) ECNO - calculates nuclear energy generation rate

(k) OPASET (entry OPACTY) - calculates either total or electron scattering opacity depending on the calling arguments

(l) LUMRAD - calculates luminosity in the radiative envelope

(m) LUMCON - solves for luminosity in the convective core by the method indicated in section (3.4)

(n) LUMSAV - stores values of luminosity at selected points in the model on a disk data set for future reference (e.g. for generating plots)

(o) BNDSET (entry BNDRY) - sets up the photospheric boundary conditions

(p) STEPSI - computes stepsize for time integration by considering the Courant and thermal conditions

(q) DZCALC - similar to STEPSI, but indicates how and where the minimum stepsize condition is met

(r) GAMMA1 - calculates the value of the adiabatic exponent Γ_1

(s) OUTPUT - controls nature and format of output information

(t) PLTOUT - stores values of r, v, and L at selected times and spatial points for later plotting

(u) STORE - stores those model values at the end of a run which are necessary to initialize the code (through STAR2) for a subsequent run

(v) PULSEN - computes the spatially integrated values of kinetic, nuclear and gravitational energies at each time step and stores them along with the surface values of radius, velocity and luminosity on magnetic disk or tape, as specified

(w) TAPSET - positions magnetic tape for addition of PULSEN data

(x) CALPRT - prints the output of QCALC, QCALC2, and DZCALC from intermediate disk storage

```
/ID   1444   T UNGAR,S. CEPHEID LIBRARY    005          02   000
/DD  99  DSNAME=LIBP,DISP=SCR       SCRATCH OLD VERSION OF LIBP LIBRARY
/EXEC    EXIT     DUMMY STEP TO PERMIT EXECUTION OF PREVIOS INSTRUCTION
/DD  7   DSNAME=LIBP,DEVT=(2314,FILPRO),RECFM=F,LRECL=80,DISP=PASS
/DD  5   UPD(2,0,1)
/DD  1   DUMMY(80)
/DD  9   DEVT=CORE
/EXEC    FORTX,'DECK,OPT=2,XREF'   BASED ON LIBP  VERSION VIII  2/20/70
.NOPRT
.NUM                                                                  LIB00000
C
C     REVISED 6/9/70 TO ACCOMODATE CUMULATIVE STEP NUMBER AND MORE
C     FLEXIBLE ZONE CHOICE FOR LUMSAV.  THE FOLLOWING SUBROUTINES HAVE
C     BEEN AFFECTED - STAR1, STAR2, OUTPUT, STORE, LUMSAV (LUMSET ENTRY
C     ADDED), PLTOUT, BNDRY (PHOTOSPHERIC B.C.'S).....
C
C
C     REVISED 5/12/70 ... TO ADJUST DIFFERENCE SCHEME TO ALLOW FOR MORE
C     FLEXIBLE MASS ZONING.  SUBROUTINES STAR1 AND STAR2 AFFECTED.
C
C



      SUBROUTINE CONSET(XC,YC,XF,YF,NF,NCON)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KBOLTZ,KAPPA
      DIMENSION XCHEM(605), YCHEM(605), ZCHEM(605)
      DIMENSION CON1(605), CON3(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H
      X = XC
      Y = YC
      PI = 3.1415926535897932D0
      FPI = 4.D0*PI
      FPITHD = FPI/3.D0
      PICON = 1/(4.D0*FPITHD)
      A = 7.5641D-15
      A0 = 2.81765D-13
      C = 2.997929D10
      G = 6.673D-8
      H = 1.67333D-24
      KBOLTZ = 1.38046D-16
      RSUN = 6.9598D10
      MSUN = 1.989D33
      LSUN = 3.90D33
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      SIGMA = A*C/4.D0
      C1 = KBOLTZ*MUINV/H
      C2 = 4.D0*A/3.D0
      C3 = 3.D0*C1
      C4 = 4.D0*A
      C5 = A/3.D0
      OPCON = 32.D0*A*C*PI**2/3.D0
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)
```

```
      Z = 1.D0 - X - Y
      XCNO = (1.D0 - X - Y)/2.D0
      ECONST = 7.9D31*X*XCNO
      EXP1 = 1.D0/3.D0
      EXP2 = 2.D0/3.D0
      FRAC = 8.D0/3.D0
      GCONST = G/FPI
      DO 10 I=1,NF
      XCHEM(I) = XC
      YCHEM(I) = YC
      ZCHEM(I) = 1.D0 - XCHEM(I) - YCHEM(I)
      CON1(I) = (0.5D0 + 1.5D0*XCHEM(I) + 0.25D0*YCHEM(I))*KBOLTZ/H
      CON3(I) = 3.D0*CON1(I)
   10 CONTINUE
      DO 20 I=NF,N
      XCHEM(I) = XF
      YCHEM(I) = YF
      ZCHEM(I) = 1.D0 - XCHEM(I) - YCHEM(I)
      CON1(I) = (0.5D0 + 1.5D0*XCHEM(I) + 0.25D0*YCHEM(I))*KBOLTZ/H
      CON3(I) = 3.D0*CON1(I)
   20 CONTINUE
      CALL THMSET(CON1,CON3)
      CALL OPASET(NCON,XCHEM,YCHEM,ZCHEM)
      RETURN
      END


      SUBROUTINE STAR1(NREZNE)
C
C     REVISED 5/12/70 ... TO ADJUST DIFFERENCE SCHEME TO ALLOW FOR MORE
C     FLEXIBLE MASS ZONING.
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)                                    LIB00430
      REAL*8 KAPPA                                                      LIB00440
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)                      LIB00450
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)                  LIB00460
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1     LIB00470
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,   LIB00480
     X          ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,HATOM,VCOEF,NMODEL,NTSTEP
  300 FORMAT(8A8)                                                       LIB00500
      NTSTEP = -1
    1 I = I + 1                                                         LIB00510
      J = I/2 + 1                                                       LIB00520
      K = (I + 1)/2                                                     LIB00530
      READ(9,300,END=4) M(I),R(J,1),RHO(K),P(K),T(K),S(K,1),L(J),E(K)   LIB00540
      NTEST = NTEST + 1                                                 LIB00550
      GO TO 1                                                           LIB00560
    4 CONTINUE                                                          LIB00570
C     THESE VALUES INSERTED TO ALLOW ELECTRON SCATTERING OPACITY TO BE  LIB00580
C     INAUGURATED BY CALLING OPACTY(N+1).    VERSION II, 8/26/69        LIB00590
      RHO(N+1) = RHO(N)                                                 LIB00600
      T(N+1) = T(N)                                                     LIB00610
      NVALS = 2*N - 1                                                   LIB00620
      IF(NVALS .EQ. NTEST) GO TO 5                                      LIB00630
      WRITE(3,200) NVALS, NTEST                                         LIB00640
  200 FORMAT(//' E R R O R   D U R I N G   C A L L   T O   S T A R 1'// LIB00650
```

```
     X       ' NUMBER OF STARTING VALUES REQUIRED BY CODE =',I4//       LIB00660
     X        ' NUMBER OF STARTING VALUES READ BY ROUTINE =',I4////)    LIB00670
      CALL EXIT                                                         LIB00680
    5 CONTINUE                                                          LIB00690
      NM1$2 = NM1 * 2                                                   LIB00700
      DO 20 I=2,NM1$2                                                   LIB00710
      DM(I) = M(I+1) - M(I-1)                                           LIB00720
C
C     THE FOLLOWING TEST ALLOWS THIS ROUTINE TO BE USED WITH ZONES OF
C     CONTINUOUSLY VARYING MASS SIZE.
C
      IF(NREZNE .GE. N)   GO TO 20
C
      IF(M(I+1) - M(I) .NE. M(I) - M(I-1))  DM(I) = 2.D0*(M(I) - M(I-1))
   20 CONTINUE                                                          LIB00730
      DM(2*N-1) = 2.D0*(M(2*N-1) - M(2*N-2))                            LIB00740
      WRITE(50,500) NM1, M(2*N-1), NMODEL, VCOEF
  500 FORMAT(I4,' ZONES  -  MASS =',1PE10.3,' GM  -  A CEPHEID MODEL NUM
     XBER',I3,'  -  VISCOSITY COEF =',1PE8.1,
     X       '  -  COMPLETE OPACITY')
      REWIND 50
      RETURN                                                            LIB00790
      END                                                               LIB00800


      SUBROUTINE PUSH(VSURF)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      PROCON = VSURF/R(N,1)
      DO 10 I=1,N
      V(I,1) = PROCON*R(I,1)
      V(I,2) = V(I,1)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE AMPUSH(PERIOD,FACTOR)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      REAL*4 XX(500), YY(500)
      DIMENSION Q(500), A(500)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,U,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,HATOM
   10 FORMAT(2F10.4)
      SIGMA = 2.D0*PI/PERIOD
   20 K = K + 1
      READ(2,10,END=50)  Q(K), A(K)
      GO TO 20
   50 CONTINUE
```

```
      Q(K) = Q(K-1) + 1.D0
      A(K) = A(K-1) + 1.D0
      K = 0
      DO 30 I=1,N
      MFRAC = M(2*I-1)/M(2*N-1)
      XX(I) = MFRAC
   40 K = K + 1
      IF(MFRAC .GE. Q(K)) GO TO 40
      K = K - 1
      RAMPL = A(K) + (MFRAC - Q(K))*(A(K+1) - A(K))/(Q(K+1) - Q(K))
      YY(I) = RAMPL
   30 V(I,1) = FACTOR*SIGMA*RAMPL*R(I,1)
      CALL GWINK
      CALL GTITLE('INITIAL RADIUS AMPLITUDE VS MASS FRACTION',41,7673)
      CALL GPLOT(XX,YY,N,7673)
      DO 60 I=1,N
   60 YY(I) = V(I,1)
      CALL GTITLE('INITIAL VELOCITY VS MASS FRACTION - 100MSUN MODEL',
     X            49,7673)
      CALL GPLOT(XX,YY,N,7673)
      RETURN
      END


      SUBROUTINE V0PUSH(PERIOD,FACTOR)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      REAL*4 XX(500), YY(500)
      DIMENSION Q(500), A(500)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,U,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,HATOM
   10 FORMAT(2F10.4)
      SIGMA = 2.D0*PI/PERIOD
   20 K = K + 1
      READ(2,10,END=50)  Q(K), A(K)
      GO TO 20
   50 CONTINUE
      Q(K) = Q(K-1) + 1.D0
      A(K) = A(K-1) + 1.D0
      VFAC = FACTOR*SIGMA*R(N,1)/A(K-1)
      K = 0
      DO 30 I=1,N
      MFRAC = M(2*I-1)/M(2*N-1)
      XX(I) = MFRAC
   40 K = K + 1
      IF(MFRAC .GE. Q(K)) GO TO 40
      K = K - 1
      AVAL = A(K) + (MFRAC - Q(K))*(A(K+1) - A(K))/(Q(K+1) - Q(K))
      YY(I) = AVAL
   30 V(I,1) = VFAC*AVAL
      CALL GDATE
      CALL GWINK
      CALL GWRITE(50,64)
```

```
      CALL GTITLE('UNSCALED VELOCITY VALUES VS MASS FRACTION',41,7673)
      CALL GPLOT(XX,YY,N,7673)
      DO 60 I=1,N
   60 YY(I) = V(I,1)
      CALL GWRITE(50,64)
      CALL GTITLE('INITIAL VELOCITY VS MASS FRACTION - 15MSUN MODEL',
     X            48,7673)
      CALL GPLOT(XX,YY,N,7673)
      RETURN
      END


      SUBROUTINE STAR2(Z,NREZNE)
C
C     REVISED 5/12/70 ... TO ADJUST DIFFERENCE SCHEME TO ALLOW FOR MORE
C     FLEXIBLE MASS ZONING.
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)                                    LIB01690
      REAL*8 KAPPA                                                      LIB01700
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)                      LIB01710
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)                  LIB01720
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1     LIB01730
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X          ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,HATOM,VCOEF,NMODEL,NTSTEP
  300 FORMAT(7A8)                                                       LIB01740
  500 FORMAT(I4,' ZONES  -  MASS =',1PE10.3,' GM  -  A CEPHEID MODEL NUM
     XBER',I3,'  -  VISCOSITY COEF =',1PE8.1,
     X       '  -  COMPLETE OPACITY')
      READ(8,300) Z, VCOEF, NTSTEP, NMODEL                              LIB01780
      READ(8,300) (M(2*I-1), M(2*I), R(I,1), V(I,1), S(I,1), RHO(I),    LIB01790
     X             T(I), I=1,N)                                         LIB01800
C     THESE VALUES INSERTED TO ALLOW ELECTRON SCATTERING OPACITY TO BE  LIB01810
C     INAUGURATED BY CALLING OPACTY(N+1).     VERSION II, 8/26/69       LIB01820
      RHO(N+1) = RHO(N)                                                 LIB01830
      T(N+1) = T(N)                                                     LIB01840
      NM1$2 = NM1 * 2                                                   LIB01850
      DO 20 I=2,NM1$2                                                   LIB01860
      DM(I) = M(I+1) - M(I-1)                                           LIB01870
C
C     THE FOLLOWING TEST ALLOWS THIS ROUTINE TO BE USED WITH ZONES OF
C     CONTINUOUSLY VARYING MASS SIZE.
C
      IF(NREZNE .GE. N)   GO TO 20
C
      IF(M(I+1) - M(I) .NE. M(I) - M(I-1))  DM(I) = 2.D0*(M(I) - M(I-1))
   20 CONTINUE                                                          LIB01880
      DM(2*N-1) = 2.D0*(M(2*N-1) - M(2*N-2))                            LIB01890
      RETURN                                                            LIB01900
      END                                                               LIB01910


      SUBROUTINE QCALC(Z,RHOZM1,Q,VCOEF)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      DIMENSION RHOZM1(605), Q(605)
      REAL*8 KAPPA
```

```
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
   97 FORMAT(/'-MODEL AT TIME =',1PD23.16,' SECONDS')
   98 FORMAT( '0VISCOSITY WAS TURNED ON IN THE FOLLOWING ZONES, WITH THE
     X FOLLOWING VALUES'      /                              '0VISCOUS COEF
     XFICENT =',1PD11.4/'0  ZONE NUMBER',07X,'MASS',14X,'RADIUS',13X,
     X 'VELOCITY',13X,'DEL-VEL',12X,'PRESSURE',11X,'VISCOSITY'/)
   99 FORMAT(I10,1P6D20.9)
      NDSRN = 90
      ISWTCH = 1
      DO 2 I=1,NM1
      Q(I) = 0.D0
      IF(V(I+1,2) .GE. V(I,2)) GO TO 2
      Q(I) = VCOEF*(V(I+1,2)-V(I,2))**2/(1/RHO(I)+1/RHOZM1(I))
      IF(ISWTCH .NE. 1) GO TO 1
      WRITE(NDSRN,97) Z
      WRITE(NDSRN,98) VCOEF
      ISWTCH = 2
    1 CONTINUE
      DELV = V(I+1,2) - V(I,2)
      WRITE(NDSRN,99) I, M(2*I), R(I,2), V(I,2), DELV, P(I), Q(I)
    2 CONTINUE
      RETURN
      END


      SUBROUTINE QCALC2(Z,Q,N1,N2,VCOEF)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      DIMENSION Q(1)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
   97 FORMAT(/'-MODEL AT TIME =',1PD23.16,' SECONDS')
   98 FORMAT('0VISCOSITY WAS CALCULATED IN THE FOLLOWING ZONES, WITH THE
     X FOLLOWING VALUES'      /                              '0VISCOUS COEF
     XFICENT =',1PD11.4/'0  ZONE NUMBER',07X,'MASS',14X,'RADIUS',13X,
     X                 'DEL-VIS',12X,'DEL-PRES',11X,'ACC. TERM'/)
   99 FORMAT(I10,1P6D20.9)
      IF(ISWTCH .EQ. 1) GO TO 1
      NDSRN = 90
      PI = 3.1415926535897932D0
      FPI = 4.D0*PI
      G = 6.673D-8
      GCONST = G/FPI
      NBEG = N1
      NFIN = N2
      ISWTCH = 1
    1 CONTINUE
      WRITE(NDSRN,97) Z
      WRITE(NDSRN,98) VCOEF
      DO 10 I=NBEG,NFIN
      DQ = Q(I) - Q(I-1)
      DP = P(I) - P(I-1)
      DV = DP + GCONST*M(2*I-1)*DM(2*I-1)/R(I,2)**4
```

```
      WRITE(NDSRN,99) I, M(2*I-1), R(I-1,2), DQ, DP, DV
   10 CONTINUE
      RETURN
      END


      SUBROUTINE THERMO
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      DIMENSION C1(1), C3(1)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,D1,C2,D3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H
      F(T) = C1(I)*DLOG(T**1.5D0/RHO(I)) + C2*T**3/RHO(I) - S(I,2)
      DF(T) = C3(I)/(2.D0*T) + C4*T**2/RHO(I)
      DO 20 I=1,NM1
   10 TT = T(I)
      T(I) = TT - F(TT)/DF(TT)
      IF(DABS(T(I)-TT) .GT. 1.D2) GO TO 10
      P(I) = C1(I)*RHO(I)*T(I) + C5*T(I)**4
   20 CONTINUE
      ENTRY THMSET(C1,C3)
      RETURN
      END


      SUBROUTINE ECNO(NCNO)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H
      DO 10 I=1,NCNO
      E(I) = ECONST*RHO(I)*DEXP(-1.5231D4/T(I)**EXP1)/T(I)**EXP2
   10 CONTINUE
      RETURN
      END


      SUBROUTINE OPASET(NCON,X,Y,Z)
C
      IMPLICIT REAL*8(A-H,K-M,O-Z,$)
      DIMENSION X(1), Y(1), Z(1)
      DIMENSION ALPHAS(605), ALPHAX(605), ALPHAY(605), ALPHAZ(605)
      DIMENSION ZCON(605)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,DUMALF,
     X              ECONST,EXP1,EXP2,FRAC,DUMX,DUMY,DUMZ,A0,H
      KBOLTZ = 1.38046D-16
      DO 10 I=1,NCON
```

```
      KAPPA(I) = (FPI/3.D0)*A0**2*(1.D0 + X(I))/H
   10 CONTINUE
      KAPPA(N) = (FPI/3.D0)*A0**2*(1.D0 + X(N))/H
      DO 20 I=NCON,NM1
      ALPHA = KBOLTZ*(1.D0 + X(I))/(2.D0*H)
      ALPHAS(I) = ALPHA*1.D4*4.85D-13/2.2D-5
      ALPHAX(I) = ALPHA*1.D4*X(I)/2.1D0
      ALPHAY(I) = ALPHA*1.D4*Y(I)
      ALPHAZ(I) = ALPHA*1.D4*Z(I)/35.D0
      ZCON(I) = 3.8D2*Z(I)*((2.D0 - Z(I))**2*(1 + X(I)))**0.33D0/35.D0
   20 CONTINUE
      OC0 = 1.D0/2.2D-5
      OC1 = 2.D6/2.1D0
      OC2 = 1.D0/4.5D0
      OC3 = OC2*2.1D0
      OC4 = 20.D0/35.D0
      RETURN
      ENTRY OPACTY
      DO 30 I=NCON,NM1
      T4 = T(I)/1.D4
      RTT4 = DSQRT(T4)
      T4SQ = T4*T4
      T4CU = T4SQ*T4
      T4FIV = T4CU*T4SQ
      KAPPAS = ALPHAS(I)/(OC0 + T4)
      KAPPAX = ALPHAX(I)*(RTT4/(OC1/T4FIV + T4FIV) + OC3/(T4FIV + OC2
     X          /(4.D-3/T4SQ + 2.D-4*T4SQ/RHO(I)**0.25D0)))
      KAPPAY= ALPHAY(I)*(1.D0/(1.4D3 + T4FIV) + 15.D0/(1.D7/T4 + T4FIV))
      KAPPAZ = ALPHAZ(I)*RTT4/(OC4 + T4CU + ZCON(I)*RHO(I)**0.33D0
     X             *T4**3.71D0)
      KAPPA(I) = KAPPAS + (KAPPAX + KAPPAY + KAPPAZ)*RHO(I)
   30 CONTINUE
      RETURN
      END


      SUBROUTINE LUMRAD(NCON,NSUBZN,NRZNE1)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)                                    LIB02920
      REAL*8 KAPPA                                                      LIB02930
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)                      LIB02940
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)                  LIB02950
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1     LIB02960
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,   LIB02970
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H                    LIB02980
      NREZNE = NRZNE1
      DO 10 I=NCON,NM1                                                  LIB02990
      J = I
      OPCON2 = OPCON/(KAPPA(I)+KAPPA(I-1))                              LIB03000
      IF(I .NE. NREZNE) GO TO 27
      IF(NREZNE .EQ. NM1) GO TO 27
      NREZNE = NREZNE + NSUBZN - 1
      J = I + NSUBZN/2
   27 CONTINUE
      L(I) = OPCON2*R(I,2)**4*(T(I-1)**4 - T(J)**4)/DM(2*I-1)           LIB03010
   10 CONTINUE                                                          LIB03020
```

```
      RETURN                                                            LIB03030
      END                                                               LIB03040


      SUBROUTINE LUMCON(NPIVPT)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      DIMENSION A(1000,4)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H
      DO 10 I=1,NPIVPT
      COEF = DLOG(T(I)/T(I+1))/2.D0
      A(I,1) = 1.D0 - COEF
      A(I,2) = -2.D0
      A(I,3) = 1.D0 + COEF
      A(I,4)= (T(I) + T(I+1))*(E(I+1)/T(I+1) - E(I)/T(I))*DM(2*I+1)/2.D0
   10 CONTINUE
      A(NPIVPT,4) = A(NPIVPT,4) - A(NPIVPT,3)*L(NPIVPT+2)
      NPIVM1 = NPIVPT - 1
      DO 20 I=2,NPIVM1
      A(I,2) = A(I,2)*A(I-1,2) - A(I,1)*A(I-1,3)
      A(I,3) = A(I,3)*A(I-1,2)
   20 A(I,4) = A(I,4)*A(I-1,2) - A(I,1)*A(I-1,4)
      L(NPIVPT+1) = (A(NPIVPT,4)*A(NPIVM1,2) - A(NPIVPT,1)*A(NPIVM1,4))/
     X              (A(NPIVPT,2)*A(NPIVM1,2) - A(NPIVPT,1)*A(NPIVM1,3))
      DO 30 I=1,NPIVM1
      J = NPIVPT - I
   30 L(J+1) = (A(J,4) - A(J,3)*L(J+2)) / A(J,2)
      RETURN
      END


      SUBROUTINE LUMSAV(Z,NSTEP)
C
C     ALTERED 6/9/70 TO ALLOW FOR CHOICE OF ZONES WHERE DATA IS GOTTEN
C     BY INDICATING WHERE IN LUMSET.  LUM DATA SET WILL ONLY CONTAIN
C     MASS ZONING DATA IF LUMSAV IS CALLED DURING FIRST 100 MODELS OR
C     IF LUMSAV IS NEVER REFERRED TO....
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      DIMENSION MFRAC(100), ILUM(100), ILUMIN(1)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X          ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,HATOM,VCOEF,NMODEL,NTSTEP
  100 FORMAT(13A4)
      NCOUNT = NCOUNT - 1
      IF(NCOUNT .GT. 0) RETURN
      NCOUNT = NSTEP
      IF(ISWTCH .EQ. 1) GO TO 10
      JDEL = N/12
      JFIN = 12*JDEL
      DO 5 K=1,12
```

```
      IF(ISWTCH .NE. 2)   ILUM(K) = JDEL*K
      MFRAC(K) = M(2*ILUM(K)-1)/M(2*N-1)
    5 CONTINUE
      WRITE(12,100) M(2*N-1), (MFRAC(K),K=1,12)
      ISWTCH = 1
   10 CONTINUE
      WRITE(12,100) Z, (L(ILUM(J)),J=1,12)
      RETURN
      ENTRY LUMSET(ILUMIN)
      DO 20 J=1,12
      ILUM(J) = ILUMIN(J)
   20 CONTINUE
      ISWTCH = 2
      IF(NTSTEP .GE. 100) ISWTCH = 1
      RETURN
      END


      SUBROUTINE BNDSET(X,Y)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      REAL*8 KAPPAS, KAPPAX, KAPPAY, KAPPAZ, KAPPAE, KAPPAT,KBOLTZ
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,D1,C2,C3,C4,C5,OPCON,PICON,DUMALF,
     X              ECONST,EXP1,EXP2,FRAC,DUMX,DUMY,DUMZ,A0,H,VCOEF
      Z = 1.D0 - X - Y
      DM(2*N-1) = M(2*N-1) - M(2*N-2)
      OPCON1 = 4.D0*FPITHD/DM(2*N-1)
      KAPPAE = (FPI/3.D0)*A0**2*(1.D0 + X)/H
      A = 7.5641D-15
      C = 2.997929D10
      OPCON2 = PI*A*C
      KBOLTZ = 1.38046D-16
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      C1 = KBOLTZ*MUINV/H
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)
      ALPHAS = ALPHA*1.D4*4.85D-13/2.2D-5
      ALPHAX = ALPHA*1.D4*X/2.1D0
      ALPHAY = ALPHA*1.D4*Y
      ALPHAZ = ALPHA*1.D4*Z/35.D0
      ZCON = 3.8D2*Z*((2.D0 - Z)**2*(1 + X))**0.33D0/35.D0
      OC0 = 1.D0/2.2D-5
      OC1 = 2.D6/2.1D0
      OC2 = 1.D0/4.5D0
      OC3 = OC2*2.1D0
      OC4 = 20.D0/35.D0
      RETURN
C
      ENTRY BNDRY
C
   10 CONTINUE
      KAPPAT = KAPPAE
      OPCONR = OPCON1*R(N,2)**2/KAPPAE
```

```
      TE4TH = OPCONR*T(N-1)**4/(OPCONR + 1.D0)
      TE = TE4TH**0.25D0
      PE = (OPCONR*P(N-1) + C5*TE4TH)/(OPCONR + 2.D0)
      RHOE = (PE - C5*TE4TH)/(C1*TE)
      T4 = TE/1.D4
      RTT4 = DSQRT(T4)
      T4SQ = T4*T4
      T4CU = T4SQ*T4
      T4FIV = T4CU*T4SQ
      KAPPAS = ALPHAS/(OC0 + T4)
      KAPPAX = ALPHAX*(RTT4/(OC1/T4FIV + T4FIV) + OC3/(T4FIV + OC2
     X          /(4.D-3/T4SQ + 2.D-4*T4SQ/RHOE**0.25D0)))
      KAPPAY = ALPHAY*(1.D0/(1.4D3 + T4FIV) + 15.D0/(1.D7/T4 + T4FIV))
      KAPPAZ = ALPHAZ*RTT4/(OC4 + T4CU + ZCON*RHOE**0.33D0*T4**3.71D0)
      KAPPAE = KAPPAS + (KAPPAX + KAPPAY + KAPPAZ)*RHOE
      IF(DABS(KAPPAT - KAPPAE)/KAPPAE .GT. 1.D-12)   GO TO 10
      T(N) = TE
      P(N) = PE
      RHO(N) = RHOE
      KAPPA(N) = KAPPAE
      L(N) = OPCON2*R(N,2)**2*TE4TH
      RETURN
      END


      SUBROUTINE STEPSI(DZ)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H
      DZMIN = 1.D60
      DO 10 I=2,NM1
      DZI = DM(2*I)/(FPI*R(I,1)**2
     X      *DSQRT(GAMMA1(RHO(I),T(I))*P(I)*RHO(I)))
      OPCON3 = 4.D0*OPCON/(KAPPA(I)+KAPPA(I-1))
      DZI2 = (C3 + C4*T(I)**3/(2.D0*RHO(I)))*(DM(2*I)**2/
     X       (OPCON3*PI*R(I,1)**4*T(I)**3))
      DZMIN = DMIN1(DZI,DZI2,DZMIN)
   10 CONTINUE
      DZ = 0.8D0 * DZMIN
      RETURN
      END


      SUBROUTINE DZCALC(Z,DZ)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
```

```
     X              ECONST,EXP1,EXP2,FRAC,X,Y,XZ,A0,HATOM
    97 FORMAT(///'-MODEL AT TIME =',1PD23.16,' SECONDS')
    98 FORMAT('0STEPSIZE THERMALLY LIMITED'/'0ZONE NUMBER =',I4,
     X   '         MASS FRACTION =',F7.4,'         RADIUS=',1PD13.6,
     X   '         TIME STEP =',1PD23.16//)
    99 FORMAT('0STEPSIZE COURANT LIMITED'/'0ZONE NUMBER =',I4,
     X   '         MASS FRACTION =',F7.4,'         RADIUS=',1PD13.6,
     X   '         TIME STEP =',1PD23.16//)
        NDSRN = 90
        DZMIN = 1.D60
        DO 10 I=2,NM1
        DZI = DM(2*I)/(FPI*R(I,1)**2
     X       *DSQRT(GAMMA1(RHO(I),T(I))*P(I)*RHO(I)))
        OPCON3 = 4.D0*OPCON/(KAPPA(I)+KAPPA(I-1))
        DZI2 = (C3 + C4*T(I)**3/(2.D0*RHO(I)))*(DM(2*I)**2/
     X        (OPCON3*PI*R(I,1)**4*T(I)**3))
        DZMINT = DMIN1(DZI,DZI2)
        IF(DZMINT .GE. DZMIN) GO TO 10
        DZMIN = DZMINT
        IMIN = I
        MFRAC = M(2*IMIN-1)/M(2*N-1)
        IF(DZMIN .EQ. DZI) NDZ = 1
    10 CONTINUE
        WRITE(NDSRN,97) Z
        IF(NDZ .EQ. 1) GO TO 20
        WRITE(NDSRN,98) IMIN, MFRAC, R(IMIN,2), DZMIN
        GO TO 30
    20 CONTINUE
        WRITE(NDSRN,99) IMIN, MFRAC, R(IMIN,2), DZMIN
    30 CONTINUE
        DZ = 0.8D0 * DZMIN
        RETURN
        END


        FUNCTION GAMMA1(RHO,T)
C
        IMPLICIT REAL*8(A-H,L,M,O-Z,$)
        BETA = 1.D0 / ( 1.D0 + 1.8895D-23*T**3/RHO )
        GAMMA1 = BETA + (4.D0 - 3.D0*BETA)**2/(12.D0 - 10.5D0*BETA)
        RETURN
        END


        SUBROUTINE OUTPUT(NCON,NSURF,Z,DZ,NPSTEP,NPZONE,NPLOT,NZONE)
C
        IMPLICIT REAL*8(A-H,L,M,O-Z,$)
        REAL*8 KAPPA
        COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
        COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
        COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
        COMMON /CONST/PI,FPI,FPITHD,U,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X           ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,HATOM,VCOEF,NMODEL,NTSTEP
    10 FORMAT(F9.4,1P8D15.3)
    20 FORMAT(' TIME =',1PD10.3,' SECONDS',' - DT =',1PD9.2,' SECONDS',
     X       ' - STEP NO.',I5,' -',I5,' ZONES - MASS =',1PD10.3,
```

```
     X       ' GM  -  RADIUS =',1PD10.3,' CM'/
     X       T56,'BETA CEPHEID MODEL NUMBER',I3,'  -  VISCOSITY COEF =',
     X       1PD8.1,'  -  COMPLETE OPACITY')
  30 FORMAT(F9.4,1P2D15.3,1PD75.3)
 100 FORMAT(1H1,T33,'P U L S A T I N G   S T A R          1 5   S O L A
     X R   M A S S E S'/)
 200  FORMAT(' C O N V E C T I V E    Z O N E'/)
 300  FORMAT(' R A D I A T I V E    Z O N E'/)
 400 FORMAT(1H0,T5,'MASS',T17,'RADIUS',T31,'VELOCITY',T46,'PRESSURE',
     X       T62,'DENSITY',T75,'TEMPERATURE',T92,'ENTROPY',T106,
     X       'LUMINOSITY',T122,'ENERGY'/T3,'FRACTION',T18,'(CM)',T31,
     X       '(CM/SEC)',T45,'(GM/SQ.CM)',T62,'(GM/CC)',T76,'(DEG-ABS)',
     X       T89,'(ERGS/GM-DEG)',T106,'(ERGS/SEC)',T119,
     X       '(ERGS/GM-SEC)'/)
 405 FORMAT(1H0,T5,'MASS',T17,'RADIUS',T31,'VELOCITY',T46,'PRESSURE',
     X       T62,'DENSITY',T75,'TEMPERATURE',T92,'ENTROPY',T106,
     X       'LUMINOSITY',T122,'OPACITY'/T3,'FRACTION',T18,'(CM)',T31,
     X       '(CM/SEC)',T45,'(GM/SQ.CM)',T62,'(GM/CC)',T76,'(DEG-ABS)',
     X       T89,'(ERGS/GM-DEG)',T106,'(ERGS/SEC)'/)
 500 FORMAT(/)
 900 FORMAT(6A4)
     NTSTEP = NTSTEP + 1
     IPLOT = IPLOT - 1
     IF(IPLOT .GT. 0) GO TO 8
     WRITE(22,900) Z,R(NZONE,2),R(NSURF,2),L(NZONE),L(NSURF),V(NSURF,2)
     IPLOT = NPLOT
   8 CONTINUE
     NSTEP = NSTEP - 1
     IF(NSTEP .GT. 0) RETURN
     NSTEP = NPSTEP
     I = 1 - NPZONE
     NTEST = NCON - NPZONE
     ASSIGN 25 TO NLABEL
   1 WRITE(3,100)
     WRITE(3,20) Z, DZ, NTSTEP, NM1, M(2*N-1), R(NSURF,1), NMODEL,VCOEF
     WRITE(3,200)
     WRITE(3,400)
     DO 2 J=1,50
     I = I + NPZONE
     MFRAC = M(2*I-1)/M(2*N-1)
     WRITE(3,10) MFRAC,R(I,2),V(I,2),P(I),RHO(I),T(I),S(I,2),L(I),E(I)
     IF(I .GE. NTEST) GO TO NLABEL,(25,5,6)
   2 CONTINUE
     GO TO 1
  25 CONTINUE
     NTEST = NSURF - NPZONE
     ASSIGN 5 TO NLABEL
   3 WRITE(3,100)
     WRITE(3,20) Z, DZ, NTSTEP, NM1, M(2*N-1), R(NSURF,1), NMODEL,VCOEF
     WRITE(3,300)
     WRITE(3,405)
     DO 4 J=1,50
     I = I + NPZONE
     MFRAC = M(2*I-1)/M(2*N-1)
     WRITE(3,10) MFRAC,R(I,2),V(I,2),P(I),RHO(I),T(I),S(I,2),L(I),
    X            KAPPA(I)
```

```
      IF(I .GE. NTEST) GO TO NLABEL,(25,5,6)
    4 CONTINUE
      GO TO 3
    5 CONTINUE
      IF(NSURF .EQ. N) GO TO 6
      I = NSURF
      MFRAC = M(2*I-1)/M(2*N-1)
      WRITE(3,10) MFRAC,R(I,2),V(I,2),P(I),RHO(I),T(I),S(I,2),L(I),
     X            KAPPA(I)
      WRITE(3,500)
      NTEST = N - NPZONE
      ASSIGN 6 TO NLABEL
      IF(I .GE. NTEST) GO TO NLABEL,(25,5,6)
      GO TO 4
    6 CONTINUE
      MFRAC = 1.D0
      WRITE(3,30) MFRAC,R(N,2),V(N,2),L(N)
      RETURN
      END


      SUBROUTINE PLTOUT(NSURF,Z,DZ,NDEVT)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      DIMENSION MASS(605)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,U,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X          ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,HATOM,VCOEF,NMODEL,NTSTEP
      DO 10 I=1,N
      MASS(I) = M(2*I-1)
   10 CONTINUE
      NTRY = 1
      GO TO 15
      ENTRY ZNEPLT(NSURF,Z,DZ,NDEVT)
      DO 11 I=1,N
      MASS(I) = I
   11 CONTINUE
      NTRY = 2
   15 CONTINUE
      WRITE(50,20) Z, DZ, NTSTEP, NM1, M(2*N-1), R(NSURF,1), X, Y, VCOEF
   20 FORMAT(' TIME =',1PD10.3,' SECONDS',' -  DT =',1PD9.2,' SECONDS',
     X       ' -  STEP NO.',I5,'  -',I5,' ZONES  -  MASS =',1PD10.3,
     X       ' GM' / ' RADIUS =',1PD10.3,' CM  -  X =',
     X                0PF5.2,'  -  Y =',F5.2,'  -  VISCOSITY COEF =',
     X       1PD8.1,'  -  COMPLETE SCATTERING')
      CALL GWRITE(50,63)
      CALL GTITLE('R   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('R VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,R,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('V   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('V VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,V,N,NDEVT)
```

```
      CALL GWRITE(50,63)
      CALL GTITLE('P   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('P VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,P,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('R H O   V S   M',15,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('RHO VS ZONE  ',13,NDEVT)
      CALL DGPLOT(MASS,RHO,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('T   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('T VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,T,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('S   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('S VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,S,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('L   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('L VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,L,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('E   V S   M',11,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('E VS ZONE  ',11,NDEVT)
      CALL DGPLOT(MASS,E,N,NDEVT)
      CALL GWRITE(50,63)
      CALL GTITLE('K A P P A   V S   M',19,NDEVT)
      IF(NTRY .EQ. 2)  CALL GTITLE('KAPPA VS ZONE  ',15,NDEVT)
      CALL DGPLOT(MASS,KAPPA,N,NDEVT)
      RETURN
      END


      SUBROUTINE STORE(Z)
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X          ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,HATOM,VCOEF,NMODEL,NTSTEP
  300 FORMAT(7A8)
      REWIND 8
      WRITE(8,300) Z,VCOEF,NTSTEP,NMODEL
      WRITE(8,300) (M(2*I-1), M(2*I), R(I,1), V(I,1), S(I,1), RHO(I),
     X             T(I), I=1,N)
      RETURN
      END


      SUBROUTINE PULSEN(Z,NSURF,DEVT)
C
C
C     THIS SUBROUTINE CALCULATES THE TOTAL KINETIC ENERGY OF THE STELLAR
C     MODELS CALCULATED BY THE PULS III CODE.  IT THEN STORES THIS RESULT,
C     PRECEDED BY THE MODEL'S TIME  PHOTOSPHERIC RADIUS, VELOCITY, AND
```

```
C     LUMINOSITY ON DSRN 11. IT UTILIZES COMMON CARDS OF THE PULS III
C     CODE.  IF DEVT .EQ. 'T', IT IS ASSUMED THAT DSRN 11 REFERNCES A
C     TAPE DATA SET AND TAPSET IS CALLED.  OTHERWISE, IT IS NOT.
C
C   NOTE - Z IS THE MODEL TIME FROM THE PULS III CODE.
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      LOGICAL*1 DEVT, TAPE / 'T' /
      REAL*8 EGCON(605)
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X              ECONST,EXP1,EXP2,FRAC,X,Y,W,A0,H
      IF(ISWTCH .NE. 0) GO TO 5
C     SET POSSIBLE DATA TAPE TO PROPER POSITION.
      IF(DEVT .EQ. TAPE) CALL TAPSET(11)
      DO 1 I=1,NM1
      EGCON(I) = G*DM(2*I)*(M(I) + M(I+1))
    1 CONTINUE
      ISWTCH = 1
    5 CONTINUE
C*********************************************************************
C     NUCLEAR (EN) AND GRAVITATIONAL (EG) ENERGIES ADDED 9/7/69
C*********************************************************************
C     INITIALIZE KINETIC ENERGY PRIOR TO INTEGRATION.
      EK = 0.D0
      EN = 0.D0
      EG = 0.D0
C     INTEGRATE KINETIC ENERGY THROUGHOUT STAR.
      DO 10 I=1,NM1
      EK = EK + (V(I,2)**2 + V(I+1,2)**2)*DM(2*I)/2.D0
      EN = EN + E(I)*DM(2*I)
      EG = EG - EGCON(I)/(R(I,2) + R(I+1,2))
   10 CONTINUE
C     STORE DATA ON DSRN 11 (IDEAL LRECL=56).
      WRITE(11,100) Z, R(NSURF,2), V(NSURF,2), L(NSURF), EK, EN, EG
  100 FORMAT(7A8)
      RETURN
      END


      SUBROUTINE TAPSET(NDSRN)
C
C     REVISED FROM 80 TO 56 BYTE READ + REWIND ON 9/16/69.
C     WHEN FIRST REFERENCED, THIS SUBROUTINE POSITIONS A DATA TAPE SUCH
C     THAT DATA MAY BE ADDED TO THE END OF DATA SET.  IT REMOVES THE
C     ORIGINAL END OF FILE.  SUBSEQUENT CALLS TO THIS ROUTINE RESULT IN
C     IMMEDIATE RETURNS TO THE CALLING PROGRAM.
C
C
C   NOTE - NDSRN IS THE LOGICAL ADDRESS OF THE DATA TAPE.  IT IS ASSUMED
C          (BUT NOT ESSENTIAL) THAT THE DATA RECORDS ARE 56 BYTES LONG.
C
      REAL*8 A(7)
```

```
C     RETURN PARAMETER INITIALIZED.
      DATA ISWTCH / 0 /
C     IF FIRST PASS PROCEED, OTHERWISE RETURN.
      IF(ISWTCH .NE. 0) RETURN
      REWIND NDSRN
C     TAPE IS READ TO END OF ORIGINAL DATA SET IN 56 BYTE RECORDS.
    5 READ(NDSRN,100,END=10) A
  100 FORMAT(7A8)
      GO TO 5
   10 CONTINUE
C     BACKSPACE OVER ORIGINAL END OF FILE .
      BACKSPACE NDSRN
C     TAPE IS NOW READY TO HAVE DATA ADDED TO ORIGINAL DATA SET.
C
C     RETURN PARAMETER RESET FOR IMMEDIATE RETURNS ON FUTURE CALLS.
      ISWTCH = 1
      RETURN
      END


      SUBROUTINE CALPRT
C
      DIMENSION COPY(33)
   98 FORMAT(1H1)
   99 FORMAT(33A4)
      NDSRN = 90
      ENDFILE NDSRN
      REWIND NDSRN
      WRITE(3,98)
    5 CONTINUE
      READ(NDSRN,99,END=10) COPY
      WRITE(3,99) COPY
      GO TO 5
   10 CONTINUE
      WRITE(3,98)
      RETURN
      END


/*
/DD  5    DSNAME=GPLT,DISP=PASS
/DD  7    DSNAME=LIBP,DISP=KEEP
/EXEC     PUNCH
/*
```

F.8 SUBROUTINE BNDRY

When included in CEPHEI LIBRARY (F.7), this subroutine sets up the Eddington boundary conditions.

```
      SUBROUTINE BNDRY
C
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA
      COMMON /DYNVAR/ R(605,2), V(605,2), RHO(605)
      COMMON /STAVAR/ P(605), T(605), S(605,2), E(605)
      COMMON /MASLUM/ M(1210), DM(1210), L(605), KAPPA(605), N, NM1
      COMMON /CONST/PI,FPI,FPITHD,G,C1,C2,C3,C4,C5,OPCON,PICON,ALPHA,
     X               ECONST,EXP1,EXP2,FRAC,X,Y,Z,A0,H
      K = 2*N - 1
      OPCON1 = PICON*(KAPPA(N)+KAPPA(N-1))
      P(N) = P(N-1)*(R(N,2)**2-OPCON1*DM(K))/(R(N,2)**2+OPCON1*DM(K))
      TN4TH =T(N-1)**4*(R(N,2)**2-OPCON1*DM(K))/(R(N,2)**2+OPCON1*DM(K))
      OPCON2 = OPCON/(KAPPA(N)+KAPPA(N-1))
      L(N) = OPCON2*R(N,2)**4*(T(N-1)**4 - TN4TH)/DM(2*N-1)
      RETURN
      END
/*
```

F.9 EQUILIB ZONE III

This set of routines serves as an example of an equilibrium zoning scheme, for massive star models, which utilizes an exponential mass zoning scheme and computes a stellar atmosphere of a specified number of zones.

```
/ID   1444   T UNGAR,S. EQUILIB ZONE III
C     SUBROUTINES MFIT3 AND MFIT4 UPDATED 9/11/69.
/DD  99  DSNAME=TEMP,DISP=SCR
/EXEC    EXIT
/DD  4   DSNAME=CALC,DISP=PASS
/DD  9   DEVT=CORE
/DD  5   DISP=PASS,UPD(2,0,22)
/DD  22  DISP=PASS,SYSPCH
/EXEC    FORTX,'OPT=2'              VERSION 2,     8/13/69
.NUM                                                                    ZNF00000
.NOPRT
C     THIS PROGRAM MODIFIED ON 8/30/69 TO ALLOW FOR 'A' FORMAT DATA
C     TRANSFER TO THE PULS III CODE.
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      REAL*8 KAPPA, KBOLTZ, KAPPAT
      DIMENSION FPCON(5), FTCON(5), FLCON(5), FRCON(5)
      DIMENSION FPRAD(5), FTRAD(5), FLRAD(5), FRRAD(5)
      DATA AD, RAD / '  AD    ', '  RAD   ' /
  100 FORMAT(1H1,T33,'P U L S A T I N G   S T A R       ',I5,'   S O L A
     X R   M A S S E S       (EQUILIBRIUM MODEL)'/)
  200 FORMAT('/*')
  300 FORMAT(8A8)
  400 FORMAT(1H0,T5,'MASS',T17,'RADIUS',T33,'HEAT',T46,'PRESSURE',
     X       T62,'DENSITY',T75,'TEMPERATURE',T92,'ENTROPY',T106,
     X       'LUMINOSITY',T122,'ENERGY'/T3,'FRACTION',T16,'(CM)',T31,
     X       'TRANSFER',T45,'(GM/SQ.CM)',T62,'(GM/CC)',T76,'(DEG-ABS)',
     X       T89,'(ERGS/GM-DEG)',T103,'(ERGS/SQ.CM-SEC)',T121,
     X       '(ERGS/GM)'/)
  500 FORMAT(F9.4,1PD15.3,7X,A8,1P6D15.3)
C     CHOOSE NUMBER OF OUTPUT ZONES FOR EQUIVALENT EQUALLE-ZONED MODEL.
      NZONE = 300
C     READ MODEL'S MASS AND CHEMICAL COMPOSITION.
C                  ********** AND **********
C     READ STARTING VALUES FOR CENTRAL PRESSURE AND TEMPERATURE AS WELL
C     AS STARTING VALUES FOR SURFACE LUMINOSITY AND RADIUS.
      READ(4,98) NMSUN, X, Y, PC, TC, L0, R0
   98 FORMAT(I5,1P6D25.16)
      WRITE(9,198) NMSUN, X, Y, PC, TC, L0, R0
  198 FORMAT(A4,6A8)
      WRITE(3,99) NMSUN, X, Y, PC, TC, L0, R0
   99 FORMAT(///I9,' SOLAR MASSES'/'0      X ='F7.4/'       Y =',F7.4/
     X          '0     PC =',1PD23.16/'      TC =',1PD23.16/
     X          '      L0 =',1PD23.16/'      R0 =',1PD23.16////)
C     SET MAXIMUM MASS ZONE SIZE.
      N = 1000
      PI = 3.1415926535897932D0
      FPI = 4.D0*PI
      FPITHD = FPI/3.D0
      A = 7.5641D-15
      A0 = 2.81785D-13
      C = 2.997929D10
      G = 6.673D-8
      H = 1.67333D-24
      KBOLTZ = 1.380646D-16
      RSUN = 6.9598D10
      MSUN = 1.989D33
```

```
      LSUN = 3.90D33
      KAPPAT = (FPI/3.D0)*A0**2*(1.D0 + X)/H
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      SIGMA = A*C/4.D0
      C1 = KBOLTZ*MUINV/H
      C2 = 4.D0*A/3.D0
      C3 = 3.D0*C1
      C4 = 4.D0*A
      C5 = A/3.D0
      OPCON4 = 64.D0*A*C*PI**2/3.D0
      MSTAR = NMSUN*MSUN
C     CHOOSE FITTING POINT MASS.
      MFF = 0.89D0*MSTAR
      DMM = MSTAR/N
      XCNO = (1.D0 - X - Y)/2.D0
      ECONST = 7.9D31*X*XCNO
      EXP1 = 1.D0/3.D0
      EXP2 = 2.D0/3.D0
      FRAC = 8.D0/3.D0
      GCONST = G/FPI
C     SET MAXIMUM PRESSURE ZONE SIZE.
      GCON2 = 1.D-1/GCONST
      NFF = NZONE*MFF/MSTAR + 1.D-6
C     SET OUTPUT MASS ZONING PARAMETER.
      NEXP = 4
C     FIND THE ADJUSTED FITTING POINT MASS AND ZONE NUMBER FOR OUTPUT
C     MASS ZONE DISTRIBUTION CORRESPONDING TO NEXP AND MFF.
      CALL MFIT1(NEXP,MSTAR,MFF,NFF,L0,R0)
C     CHOOSE CONSTANT ZONE SIZE FOR CORE REGION
      DMZONE = MFF/NFF
C     ********************************************************************
C     START INTEGRATION FROM CENTER OUT                *****CORE*****
C     ********************************************************************
C     CALCULATE CENTRAL VALUES.
      V = A0
      M = 0.D0
      MFRAC = 0.D0
      R = 0.D0
      L = 0.D0
      T = TC
      P = PC
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      F = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,100) NMSUN
      WRITE(9,300) M, R, RHO, P, T, S, L, F
      WRITE(3,400)
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
C     CALCULATE VALUES ONE ZONE OUT BY APPROXIMATE EXPANSION.
      BETA = C1*RHO*T/P
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)
      M = DMM
      R = ((3.D0/FPI)*DMM/RHO)**EXP1
      T = T - (FPI/6.D0)*G*R**2*RHO**2*(1.D0 - 1.D0/GAMMA2)*T/P
      P = P - (FPI/6.D0)*G*R**2*RHO**2
      L = DMM*F
```

```
      MFRAC = M/MSTAR
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      F = FCONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
C     BEGIN INTEGRATION
      MF = DMZONE
    6 CONTINUE
      MM = M
      PM = P
      TM = T
      LM = L
      RM = R
      DMP = GCON2*R**4*P/M
      DM = DMIN1(DMM,DMP)
    8 DO 10 K=2,5
      A = ((K - 1)/2)/2.D0
      MT = M + DM*A
      PTCON = P + FPCON(K-1)*A
      PTRAD = P + FPRAD(K-1)*A
      TTCON = T + FTCON(K-1)*A
      TT4RAD = T**4 + FTRAD(K-1)*A
      TTRAD = TT4RAD**0.25D0
      LTCON = L + FLCON(K-1)*A
      LTRAD = L + FLRAD(K-1)*A
      RT3CON = R**3 + FRCON(K-1)*A
      RTCON = RT3CON**EXP1
      RT3RAD = R**3 + FRRAD(K-1)*A
      RTRAD = RT3RAD**EXP1
      RHOCON = (PTCON - C5*TTCON**4)/(C1*TTCON)
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)
      BETA= C1*RHOCON*TTCON/PTCON
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)
      FPCON(K) = -GCONST*MT*DM/RTCON**4
      FPRAD(K) = -GCONST*MT*DM/RTRAD**4
      FTCON(K) = (1.D0 - 1.D0/GAMMA2)*TTCON*FPCON(K)/PTCON
      OPCON3 = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)
      FLCON(K)=ECONST*RHOCON*DEXP(-1.5231D4/TTCON**EXP1)*DM/TTCON**EXP2
      FLRAD(K)=ECONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2
      FRCON(K) = DM/(FPITHD*RHOCON)
      FRRAD(K) = DM/(FPITHD*RHORAD)
   10 CONTINUE
      M = M + DM
      MFRAC = M/MSTAR
      PCON = P + (FPCON(2)+2.D0*FPCON(3)+2.D0*FPCON(4)+FPCON(5))/6.D0
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0
      TCON = T + (FTCON(2)+2.D0*FTCON(3)+2.D0*FTCON(4)+FTCON(5))/6.D0
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0
      TRAD = T4RAD**0.25D0
      LCON = L + (FLCON(2)+2.D0*FLCON(3)+2.D0*FLCON(4)+FLCON(5))/6.D0
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0
      R3CON = R**3+(FRCON(2)+2.D0*FRCON(3)+2.D0*FRCON(4)+FRCON(5))/6.D0
      RCON = R3CON**EXP1
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0
      RRAD = R3RAD**EXP1
```

```
      IF(TCON .GT. TRAD) GO TO 12
      P = PRAD
      T = TRAD
      L = LRAD
      R = RRAD
      V = RAD
      GO TO 14
   12 CONTINUE
      P = PCON
      T = TCON
      L = LCON
      R = RCON
      V = AD
   14 CONTINUE
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**4/RHO
      F = FCONST*RHO*DEXP(-1.5241D4/T**EXP1)/T**EXP2
      IF(M - MF) 6,20,16
   16 CONTINUE
      M = MM
      P = PM
      T = TM
      L = LM
      R = RM
      DM = MF - M
      GO TO 8
   20 CONTINUE
      WRITE(9,300) M, R, RHO, P, T, S, L, F
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
      MF = MF + DMZONE
      IF(MF .LE. MFF) GO TO 6
      WRITE(9,200)
      PFO = P
      TFO = T
      LFO = L
      RFO = R
C     ****************************************************************
C     START INTEGRATION FROM SURFACE IN              *****FX,F,CFF*****
C     ****************************************************************
C *   CALCULATE SURFACE VALUES
      V = RAD
      T0 = (L0/(FFI*SIGMA*R0**2))**0.25D0
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPAT*R0**2) + C5*T0**4/2.D0
      RHOT = (P0 - C5*T0**4)/(C1*T0)
C     ITERATE TO FIND SURFACE PRESSURE AND DENSITY
 1045 CONTINUE
      RHO = RHOT
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPA(X,Y,RHO,T0)*R0**2) + C5*T0**4/2.D0
      RHOT = (P0 - C5*T0**4)/(C1*T0)
      IF(DABS((RHO-RHOT)/RHOT) .GT. 1.D-12) GO TO 1045
      M = MSTAR
      MFRAC = 1.D0
      L = L0
      R = R0
      P = P0
      T = T0
```

```
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,100) NMSUN
      WRITE(3,400)
      WRITE(9,300) M, R, RHO, P, T, S, L, E
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
C     FIND OUTPUT MASS ZONE SIZES IN ENVELOPE REGION
      CALL MFIT2(MF)
C     BEGIN INTEGRATION.
  106 CONTINUE
      MM = M
      PM = P
      TM = T
      LM = L
      RM = R
      DMP = GCON2*R**4*P/M
      DM = -DMIN1(DMM,DMP)
  108 DO 110 K=2,5
      A = ((K - 1)/2)/2.D0
      MT = M + DM*A
      PTCON = P + FPCON(K-1)*A
      PTRAD = P + FPRAD(K-1)*A
      TTCON = T + FTCON(K-1)*A
      TT4RAD = T**4 + FTRAD(K-1)*A
      TTRAD = TT4RAD**0.25D0
      LTCON = L + FLCON(K-1)*A
      LTRAD = L + FLRAD(K-1)*A
      RT3CON = R**3 + FRCON(K-1)*A
      RTCON = RT3CON**EXP1
      RT3RAD = R**3 + FRRAD(K-1)*A
      RTRAD = RT3RAD**EXP1
      RHOCON = (PTCON - C5*TTCON**4)/(C1*TTCON)
      RHORAD = (PTRAD - C5*TTRAD**4)/(C1*TTRAD)
      BETA= C1*RHOCON*TTCON/PTCON
      GAMMA2 = 1.D0 + (FRAC - 2.D0*BETA)/(8.D0 - 6.D0*BETA - BETA**2)
      FPCON(K) = -GCONST*MT*DM/RTCON**4
      FPRAD(K) = -GCONST*MT*DM/RTRAD**4
      FTCON(K) = (1.D0 - 1.D0/GAMMA2)*TTCON*FPCON(K)/PTCON
      OPCON3 = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON3*RTRAD**4)
      FLCON(K)=FCONST*RHOCON*DEXP(-1.5231D4/TTCON**EXP1)*DM/TTCON**EXP2
      FLRAD(K)=FCONST*RHORAD*DEXP(-1.5231D4/TTRAD**EXP1)*DM/TTRAD**EXP2
      FRCON(K) = DM/(FPITHD*RHOCON)
      FRRAD(K) = DM/(FPITHD*RHORAD)
  110 CONTINUE
      M = M + DM
      MFRAC = M/MSTAR
      PCON = P + (FPCON(2)+2.D0*FPCON(3)+2.D0*FPCON(4)+FPCON(5))/6.D0
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0
      TCON = T + (FTCON(2)+2.D0*FTCON(3)+2.D0*FTCON(4)+FTCON(5))/6.D0
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0
      TRAD = T4RAD**0.25D0
      LCON = L + (FLCON(2)+2.D0*FLCON(3)+2.D0*FLCON(4)+FLCON(5))/6.D0
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0
      R3CON = R**3+(FRCON(2)+2.D0*FRCON(3)+2.D0*FRCON(4)+FRCON(5))/6.D0
```

```
      RCON = R3CON**EXP1
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0
      RRAD = R3RAD**EXP1
      IF(TCON .LT. TRAD) GO TO 112
      P = PRAD
      T = TRAD
      L = LRAD
      R = RRAD
      V = RAD
      GO TO 114
  112 CONTINUE
      P = PCON
      T = TCON
      L = LCON
      R = RCON
      V = AD
  114 CONTINUE
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      IF(M - MF) 116,60,106
  116 CONTINUE
      M = MM
      P = PM
      T = TM
      L = LM
      R = RM
      DM = MF - M
      GO TO 108
   60 CONTINUE
      WRITE(9,300) M, R, RHO, P, T, S, L, E
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
      CALL MFIT2(MF)
      IF(MF .GE. MFF) GO TO 106
      WRITE(9,200)
      PFI = P
      TFI = T
      LFI = L
      RFI = R
C     ***********************************************************************
C     START INTEGRATION FROM SURFACE OUT.              *****ATMOSPHERE*****
C     ***********************************************************************
C     CALCULATE SURFACE VALUES.
      V = RAD
      M = MSTAR
      MFRAC = 1.D0
      L = L0
      R = R0
      P = P0
      T = T0
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      WRITE(3,100) NMSUN
      WRITE(3,400)
      WRITE(9,300) M, R, RHO, P, T, S, L, E
```

```
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,F
C     CHOOSE NUMBER OF ATMOSPHERIC ZONES
      NATMO = 8
C     SET ATMOSPHERIC CUTOFF MASS CORRESPONDING TO NATMO.
      CALL MFIT3(NATMO,MF,MFF)
C     BEGIN INTEGRATION.
  200 CONTINUE
      MM = M
      PM = P
      TM = T
      LM = L
      RM = R
      DMP = .3CON2*R**4*P/M
      DM = DMIN1(DMM,DMP)
  205 DO 210 K=2,5
      A = ((K - 1)/2)/2.D0
      MT = M + DM*A
      PTCON = P + FPCON(K-1)*A
      PTRAD = P + FPRAD(K-1)*A
      TTCON = T + FTCON(K-1)*A
      TT4RAD = T**4 + FTRAD(K-1)*A
      TTRAD = TT4RAD**0.2500
      LTCON = L + FLCON(K-1)*A
      LTRAD = L + FLRAD(K-1)*A
      RT4CON = R**4 + FRCON(K-1)*A
      RTCON = RT4CON**EXP1
      RT3RAD = R**3 + FRRAD(K-1)*A
      RTRAD = RT3RAD**EXP1
      RHOCON = (PTCON - .5*TTCON**4)/(.1*TTCON)
      RHORAD = (PTRAD - .5*TTRAD**4)/(.1*TTRAD)
      BETA= .1*RHOCON*TTCON/PTCON
      GAMMA2 = 1.D0 + (BRAC - 2.D0*BETA)/(8.D0 + 1.6*BETA + BETA**2)
      FPCON(K) = -.3CONST*MT*DM/RTCON**4
      FPRAD(K) = -.3CONST*MT*DM/RTRAD**4
      FTCON(K) = (1.D0 - 1.5/GAMMA2)*TTCON*FPCON(K)/PTCON
      OPCON = OPCON4/KAPPA(X,Y,RHORAD,TTRAD)
      FTRAD(K) = -4.D0*LTRAD*DM/(OPCON*RTRAD**4
      FLCON(K)=FCONST*RHOCON*OPEXP1*2.4/TTCON**EXP1)*DM/TTCON**EXP2
      FLRAD(K)=FCONST*RHORAD*OPEXP1*2.4/TTRAD**EXP1)*DM/TTRAD**EXP2
      FRCON(K) = DM/(FPITHO*RHOCON)
      FRRAD(K) = DM/(FPITHO*RHORAD)
  210 CONTINUE
      M = M + DM
      MFRAC = M/MSTAR
      PCON = P + (FPCON(2)+2.D0*FPCON(3)+2.D0*FPCON(4)+FPCON(5))/6.D0
      PRAD = P + (FPRAD(2)+2.D0*FPRAD(3)+2.D0*FPRAD(4)+FPRAD(5))/6.D0
      TCON = T + (FTCON(2)+2.D0*FTCON(3)+2.D0*FTCON(4)+FTCON(5))/6.D0
      T4RAD = T**4+(FTRAD(2)+2.D0*FTRAD(3)+2.D0*FTRAD(4)+FTRAD(5))/6.D0
      TRAD = T4RAD**0.2500
      LCON = L + (FLCON(2)+2.D0*FLCON(3)+2.D0*FLCON(4)+FLCON(5))/6.D0
      LRAD = L + (FLRAD(2)+2.D0*FLRAD(3)+2.D0*FLRAD(4)+FLRAD(5))/6.D0
      R3CON = R**3+(FRCON(2)+2.D0*FRCON(3)+2.D0*FRCON(4)+FRCON(5))/6.D0
      RCON = R3CON**EXP1
      R3RAD = R**3+(FRRAD(2)+2.D0*FRRAD(3)+2.D0*FRRAD(4)+FRRAD(5))/6.D0
      RRAD = R3RAD**EXP1
      IF(TCON .GT. TRAD) GO TO 212
```

```
      P = PRAD
      T = TRAD
      L = LRAD
      R = RRAD
      V = RAD
      GO TO 214
  212 CONTINUE
      P = PCON
      T = TCON
      L = LCON
      R = RCON
      V = AD
  214 CONTINUE
      RHO = (P - C5*T**4)/(C1*T)
      S = C1*DLOG(T**1.5D0/RHO) + C2*T**3/RHO
      E = ECONST*RHO*DEXP(-1.5231D4/T**EXP1)/T**EXP2
      IF(M - MF) 206,80,216
  216 CONTINUE
      M = MM
      P = PM
      T = TM
      L = LM
      R = RM
      DM = MF - M
      GO TO 208
   80 CONTINUE
      WRITE(9,300) M, R, RHO, P, T, S, L, E
      WRITE(3,500) MFRAC,R,V,P,RHO,T,S,L,E
C     FIND OUTPUT MASS ZONE SIZES IN ATMOSPHERE
      CALL MFIT4(MF)
      IF(MF .LE. MFF) GO TO 206
      WRITE(9,200)
      DELP = PFO - PFI
      DELT = TFO - TFI
      DELL = LFO - LFI
      DELR = RFO - RFI
      WRITE(3,97) DELP, DELT, DELL, DELR
   97 FORMAT(1H1//'-      GOODNESS OF FIT AT FITTING POINT'///
     X           '       DELP =',1PD24.16//'       DELT =',1PD24.16/
     X           '       DELL =',1PD24.16//'       DELR =',1PD24.16/////)
      REWIND 9
      STOP
      END



      SUBROUTINE MFIT1(NEXP,MSTAR,MFF,NFF,L0,R0)
      IMPLICIT REAL*8(A-H,K-M,O-Z,$)
      COMMON MSTART,MDEL,MFFAC,XNFF,XNF,NDUM
       COMMON NSTAR
      NDUM = NEXP
      X = 0.7D0
      Y = 0.27D0
      PI = 3.14159265358979320
      FPI = 4.D0*PI
      FPITHD = FPI/3.D0
```

```
      A = 7.5641D-15
      A0 = 2.81785D-13
      C = 2.997929D10
      G = 6.673D-8
      H = 1.67333D-24
      KBOLTZ = 1.38046D-16
      KAPPAT = (FPI/3.D0)*A0**2*(1.D0 + X)/H
      MUINV = 0.5D0 + 1.5D0*X + 0.25D0*Y
      SIGMA = A*C/4.D0
      C1 = KBOLTZ*MUINV/H
      C5 = A/3.D0
      T0 = (L0/(FPI*SIGMA*R0**2))**0.25D0
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPAT*R0**2) + C5*T0**4/2.D0
      RHOT = (P0 - C5*T0**4)/(C1*T0)
 1045 CONTINUE
      RHO = RHOT
      P0 = 2.D0*G*MSTAR/(3.D0*KAPPA(X,Y,RHO,T0)*R0**2) + C5*T0**4/2.D0
      RHOT = (P0 - C5*T0**4)/(C1*T0)
      IF(DABS((RHO-RHOT)/RHOT) .GT. 1.D-12) GO TO 1045
      KAPPA0 = KAPPA(X,Y,RHOT,T0)
      MSTART = MSTAR + 2.D0*FPITHD*R0**2/KAPPA0
      CALL CURFIT(NEXP,MSTART,MSTAR,MFF,NFF,NSTAR)
      MDEL = MSTART - MFF
      XNFF = NFF
      XNF = NSTAR
      MFFAC = MFF/(NEXP*MDEL)
      RETURN
      END


      SUBROUTINE MFIT2(MF)
      IMPLICIT REAL*8(A-H,K-M,O-Z,$)
      COMMON MSTART,MDEL,MFFAC,XNFF,XNF,NEXP
      XNF = XNF - 1.D0
      MF = MSTART - MDEL*DEXP(MFFAC*(1.D0 - (XNF/XNFF)**NEXP))
      RETURN
      END


      SUBROUTINE MFIT3(NATMO,MF,MFF)
C     UPDATED 9/11/69 TO ACCOMODATE RUNS WITH NATMO=0
      IMPLICIT REAL*8(A-H,K-M,O-Z,$)
      COMMON MSTART,MDEL,MFFAC,XNFF,XNF,NEXP
       COMMON NSTAR
      XNF = NSTAR + NATMO
      MF = MSTART - MDEL*DEXP(MFFAC*(1.D0 - (XNF/XNFF)**NEXP))
      MFF = MF
      XNF = NSTAR + 1
      MF = MSTART - MDEL*DEXP(MFFAC*(1.D0 - (XNF/XNFF)**NEXP))
      IF(NATMO .NE. 0) RETURN
      MF = MSTAR
      MFF = MF
      RETURN
      END
```

```
      SUBROUTINE MFIT4(MF)
C     UPDATED 9/11/69 TO ACCOMODATE RUNS WITH NATMO=0.
      IMPLICIT REAL*8(A-H,K-M,O-Z,$)
      COMMON MSTART,MDEL,MFFAC,XNFF,XNF,NFXP
      XNF = XNF + 1.D0
      MF = MSTART - MDEL*DEXP(MFFAC*(1.D0 - (XNF/XNFF)**NFXP))
      IF(NATMO .NE. 0) RETURN
      MF = MSTAR
      MFF = MF
      WRITE(3,10)
   10 FORMAT(/' N A T M O   =   0          ALL VALUES ON THIS PAGE REFER
     XTO PHOTOSPHERE'/)
      RETURN
      END


      SUBROUTINE CURFIT(N,Y,YS,Y0,NX0,NXS)
      IMPLICIT REAL*8(A-H,L,M,O-Z,$)
      G(Y0) = XN*(Y - Y0)*DLOG((Y - Y0)/(Y - YS))/Y0
      F(Y0) = 1.D0 + G(Y0) - (XS/X0)**N
      DF(Y0) = -(XN + G(Y0)*Y/(Y - Y0))/Y0
      XN = N
      X0 = NX0
      Y01 = Y0
      XS = X0*(1.D0 + G(Y0))**(1.D0/XN)
      NXS = IDINT(XS)
      XS = NXS
C     NEWTON-RAPHSON ITERATION
   10 Y0T = Y0
      Y0 = Y0T - F(Y0T)/DF(Y0T)
      IF(DABS(Y0 - Y0T) .GT. 1.D-14*Y0T) GO TO 10
      RETURN
      END


      FUNCTION KAPPA(X,Y,RHO,T)
      IMPLICIT REAL*8(A-H,K,O-Z,$)
      DATA ISWTCH / 1 /
      IF(ISWTCH .NE. 1) GO TO 2
      H = 1.67333D-24
      KBOLTZ = 1.38046D-16
      Z = 1.D0 - X - Y
      ALPHA = KBOLTZ*(1.D0 + X)/(2.D0*H)
      ISWTCH = 2
    2 CONTINUE
      PE = ALPHA*RHO*T
      T4 = T / 1.D4
      KAPPAS = 4.85D-13*PE/((1.D0+2.2D-5*T4)*RHO*T4)
      KAPPAX = PE*X*(T4**0.5D0/(2.D6/T4**4+2.1D0*T4**6)+1.D0
     1          /(4.5D0*T4**6+1.D0/(T4*(4.D-3/T4**4+2.D-4/RHO**0.25D0))))
      KAPPAY = PE*Y*(1.D0/(1.4D3*T4+T4**6)+1.5D0/(1.D6+0.1D0*T4**6))
      KAPPAZ = PE*Z*T4**0.5D0/(2.D1*T4+3.5D1*T4**4+3.8D2*Z
     1          *(RHO*(2.D0-Z)**2*(1+X))**0.33D0*T4**4.71D0)
      KAPPA = KAPPAS + KAPPAX + KAPPAY + KAPPAZ
      RETURN
      END
```

```
/*
/EXEC     LDR
/DD  9    DSNAME=TEMP,DEVT=(2314,FILPRO),RECFM=F,LRECL=64,DISP=KEEP
/EXEC     *


/DD  99   DSNAME=M200,DISP=SCR
/EXEC     FORTX          DATA SET REARRANGEMENT          VERSION 2,  8/15/69
C      THIS PROGRAM REARRANGES THE OUTPUT FROM EQUILIBRIUM ZONE III TO
C      BE COMPATABLE WITH THE INPUT TO THE CURRENT VERSION OF THE STAR1
C      SUBROUTINE IN THE PULSE III CODE.
C      THIS PROGRAM MODIFIED ON 8/30/69 TO ALLOW FOR 'A' FORMAT DATA
C      TRANSFER TO THE PULS III CODE.
       REAL*8 X, Y, PC, TC, L0, R0
       REAL*8 A(1000), B(1000), C(1000), D(1000)
       REAL*8 E(1000), F(1000), G(1000), H(1000)
   10  FORMAT(8A8)
   20  FORMAT('/*')
   30  FORMAT(////' TOTAL NUMBER OF POINTS CONSIDERED =',I4//
     X        ' FITTING POINT IS NUMBER',I4//' SURFACE POINT IS NUMBER',
     X        I4,//' NUMBER OF POINTS IN ATMOSPHERE =',I4////)
       DO 100 J=1,2
   90  CONTINUE
       I = I + 1
       READ(8,10,END=100) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)
       GO TO 90
  100  CONTINUE
       N = I - 2
       I = 0
       REWIND 8
    1  CONTINUE
       I = I + 1
       READ(8,10,END=2) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)
       GO TO 1
    2  CONTINUE
       NFIT = I - 2
       I = N
    3  CONTINUE
       I = I - 1
       READ(8,10,END=4) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)
       GO TO 3
    4  CONTINUE
       I = N - 2
    5  CONTINUE
       I = I + 1
       READ(8,10,END=6) A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I)
       GO TO 5
    6  CONTINUE
       NP1 = I - 1
       WRITE(9,10) (A(I),B(I),C(I),D(I),E(I),F(I),G(I),H(I),I=2,NP1)
       WRITE(9,20)
       REWIND 8
       READ(8,98) NMSUN, X, Y, PC, TC, L0, R0
   98  FORMAT(A4,6A8)
```

```
      WRITE(3,99) NMSUN, X, Y, PC, TC, L0, R0
   99 FORMAT(///19,' SOLAR MASSES'/'0       X ='F7.4/'       Y =',F7.4/
     X           '0      PC =',1PD23.16/'       TC =',1PD23.16/
     X           '       L0 =',1PD23.16/'       R0 =',1PD23.16////)
      NTOT = NP1 - 1
      NSURF = N - 2
      NATMO = NTOT - NSURF
      WRITE(3,30) NTOT, NFIT, NSURF, NATMO
      STOP
      END
/*
/EXEC     LDR
/DD   8   DSNAME=TEMP,DISP=SCR
/DD   9   DSNAME=M200,DEVT=(2314,FILPRO),RECFM=F,LRECL=84,DISP=KEEP
/EXEC     *
/DD   22  DSNAME=FZNF,DISP=KEEP
/*
```

REFERENCES

Allen, C.W. 1963, Astrophysical Quantities, (London: Athlone Press).

Appenzeller, I. 1970, Astron. and Astrophys., 5, 355.

Arnett, D. 1966, Can. J. of Phys., 44, 2553.

Blanco, A.M. 1963, Ap. J., 137, 513.

Boury, A., Gabriel, M., et Ledoux, P. 1964, Ann. d'Ap., 27, 92.

Buscombe, W. 1965, M.N.R.A.S., 129, 411.

Chandrasekhar, S. 1957, An Introduction to the Study of Stellar Structures (New York: Dover).

Christy, R. 1964, Rev. Mod. Phys., 36, 555.

Christy, R. 1966a, Ann. Rev. Astr. and Ap., 4, 353.

Christy, R. 1966b, Ap. J., 144, 108.

Christy, R. 1967, in Methods in Computational Physics, 7, 191 (New York: Academic Press).

Christy, R. 1968, Lectures on Variable Stars (California Institute of Technology).

Clayton, D. 1968, Principles of Stellar Evolution and Nucleosynthesis (New York: McGraw-Hill).

Cocke, W. and Cohen, J. 1968, Nature, 219, 1009.

Colgate, S. and White 1966, Ap. J., 143, 626.

Courant, R., Friedrichs, K., and Lewy, H. 1928, Math. Ann., 100, 32.

Courant, R., Issacson, E. and Rees, M. 1952, Comm. Pure App. Math., 5, 243.

Cox, A.N. and Stewart, J.H. 1965, Ap. J. Suppl., 11, 22.

Cox, J.P. 1955, Ap. J., 122, 286.

Crawford, D.L. and Perry 1966, A.J., 71, 206.

Eddington, A. 1926, The Internal Consititution of Stars (New York: Dover).

Eisberg, R.M. 1961, Fundamentals of Modern Physics (New York: John Wiley and Sons, Inc.).

Ezer, D., and Cameron, A.G.W. 1963, Icarus, 1, 422.

Harris, D.L., III 1963, Basic Astronomical Data, ed. D.A. Strand (Chicago: University of Chicago Press), p. 263.

Hayashi, D., Hoshi, R., and Sugimoto, D. 1962, Prog. Theor. Phys. Suppl., No. 22.

Hiltner, W.A. 1956, Ap. J. Suppl., 2, 389.

Hiltner, W.A., Morgan, W., and Neff, J. 1965, Ap. J., 141, 183.

Hoag, A.A. and Applequist, N. 1965, Ap. J. Suppl., 12, 215.

Hoag, A.A., Johnson, H.L., Iriarte, B., Mitchell, R.I., Hallam, K.L. and Sharpless, S. 1961, Publ. U.S. Naval Obs., 17, part 7.

Iben, R. and Talbot, R.J. 1966. Ap. J., 144, 968.

Idlis, G.M. 1957, Astr. Zh., 34, 733.

Johnson, H.L. 1960, Lowell Obs. Bull., 5, 17.

Johnson, H.L. 1966, Ann. Rev. of Astro. and Astrophys., 4, 193.

Johnson, H.L., Hoag, A.A., Iriarte, B., Mitchell, R.I., Hallam, K.L. 1961, Lowell Obs. Bull., 5, 133.

Johnson, H.L. and Mitchell, R.I. 1958, Ap. J., 128, 31.

Johnson, H.M. and Hogg, D.E. 1965, Ap. J., 142, 1033.

Lax, P.D. and Richtmyer, R.D. 1956, Comm. Pure Appl. Math., 9, 267.

Lax, P. and Wendroff B. 1960, Comm. Pure Appl. Math., 13, 217.

Ledoux, P., and Walraven, T. 1958, Hdb. d. Phys., ed. S. Flugge (Berlin:) Springer-Verlag) 51, 353.

McCormick and Salvapori 1964, Numerical Methods in Fortran (New Jersey: Prentice Hall).

Mitchell, R.I. 1960, Ap. J., 132, 68.

Morgan, W.W., Code, A.D., and Whitford, A.E. 1955, Ap. J. Suppl., 2, 41.

Morgan, W.W. and Hiltner, W.A. 1965, Ap. J., 141, 177.

Morgan, W.W., Hiltner, W.A., Neff, J., Garrison, R., and Osterbreck, D. 1965, Ap. J., 142, 974.

Morton, D.C. and Adams, T.F. 1968, Ap. J., 151, 611.

Osaki, Y. 1966, Publ. Astron. Soc. Japan, 18, 384.

Reddish, V.C. 1967, M.N.R.A.S., 135, 251.

Reeves, H. 1965, in Stars and Stellar Systerms, 8, Chapt. 2 (Chicago: University of Chicago).

Reeves, H. 1966, Stellar Evolution (New York: Plenum Press).

Richtmyer, R.D. and Morton, D.C. 1967, Difference Methods for Initial Value Problems (New York: John Wiley and Sons, Inc.).

Salpeter, E.E. 1955, Ap. J., 121, 161.

Sandage, A. 1957a, Ap. J., 125, 422.

Sandage, A. 1957b, Ap. J., 125, 435.

Schild, R. 1967, Ap. J., 148, 449.

Schwarzchild, M. 1958, Structure and Evolution of the Stars (New Jersey: Princeton University Press).

Schwarzchild, M., and Harm, R. 1958, Ap. J., 129,637.

Schwarzchild, M., and Harm, R. 1959, Ap. J., 128, 348.

Sharpless, S. 1952, Ap. J., 116, 251.

Simon, N. 1970a, Ap. J., 159, 859.

Simon, N. 1970b, Private Communication.

Simon, N. and Stothers, R. 1969a, Ap. J., 155, 247.

Simon, N. and Stothers, R. 1969b, Ap. J., 156, 377.

Simon, N. and Stothers, R. 1970, Astron. and Astrophys., 6, 183.

Smith, L.F. 1967, A.J., 72, 829.

Stothers, R. 1965, Ap. J., 141, 671.

Stothers, R. 1966, Ap. J., 144, 959.

Stothers, R. 1968, Ap. J., 156, 541.

Stothers, R. 1969, in Stellar Astronomy, ed. H.Y. Chiu, J. Remo, R. Warsilla (New York: Gordon and Breach) p. 183.

Stothers, R. and Simon, N. 1968, Ap. J., 152, 233.

Stothers, R. and Simon, N. 1969, Ap. J., 157, 673.

Stothers, R. and Simon, N. 1970, Ap. J., 160, 1019.

Strand, K. 1958, Ap. J., 128, 14.

Talbot, R. 1970a, Ap. J., (To be published).

Talbot, R. 1970b, Ap. J., (To be published).

Thackeray, A.D. and Wesselink, A.F. 1965, M.N.R.A.S., 131, 121.

Thackeray, A.D., Wesselink, A.F. and Harding, G.A. 1962, M.N.R.A.S., 124, 445.

Underhill, A.B. 1969, Astrophys. and Space Sci., 3, 109.

Unno, W., Kato, S., and Osaki, Y. 1967, Zeits. F. Astrophys. 65, 327.

Van Den Bergh, S. 1957, Ap. J., 125, 445.

Van Den Bergh, S. and Sher, D. 1960, Pub. David Dunlap Obs., 2, 203.

Vasilevskis, S., Sanders, W., and Van Altena, W.F. 1956a, A.J., 70, 797.

Vasilevskis, S., Sanders, W., and Van Altena, W.F. 1956b, A.J., 70, 806.

Walker, M. 1956, Ap. J. Suppl., 2, 365.

Walker, M. 1957, Ap. J., 125, 636.

Walker, M. 1959, Ap. J., 130, 57.

Walker, M. 1961, Ap. J., 133, 438.

Wampler, E.J. 1962, Ap. J., 136, 100.

Westerlund, B.E. and Henize, K.G. 1963, Pub. A.S.P., 75, 332.

Wildey, R. 1964, Ap. J. Suppl., 8, 439.

Zeibarth, K. 1970, On The Upper Mass Limit for Main Sequence Stars, Preprint.